



GAMP 5

A Risk-Based Approach to
Compliant GxP Computerized Systems

Second Edition





GAMP 5

A Risk-Based Approach to Compliant GxP Computerized Systems

Second Edition

Disclaimer:

The Guide is meant to assist life-sciences companies in managing GxP regulated systems. This Guide is solely created and owned by ISPE. It is not a regulation, standard or regulatory guideline document. ISPE cannot ensure and does not warrant that a system managed in accordance with this Guide will be acceptable to regulatory authorities. Further, this Guide does not replace the need for hiring professional engineers or technicians.

Limitation of Liability

In no event shall ISPE or any of its affiliates, or the officers, directors, employees, members, or agents of each of them, or the authors, be liable for any damages of any kind, including without limitation any special, incidental, indirect, or consequential damages, whether or not advised of the possibility of such damages, and on any theory of liability whatsoever, arising out of or in connection with the use of this information.

© 2022 ISPE. All rights reserved.

All rights reserved. No part of this document may be reproduced or copied in any form or by any means – graphic, electronic, or mechanical, including photocopying, taping, or information storage and retrieval systems – without written permission of ISPE.

All trademarks used are acknowledged.

ISBN 978-1-946964-57-1

Preface

The COVID-19 global pandemic underlined the essential role of innovation and technical advance in the protection of public health. This Second Edition of the *ISPE GAMP® 5 Guide* is intended to support such innovation and technical advance while safeguarding product quality and patient safety.

Such innovation is essential for life sciences industries in providing value to society while also controlling costs and reducing time to market.

Operating in a highly regulated industry may lead practitioners to apply prescriptive and rigid compliance-based approaches that are not commensurate with and not effective in managing any actual risk to the product and the patient.

This Guide facilitates the effective and efficient use of valuable resources by the application of appropriate and proportionate practices, encouraging innovative approaches to managing risk to patient safety, product quality, and data integrity, while supporting benefit to public health.

Acknowledgements

The Guide was produced by an international Task Team, under the leadership and direction of:

Chris Clark	TenTenTen Consulting Limited	United Kingdom
Heather Watson	GSK (retired)	United Kingdom
Sion Wyn	Conformity Ltd.	United Kingdom

The work was supported by the ISPE GAMP Community of Practice (CoP).

Chapter Leads

The following individuals took lead roles in the preparation of this document.

James Canterbury	Ernst and Young LLP	USA
Mark Cherry	AstraZeneca	United Kingdom
Frank Henrichmann	QFINITY	Germany
Paul Irving	Northern Life Sciences Ltd.	United Kingdom
Arthur D. Perez, PhD	Novartis (retired)	USA
Chris Reid	Integrity	United Kingdom
Michael L. Rutherford	Syneos Health	USA
Lorrie Vuolo-Schuessler	Syneos Health	USA
Eric J. Staib	Signant Health	USA
Thana Subramanian	Integrity	United Kingdom
Charlie Wakeham	Waters Corporation	Australia
Christopher H. White	National Resilience, Inc.	USA
Guy Wingate	GSK (retired)	United Kingdom

Contributors

The Leads thank the following individuals for their valuable contribution during the preparation of this Guide.

Sam Andrews	Novartis	United Kingdom
Karen Ashworth	Karen Ashworth Consulting Ltd.	United Kingdom
Carsten Bierans	Körber Pharma Software	Germany
Stephen R. Ferrell	CompliancePath (an Ideagen Company)	USA
James Gannon	PharmaLedger Association	Ireland
Senthil Gurumoorthi	Amazon	USA
Robert Hahnraaths	Bayer	Germany
James Henderson	Eli Lilly and Company	USA
Oliver Herrmann	QFINITY	Germany
David Samuel Holt	Factorytalk Co., Ltd.	Thailand
Torsten Isenberg	Körber Pharma Software	Germany
Paige Kane, PhD	Merck & Co., Inc.	USA
David Margetts	Factorytalk Co., Ltd.	Thailand
Andrew McDonagh	Emergn	United Kingdom
Elizabeth McLellan	Suvoda	USA
Sandy Meek	Parexel	United Kingdom
Khaled Moussally	Compliance Group Inc.	USA
Ray Murphy	Boston Scientific	Ireland
Mark E. Newton	Heartland QA	USA
Donal O'Brien	Dassault Systèmes	Ireland
Margrét Pétursdóttir	Alvotech	Iceland

Rajdeep Poddar	Novartis Healthcare Pvt. Ltd.	India
Edgar Röder	PricewaterhouseCoopers	Germany
Gregory Ruklic	GMR Consultants	USA
Judith S. Samardelis	GSK	USA
Levi Schenk	CSL Behring	USA
Tanya Sharma	Assurea, LLC	USA
Ken Shitamoto	Gilead	USA
Brandi M. Stockton	Signant Health	USA
Tomos Gwyn Williams	Manchester Imaging	United Kingdom
Christian Wölbeling	Körber Pharma Software	Germany

Regulatory Input and Review

The Core Team wish to thank the following individuals for their review and valuable comments on this Guide:

Kevin Bailey	MHRA	United Kingdom
Eric Dong	FDA	USA
Arno Terhechte, PhD	Bezirksregierung Münster	Germany
Seneca D. Toms	FDA	USA

Special Thanks

The Leads would like to thank ISPE for technical writing and editing support by Jeanne Perez (ISPE Guidance Documents Technical Editor) and production support by Lynda Goldbach (ISPE Publications Manager).

The Team Leads would like to express their grateful thanks to the many individuals and companies from around the world who reviewed and provided comments during the preparation of this Guide; although they are too numerous to list here, their input is greatly appreciated.

Company affiliations are as of the final draft of the Guide.



600 N. Westshore Blvd., Suite 900, Tampa, Florida 33609 USA
Tel: +1-813-960-2105, Fax: +1-813-264-2816

www.ISPE.org

Table of Contents

1	Introduction	9
1.1	Rationale for GAMP 5 Second Edition	9
1.2	New and Revised Material	11
1.3	Purpose.....	11
1.4	Scope.....	12
1.5	Business Benefits	13
1.6	Structure	14
2	Key Concepts	17
2.1	Overview	17
2.2	Key Terms	19
3	Life Cycle Approach	23
3.1	Computerized System Life Cycle.....	23
3.2	Specification and Verification	25
3.3	Computerized System Validation Framework.....	26
3.4	Critical Thinking Through the Life Cycle	26
4	Life Cycle Phases.....	29
4.1	Concept.....	29
4.2	Project.....	29
4.3	Operation	40
4.4	Retirement	47
5	Quality Risk Management	49
5.1	Overview	49
5.2	Science-Based Quality Risk Management.....	50
5.3	Quality Risk Management Process.....	51
6	Regulated Company Activities.....	55
6.1	Governance for Achieving Compliance.....	55
6.2	System-Specific Activities	59
7	Supplier Activities.....	69
7.1	Supplier Products, Applications, and Services.....	69
7.2	Supplier Good Practices	70
7.3	Quality Management System.....	71
7.4	Requirements.....	72
7.5	Supplier Quality Planning.....	73
7.6	Sub-Supplier Assessments	73
7.7	Specifications.....	73
7.8	Design Reviews	74
7.9	Software Production/Configuration	74
7.10	Testing.....	75
7.11	Commercial Release.....	75
7.12	User Documentation and Training	75
7.13	System Support and Maintenance During Operation	76
7.14	System Replacement and Retirement	76

8	Efficiency Improvements	77
8.1	Establishing Verifiable and Objective User Requirements.....	77
8.2	Making Risk-Based Decisions	78
8.3	Leveraging Supplier Input.....	79
8.4	Leveraging Existing Information.....	79
8.5	Using Efficient Testing Practices.....	80
8.6	Employing a Well-Managed Handover Process	82
8.7	Managing Changes Efficiently	82
8.8	Anticipating Data Archiving and Migration Needs.....	84
8.9	Using Tools and Automation	84

Management Appendices

9	Appendix M1 – Validation Planning	85
10	Appendix M2 – Supplier Assessment	93
11	Appendix M3 – Science-Based Quality Risk Management.....	107
12	Appendix M4 – Categories of Software and Hardware	127
13	Appendix M5 – Design Review and Traceability	135
14	Appendix M6 – Supplier Quality Planning.....	141
15	Appendix M7 – Validation Reporting	145
16	Appendix M8 – Project Change and Configuration Management	149
17	Appendix M9 – Documentation and Information Management.....	153
18	Appendix M10 – System Retirement	157
19	Appendix M11 – IT Infrastructure.....	163
20	Appendix M12 – Critical Thinking	171

Development Appendices

21	Appendix D1 – Specifying Requirements	183
22	Appendix D2 (Retired).....	197
23	Appendix D3 – Configuration and Design.....	199
24	Appendix D4 – Management, Development, and Review of Software	207
25	Appendix D5 – Testing of Computerized Systems.....	213

26 Appendix D6 – System Descriptions	235
27 Appendix D7 – Data Migration	239
28 Appendix D8 – Agile Software Development	245
29 Appendix D9 – Software Tools.....	253
30 Appendix D10 – Distributed Ledger Systems (Blockchain).....	257
31 Appendix D11 – Artificial Intelligence and Machine Learning (AI/ML)	269

Operation Appendices

32 Appendix O – Introduction to Operation Appendices	281
33 Appendix O1 – Handover	283
34 Appendix O2 – Establishing and Managing Support Services	287
35 Appendix O3 – System Monitoring	291
36 Appendix O4 – Incident Management and Problem Management.....	295
37 Appendix O5 – Corrective and Preventive Action.....	299
38 Appendix O6 – Operational Change and Configuration Management	303
39 Appendix O7 (Retired).....	311
40 Appendix O8 – Periodic Review	313
41 Appendix O9 – Backup and Restore	319
42 Appendix O10 – Business Continuity Management	325
43 Appendix O11 – Security Management.....	331
44 Appendix O12 – System Administration	337
45 Appendix O13 – Archiving and Retrieval.....	341

Special Interest Topics Appendices

46 Appendix S1 – Alignment with ASTM E2500	347
47 Appendix S2 – Electronic Production Records.....	351
48 Appendix S3 – End User Applications Including Spreadsheets.....	361

49 Appendix S4 - Patch and Update Management..... 371
50 Appendix S5 (Retired) 375
51 Appendix S6 - Organizational Change..... 377

General Appendices

52 Appendix G1 - References 383
53 Appendix G2 - Glossary 389
53.1 Acronyms and Abbreviations 389
53.2 Definitions 392

1 Introduction

GAMP guidance aims to safeguard patient safety, product quality, and data integrity in the use of GxP computerized systems. It aims to achieve computerized systems that are fit for intended use and meet current regulatory requirements by building upon existing industry good practice in an efficient and effective manner.

GAMP provides practical guidance that:

- Facilitates the interpretation of regulatory requirements
- Establishes a common language and terminology
- Promotes a system life cycle approach based on good practice
- Clarifies roles and responsibilities

It is not a prescriptive method or a standard, but rather provides pragmatic guidance, approaches, and tools for the practitioner.

When applied with expertise and good judgment, this Guide offers a robust, cost-effective approach.

The approach described in this document is designed to be compatible with a wide range of other models, methods, and schemes including:

- Quality systems standards and certification schemes, such as the ISO 9000 Series [1]
- ISO 14971 [2]: Medical devices – Application of risk management to medical devices
- Schemes for assessing and improving organization capability and maturity, such as Capability Maturity Model Integration® (CMMI) [3]
- Software process models such as ISO 12207 [4]
- Iterative, and incremental (Agile) software development methods and models
- Approaches to IT service management, such as ITIL [5]

Where possible, terminology is harmonized with standard international sources such as ICH [6] and ISO [7].

This Guide aims to be fully compatible with the approach described in the ASTM E2500 Standard Guide for Specification, Design, and Verification of Pharmaceutical and Biopharmaceutical Manufacturing Systems and Equipment [8].

GAMP is an ISPE Community of Practice. For further information see www.ispe.org.

1.1 Rationale for GAMP 5 Second Edition

This Second Edition Guide aims to protect patient safety, product quality, and data integrity by facilitating and encouraging the achievement of computerized systems that are effective, reliable, and of high quality.

While the overall approach, framework, and key concepts remain unchanged, technical content of the Guide has been updated to reflect the increased importance of IT service providers including cloud service providers, evolving approaches to software development including incremental and iterative models and methods, and increased use of software tools and automation to achieve greater control, higher quality, and lower risks throughout the life cycle.

Associated with this is the reinforcement of the message that the GAMP specification and verification approach is not inherently linear but also fully supports iterative and incremental (Agile) methods.

Guidance on the application of new and developing technological areas such as Artificial Intelligence and Machine Learning (AI/ML), blockchain, cloud computing, and Open-Source Software (OSS) has been included or updated.

The importance of critical thinking and the application of patient-centric, risk-based approaches (aimed at quality and safety) versus primarily compliance-driven approaches is further underlined. Concepts of computerized systems assurance as discussed as part of the US FDA Center for Devices and Radiological Health (CDRH) Case for Quality program [9] are also explored and applied.

Links and references to GAMP guidance on the topic area of record and data integrity have been added.

The following ISPE initiatives and topic areas are supported and links and synergies with them have been considered:

Knowledge Management – focusing on how organizations create, manage, and use knowledge throughout the life cycle of a product, enabling organizations to better manage their knowledge as a key asset, in turn improving the effectiveness of the pharmaceutical quality system, and providing operational benefits. [10]

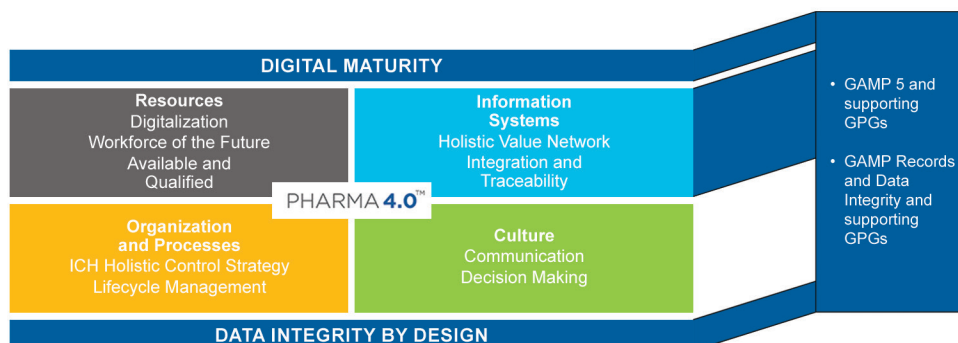
APQ (Advancing Pharmaceutical Quality) – building industry-for-industry tools and programs to help companies assess and improve their quality operations. [11]

Pharma 4.0™ – providing guidance, aligned with the regulatory requirements specific to the pharmaceutical industry, to accelerate Pharma 4.0 transformations. Also known as the Smart Factory, the objective of Pharma 4.0 is to enable organizations involved in the product life cycle to leverage the full potential of digitalization to provide faster innovations for the benefit of patients. [12]

Digital maturity and data integrity by design are enablers to an effective digitalization strategy and are underpinned by well-managed automation and information systems. GAMP guidance aims to ensure that GxP computerized systems are fit for intended use, and that GxP electronic records and data are managed throughout the data life cycle in order to ensure data integrity. See Figure 1.1.

GAMP guidance adopts a patient-centric risk-based approach that enables innovation while demonstrating compliance with regulatory requirements. Pharma 4.0 builds on best practices based on ISPE Guidelines that are enhanced by the digital transformational approach to real time data driven processes.

Figure 1.1: Pharma 4.0 [13]



1.2 New and Revised Material

New guidance has been included on the following topics:

- Appendix D8 – Agile
- Appendix D9 – Software Tools
- Appendix D10 – Distributed Ledger Systems (Blockchain)
- Appendix D11 – Artificial Intelligence and Machine Learning (AI/ML)
- Appendix M11 – IT Infrastructure
- Appendix M12 – Critical Thinking

Significantly updated guidance has been included on the following topics:

- Appendix D1 – Specifying Requirements
- Appendix S2 – Electronic Production Records

As a result of the above additions and revisions, the following guidance included in the previous version of this Guide have been removed:

- Appendix D2 – Functional Specifications
- Appendix O7 – Repair Activity
- Appendix S5 – Managing Quality within an Outsourced IS/IT Environment

The overall GAMP 5 framework, key concepts, system life cycle, specification and verification approach, and Quality Risk Management (QRM) process (aligned with ICH Q9 [14]) remains unchanged.

1.3 Purpose

The purpose of this Guide is to provide a cost-effective framework of good practice to ensure that computerized systems are effective and of high quality, fit for intended use, and compliant with applicable regulations. The framework aims to safeguard patient safety, product quality, and data integrity while also delivering business benefit. This Guide also provides suppliers to the life-science industry with guidance on the development and maintenance of systems by following good practice.

Patient safety is affected by the integrity of critical records, data, and decisions, as well as those aspects affecting physical attributes of the product. The phrase “patient safety, product quality, and data integrity” is used throughout this Guide to underline this point.

This Guide is intended for use by **regulated companies, suppliers, and regulators**. Suppliers include providers of software, hardware, equipment, system integration services, IT service providers, and IT support services, both internal and external to the regulated company.

The Guide has been designed for use by a wide range of disciplines and responsibilities, including:

- Management
- Quality Unit
- Research
- Development
- Manufacture
- Laboratory
- Engineering
- IT
- Support Staff
- All associated suppliers

GAMP documents are guides and not standards. It is the responsibility of regulated companies to establish policies and procedures to meet applicable regulatory requirements. Consequently, it is inappropriate for regulated companies, suppliers, or products to claim that they are GAMP certified, approved, or compliant.

1.4 Scope

This Guide applies to computerized systems used in regulated activities covered by:

- Good Manufacturing Practice (GMP) (pharmaceutical, including Active Pharmaceutical Ingredient (API), veterinary, and blood)
- Good Clinical Practice (GCP)
- Good Laboratory Practice (GLP)
- Good Distribution Practice (GDP)
- Good Pharmacovigilance Practices (GVP)
- Medical Device Regulations (where applicable and appropriate, e.g., for systems used as part of production or the quality system, and for some examples of Software as a Medical Device (SaMD¹))

These are collectively known as GxP regulations (see Chapter 2 for full definition).

This Guide provides an approach that is suitable for all types of computerized systems, focusing on those based on standard and configurable products, but equally applicable to custom (bespoke) applications.

¹ Medical devices have their own regulatory framework and standards (e.g., ISO 13485 [16], ISO 14971 [17], and IEC 62304 [18]), typically require individual approval, and are often subject to clinical trial evaluation after software verification. GxP computerized systems as discussed in this Guide usually support internal regulated company GxP business processes, whereas SaMD is typically a product in the hands of patients or health care providers.

The principles described can be applied to a wide range of computerized systems. Detailed application of these principles to specific system types (e.g., IT, infrastructure, process control systems, and analytical laboratory systems) is described in supporting *ISPE GAMP Good Practice Guides* [15].

Not all the activities defined in this Guide will apply to every system. The scalable approach, with application of critical thinking, enables regulated companies to select the appropriate system life cycle activities.

This Guide is also consistent with other regulatory demands such as Sarbanes-Oxley (SOX)² and those associated with data privacy. The use of this Guide, however, does not guarantee compliance with, or replace, these regulatory demands, which will define additional requirements where they are applicable.

It is recognized that there are acceptable methods other than those described in this Guide. This Guide is not intended to place any constraints on innovation and development of new concepts and technologies.

1.4.1 **Supplier Aspects**

The computerized system life cycle described in this Guide for a regulated company should not be confused with the need for a defined approach or method for software development, which is the responsibility of the supplier.

This Guide defines activities and responsibilities expected of the supplier in the provision of products and services. These activities perform an important role in supporting regulated company activities. The supplier may be a third party or an internal group of the regulated company. Such internal groups should follow processes consistent with the regulated company Quality Management System (QMS).

This Guide uses various diagrams to represent the system life cycle. These diagrams often present relationships in a linear representation. This is not intended to constrain the choice of development methods and models. Suppliers should use the most appropriate methods and models, which may include iterative and incremental (Agile), evolutionary, exploratory, and prototyping techniques, or the use of DevOps approaches (see also Appendix D8).

Modern systems may have a complex supply chain involving multiple suppliers. This Guide aims to meet the needs of each group.

1.5 **Business Benefits**

Effective, reliable, and high-quality computerized systems assist in achieving the primary objectives of patient safety, product quality, and data integrity, but there are major business benefits in having a defined process that delivers systems fit for intended use, on time, and within budget. Systems that are well defined and specified are easier to support and maintain, resulting in less downtime and lower maintenance costs. Adoption of the approaches described in this Guide will assist regulated companies in managing business risk as well as quality risks.

Specific benefits to both regulated companies and suppliers include:

- Reduction of cost and time taken to achieve and maintain compliance
- Early defect identification and resolution leading to reduced impact on cost and schedule
- Cost-effective operation and maintenance
- Effective change management and continual improvement

² The US Sarbanes-Oxley law [19], specifically Section 404, mandates control of computer systems that generate financial records. Many of the good practice principles and electronic records management controls are relevant to compliance with this law.

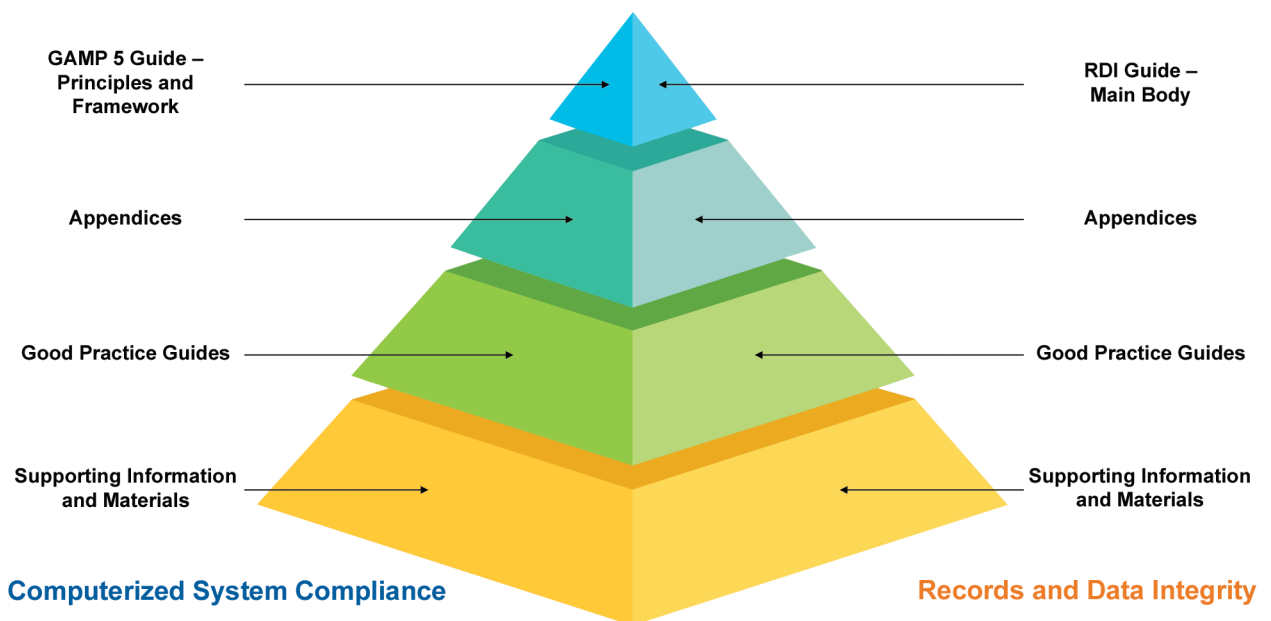
- Enabling of innovation and adoption of new technology
- Providing frameworks for user/supplier cooperation
- Assisting suppliers to produce required documentation
- Promotion of common system life cycle, language, and terminology
- Providing practical guidelines and examples
- Promoting pragmatic interpretation of regulations

1.6 Structure

1.6.1 Overview of GAMP Documentation Structure

This Guide forms part of a family of documents that together provide a powerful and comprehensive body of knowledge covering all aspects of computerized systems' good practice and compliance, as shown in Figure 1.2.

Figure 1.2: GAMP Documentation Structure [20]



This Guide comprises a main body and a set of supporting appendices.

The main body provides principles and a life cycle framework applicable to GxP regulated computerized systems.

Practical guidance on a wide range of topics is provided in the supporting appendices.

Separate *ISPE GAMP Good Practice Guides* [15] cover the application of these general principles and framework to specific types of systems and platforms. Other GAMP GPGs provide detailed approaches to specific activities and topics. For information about available *ISPE GAMP Good Practice Guides*, see www.ispe.org.

1.6.2 GAMP 5 Main Body Structure

The main body introduction covers the purpose, scope, benefits, and structure of this Guide. Subsequent sections of the main body cover the topics:

- Key concepts
- Life cycle approach
- Life cycle phases:
 - Concept
 - Project
 - Operation
 - Retirement
- QRM
- Regulated company activities:
 - Governance for achieving compliance
 - System-specific activities
- Supplier activities
- Efficiency improvements

The **key concepts**, described in Chapter 2, are the five concepts that underpin the rest of the document and should be applied using critical thinking.

The computerized system **life cycle** encompasses all activities from initial concept, understanding of the requirements, through development or purchase, release, and operational use, to system retirement. Chapter 3 describes these activities and how they are related.

Chapter 4 describes the **project life cycle phase** in more detail, including:

- Planning
- Specification, configuration, and coding
- Verification
- Reporting and release

The key supporting processes of risk management, change and configuration management, design review, traceability, and document management are also introduced.

QRM is a systematic approach for the identification, assessment, control, communication, and review of risks to patient safety, product quality, and data integrity. It is an iterative process applied throughout the system life cycle. Chapter 5 describes this approach and how these activities should be based on good science and product and process understanding.

Ensuring compliance and fitness for purpose is the responsibility of the regulated company. Effective and consistent **regulated company activities** for individual systems require a defined organizational and governance framework, covering aspects such as policies, responsibilities, management, and continual improvement. Governance and system-specific regulated company activities are covered in Chapter 6.

While the responsibility for compliance lies with the regulated company, the supplier has a key role to play. An overview of typical **supplier activities** is given in Chapter 7.

This Guide provides a flexible framework for achieving compliant computerized systems that are fit for intended use, but the full benefits can be obtained only if the framework is applied effectively in the context of a particular organization. Chapter 8 covers key topics leading to **efficiency improvements**.

2 Key Concepts

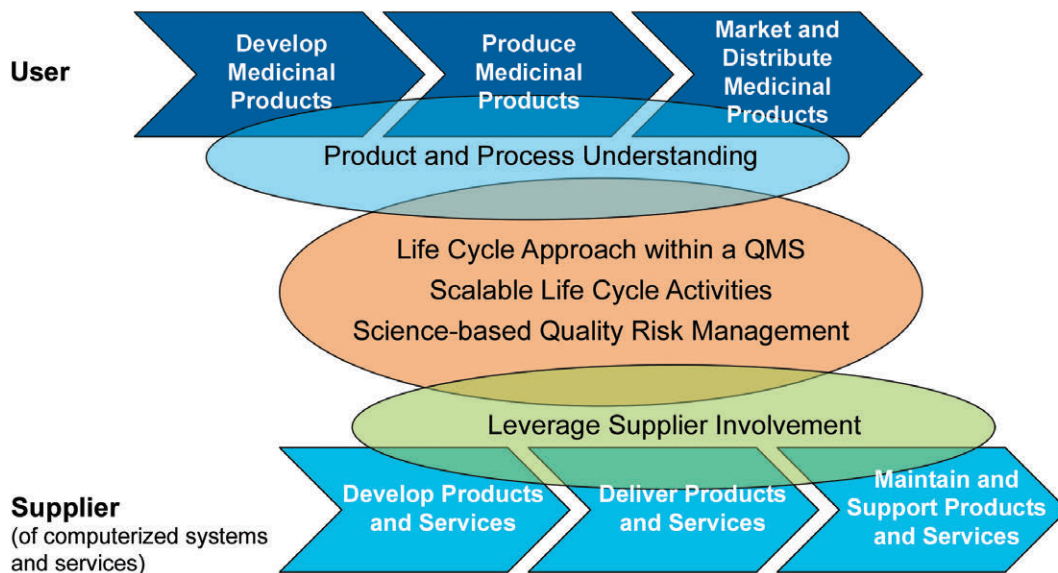
2.1 Overview

Five key concepts are applied throughout this Guide:

1. Product and process understanding
2. Life cycle approach within a QMS
3. Scalable life cycle activities
4. Science-based QRM
5. Leveraging supplier involvement

The relationship between these concepts is shown in Figure 2.1.

Figure 2.1: Key Concepts of this Guide [20]



2.1.1 Product and Process Understanding

An understanding of the supported process is fundamental to determining system requirements. Product and process understanding is the basis for making science- and risk-based decisions to ensure that the system is fit for its intended use. An understanding of the intended use of data within the process is also fundamental. Data integrity cannot be achieved without a complete understanding of the data flow.

As noted in ICH Q10 [21]:

“product and process knowledge should be managed from development through the commercial life of the product up to and including product discontinuation...Knowledge management is a systematic approach to acquiring, analysing, storing, and disseminating information related to products, manufacturing processes and components.”

Efforts to ensure fitness for intended use should focus on those aspects that are critical to patient safety, product quality, and data integrity. These critical aspects should be identified, specified, and verified.

Systems within the scope of this Guide support a wide range of processes, including but not limited to clinical trials, toxicological studies, API production, formulated product production, warehousing, distribution, and pharmacovigilance.

For some manufacturing systems, process requirements depend on a thorough understanding of product characteristics. For these systems, identification of Critical Quality Attributes (CQAs) and related Critical Process Parameters (CPPs) enable process control requirements to be defined.

Specification of requirements should be focused on critical aspects. The extent and detail of requirement specification should be commensurate with associated risk, complexity, and novelty of the system. Requirements may be developed iteratively or incrementally, based upon an initial set of base requirements.

Incomplete process understanding hinders effective and efficient compliance and achievement of business benefit.

2.1.2 Life Cycle Approach within a QMS

Adopting a complete computerized system life cycle entails defining activities in a systematic way from system conception to retirement. This enables management control and a consistent approach across systems.

The life cycle should form an intrinsic part of the company's QMS, which should be maintained and kept up-to-date as new ways of working are developed.

As experience is gained in system use, the QMS should enable continual process and system improvements based on periodic review and evaluation, operational and performance data, and root-cause analysis of failures. Identified improvements and corrective actions should follow change management.

A suitable life cycle, properly applied, enables the assurance of quality and fitness for intended use, and achieving and maintaining compliance with regulatory requirements. A well-managed and understood life cycle facilitates adoption of a Quality by Design (QbD) approach.

The life cycle approach is fundamental to this Guide and embodies each of the other key concepts. The life cycle is structured into phases and activities, as described in Chapter 3.

2.1.3 Scalable Life Cycle Activities

Life cycle activities should be scaled according to:

- System impact on patient safety, product quality, and data integrity (risk assessment)
- System complexity and novelty (architecture and nature of system components including maturity and level of configuration or customization)
- Outcome of supplier assessment (supplier capability)

Business impact also may influence the scaling of life cycle activities.

The strategy should be clearly defined in a plan and follow established and approved policies and procedures.

2.1.4 Science-Based Quality Risk Management

QRM is a systematic process for the identification, assessment, control, communication, mitigation, and review of risks.

Application of QRM enables effort to be focused on critical aspects of a computerized system in a controlled and justified manner.

QRM should be based on clear process understanding and potential impact on patient safety, product quality, and data integrity. Combining knowledge management and QRM will facilitate achievement of QMS objectives by providing the means for science- and risk-based decisions related to product quality. For systems controlling or monitoring CPPs, these should be traceable to CQAs, and ultimately back to the Quality Target Product Profile (QTPP) and relevant regulatory submissions.

Qualitative or quantitative techniques may be used to identify and manage risks. Controls are developed to reduce risks to an acceptable level. Implemented controls are monitored during operation to ensure ongoing effectiveness.

A practical risk-management process is described in Chapter 5.

2.1.5 Leveraging Supplier Involvement

Regulated companies should seek to maximize supplier involvement throughout the system life cycle in order to leverage knowledge, experience, and documentation, subject to satisfactory supplier assessment.

For example, the supplier may assist with requirements gathering, risk assessments, the creation of functional and other specifications, system configuration, testing, support, and maintenance.

Planning should determine how best to use supplier documentation, including existing test documentation, to avoid wasted effort and duplication. Justification for the use of supplier documentation should be provided by the satisfactory outcome of supplier assessments, which may include supplier audits.

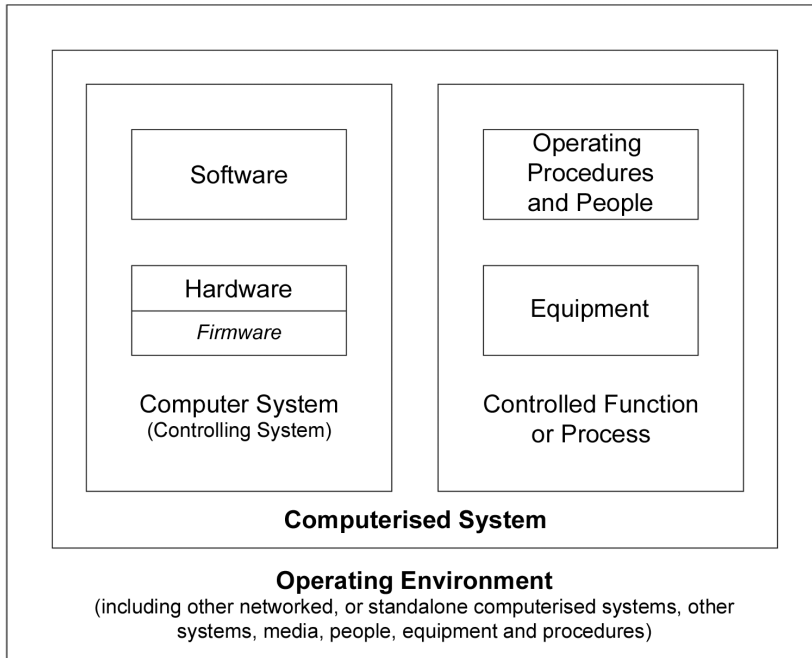
Documentation should be assessed for suitability, accuracy, and completeness. There should be flexibility regarding acceptable format, structure, and documentation practices.

Supplier assessment is described in Chapter 6.

2.2 Key Terms

Computerized System

A computerized system consists of the hardware and software components, together with the controlled function or process (including procedures, people, and equipment and associated documentation) (see also Appendix M11).

Figure 2.2: Computerised System – PIC/S Guidance [22]

This term covers a broad range of systems, including, but not limited to:

- Clinical trials data management
- Manufacturing resource planning
- Laboratory information management
- Automated manufacturing equipment
- Automated laboratory equipment
- Process control and process analysis
- Manufacturing execution
- Building management
- Warehousing and distribution
- Blood processing management
- Adverse event reporting (vigilance)
- Document management
- SaMD and digital health applications

Computerized System Validation

Achieving and maintaining compliance with applicable GxP regulations and fitness for intended use by:

- The adoption of principles, approaches, and life cycle activities within the framework of validation plans and reports
- The application of appropriate operational controls throughout the life of the system

GxP Compliance

Meeting all applicable pharmaceutical and associated life-science regulatory requirements.

GxP Regulated Computerized System

Computerized systems that are subject to GxP regulations. The regulated company must ensure that such systems comply with the appropriate regulations.

GxP Regulation

The underlying international pharmaceutical requirements, such as those set forth in the US FD&C Act [23], US PHS Act [24], FDA regulations [25], EU Directives [26], UK MHRA regulations [27], Japanese regulations [28], or other applicable national legislation or regulations under which a company operates.

These include but are not limited to (further descriptions provided in Appendix G2):

- GMP
- GCP
- GLP
- GDP
- Good Quality Practice (GQP)
- Good Pharmacovigilance Practice (GVP)
- Medical Device Regulations
- Prescription Drug Marketing Act (PDMA) [29]

Process Owner

This is the owner of the business process or processes being managed. The process owner is ultimately responsible for ensuring that the computerized system and its operation is in compliance and fit for intended use in accordance with applicable company Standard Operating Procedures (SOPs). The process owner may also be the system owner. The process owner may be the *de facto* owner of the data residing on the system (data owner) and therefore, ultimately responsible for the integrity of the data. Process owners are typically the head of the functional unit using the system. (cf. System Owner)

Quality Management System (QMS)

Management system to direct and control an organization with regard to quality. (ISO [7]).

(This is equivalent to **Quality System** as defined in ICH Q9 [14].)

Subject Matter Expert (SME)

Those individuals with specific expertise in a particular area or field. SMEs should take the lead role in the verification of computerized systems. SME responsibilities include planning and defining verification strategies, defining acceptance criteria, selection of appropriate test methods, execution of verification tests, and reviewing results. (ASTM E2500 [8])

System Owner

The system owner is responsible for the availability and support and maintenance of a system, and for the security of the data residing on that system. The system owner is responsible for ensuring that the computerized system is supported and maintained in accordance with applicable company SOPs. The system owner also may be the process owner (e.g., for IT infrastructure systems or systems not directly supporting GxP). For systems supporting regulated processes and maintaining regulated data and records, the ownership of the data resides with the GxP process owner, not the system owner.

The system owner acts on behalf of the users. The system owner for larger systems will typically be from IT or engineering functions. Global IT systems may have a global system owner and a local system owner to manage local implementation. (cf. Process Owner)

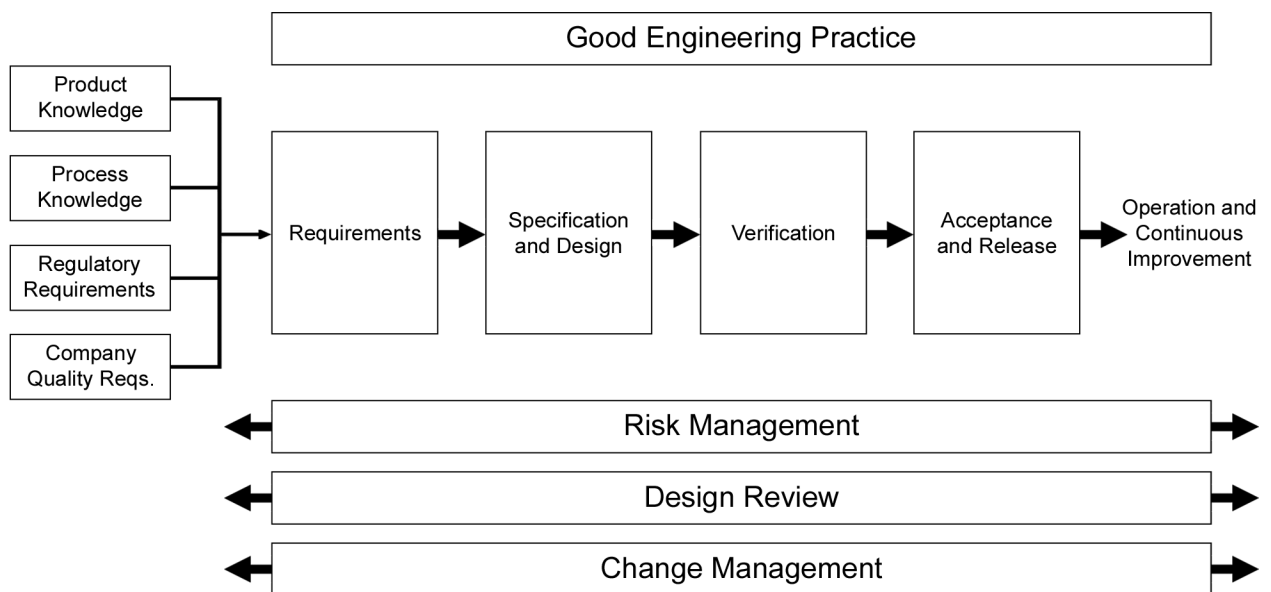
3 Life Cycle Approach

Compliance with regulatory requirements and fitness for intended use may be achieved by adopting a life cycle approach following good practice as defined in this Guide.

A life cycle approach entails defining and performing activities in a systematic way from conception, understanding the requirements, through development, release, and operational use, to system retirement. Figure 3.1 shows a general specification, design, and verification process described in ASTM E2500 [8].

Figure 3.1: The Specification, Design, and Verification Process [8]

Reprinted with permission from ASTM E2500-20 Standard Guide for Specification, Design, and Verification of Pharmaceutical and Biopharmaceutical Manufacturing Systems and Equipment, copyright ASTM International, 100 Barr Harbor Dr., West Conshohocken, PA 19428. A copy of the complete standard may be obtained from ASTM at www.astm.org.



This section of the Guide introduces the computerized system life cycle, a general approach to specification and verification, a framework for computerized system validation, and the application of critical thinking.

3.1 Computerized System Life Cycle

The computerized system life cycle encompasses all activities from initial concept to retirement.

The life cycle for any system consists of four major phases:

- Concept
- Project
- Operation
- Retirement

During the concept phase, the regulated company should consider opportunities to automate one or more business processes based upon business need and benefits. Typically at this phase, initial requirements will be developed and potential solutions considered. From an initial understanding of scope, costs, and benefits, a decision is made on whether to proceed to the project phase. An initial risk assessment or GxP assessment should be performed, so that GxP regulated systems are identified before the project phase commences.

The project phase involves planning, supplier assessment and selection, various levels of specification, configuration (or coding for custom applications), and verification leading to acceptance and release for operation. Risk management is applied to identify risks and to remove or reduce them to an acceptable level.

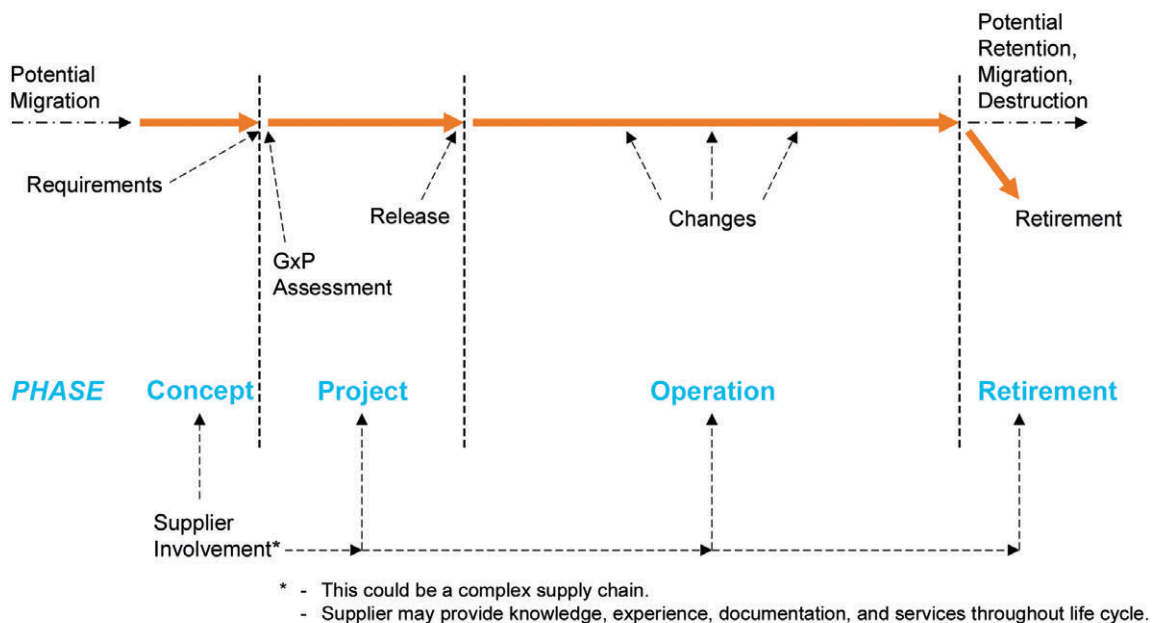
System operation is often the longest phase and is managed using defined, up-to-date, operational processes and procedures applied by personnel who have appropriate training, education, and experience. Maintaining control (including security), fitness for intended use, and compliance are key aspects. The management of changes of different impact, scope, and complexity is an important activity during this phase.

The final phase is the ultimate retirement of the system. It involves decisions about data retention, migration, or destruction, and the management of these processes.

Suppliers of products and services should be involved as appropriate throughout the life cycle. It may be appropriate to delegate many of the described activities to suppliers, subject to satisfactory supplier assessment and control measures.

These life cycle phases are shown in Figure 3.2.

Figure 3.2: Life Cycle Phases



An inventory of computerized systems should be maintained. Further details are given in Chapter 6. A GxP assessment should be performed by the beginning of the project stage to determine whether a system is GxP regulated, and if so, which specific regulations apply, and to which parts of the system they are applicable. This should be performed as part of the initial system risk assessment (Step 1 as described in Chapter 5). For similar systems, it may be appropriate to base the GxP assessment on the results of a previous assessment provided the regulated company has an appropriate established procedure.

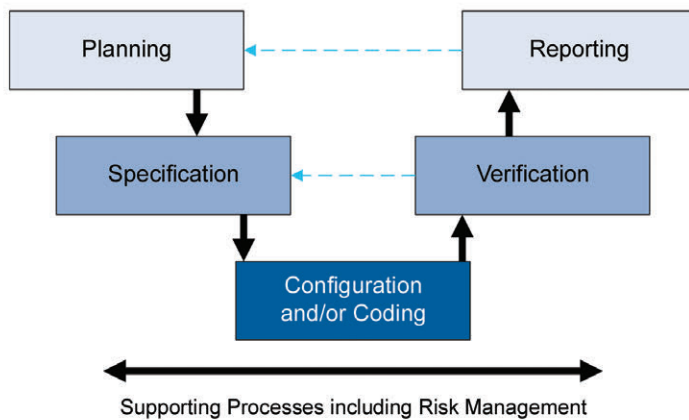
The computerized system life cycle described in this section should not be confused with the need for a defined approach or method for software development by the supplier. Chapter 7 discusses supplier activities in more detail.

3.2 Specification and Verification

The GAMP specification and verification approach is not inherently linear; it also fully supports iterative and incremental (Agile) methods, as shown below.

A linear approach is particularly suitable when system requirements are fully understood and defined upfront. Figure 3.3 shows a linear approach for achieving computerized system compliance and fitness for intended use within the system life cycle. The specification activities have equivalent verification steps to determine whether the specification has been met. A hierarchy of specifications may be required for larger systems, while specifications may be combined for smaller, simpler, or standard systems. Specifications are addressed by appropriate verification steps.

Figure 3.3: Linear Approach to Achieving Compliance and Fitness for Intended Use



The application of this general approach will vary widely depending on the risk, complexity, and novelty of the system. Specific examples showing typical activities for different types of systems are provided in Chapter 4.

As shown in Figure 3.4, the life cycle and specification and verification approach described in the Guide is not inherently linear.

Figure 3.4: Iterative and Incremental Approach to Achieving Compliance and Fitness for Intended Use [20]



This Guide supports the use of Agile approaches for product development, the development of custom applications, and incremental product configuration. Factors for the successful adoption of Agile include a robust QMS within an appropriate organizational culture, well-trained and highly disciplined teams following a well-defined process supported by effective tools and automation, and proper customer or product owner involvement. (See also Appendix D8)

This Guide describes the overall GxP system life cycle from the perspective of the regulated company and does not define the software development process in detail.

While this section provides a suggested approach for activities performed by a regulated company, it is recognized that other models and approaches are equally acceptable.

3.3 Computerized System Validation Framework

GAMP advocates a computerized system validation framework to achieve and maintain GxP compliance throughout the computerized system life cycle.

The framework is based on system-specific validation plans and reports and the application of appropriate operational controls. Validation plans and reports provide a disciplined and consistent approach to meeting regulatory requirements, leading to appropriate documentation at the right level. Such documents are valuable both in preparing for, and during, regulatory inspections.

The framework is applicable for the majority of computerized systems. This Guide has built on these principles by clarifying the scalability of the approach, and the central role of QRM, to effectively and efficiently cover the very wide range of systems in scope.

Where a computer system is regarded as one component of a wider manufacturing process or system, particularly in an integrated QbD environment, specific and separate computerized system validation may not be necessary. This environment requires both complete product and process understanding and that the CPPs can be accurately and reliably predicted and controlled over the design space. In such a case, the fitness for intended use of the computer system within the process may be adequately demonstrated by documented engineering or project activities together with subsequent process validation or continuous quality verification of the overall process or system. The same principle applies to the adoption of Process Analytical Technology (PAT).

For automated manufacturing equipment, separate computer system validation should be avoided. Computer system specification and verification should be part of an integrated engineering approach to ensure compliance and fitness for intended use of the complete automated equipment. Further information can be found in *ISPE Baseline® Guide: Volume 5 – Commissioning and Qualification (Second Edition)* [30].

This framework is described in Appendix M1 and Appendix M7.

3.4 Critical Thinking Through the Life Cycle

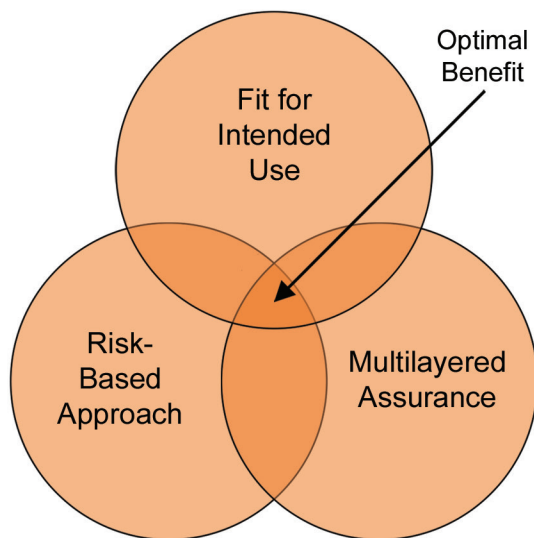
Critical thinking promotes informed decision-making and good judgment on where and how to apply and scale quality and compliance activities for computerized systems. It relies on knowledge of the business process and on the detailed comprehension and analysis of where the business process can potentially impact patient safety, product quality, and data integrity. A better understanding of risks results in more confidence in the assessment and control of those risks, and therefore supporting robust scaling of controls and validation activities.

The extent and depth, formality, and level of documentation formality may, and should, vary to a considerable degree between different business processes, types of systems, functions within a system, and applications. Significant efficiency improvements can be achieved by focusing on what is essential in individual situations and avoiding unnecessary work. Critical thinking is not a one-time activity and should be applied throughout the computerized system life cycle. As such, critical thinking should become a habitual mindset based on an intellectual commitment to continual improvement.

Figure 3.5 illustrates critical thinking for computerized systems whereby proactive adoption of a risk-based approach suitable for the intended use of the computerized system takes into account the multiple layers of assurance provided elsewhere within the business process. In other words, combining technical, procedural, and behavioral controls applied throughout the business process is used when assessing the risk of the computerized system. These layers of assurance may exist upstream or downstream of the system within the business process it supports and include supplier activities [20].

This holistic approach requires an initial investment of time and effort to analyze the overall business process that the computerized system will support, and the associated regulated data. For example, the value of strong, validated data integrity technical controls in the system could be negated by errors caused by manual transcription of regulated data into the system. Data integrity cannot be recovered once lost. Business process mapping and data flow diagrams capture this knowledge and facilitate the identification and understanding of the potential risks to patient safety, product quality, and data integrity to determine where assurance is most needed.

Figure 3.5: Critical Thinking for Computerized Systems [20]



Regulatory authorities are adopting critical thinking to help determine whether controls are fit for intended use to ensure patient safety, product quality, and data integrity. Practitioners should not consider that the level of regulatory compliance achieved is directly proportionate to the amount of paperwork produced. Too much paperwork can confuse and make it harder to maintain and inspect computerized systems. Regulators look for scaled and targeted activities with well-organized information and records that have an appropriate level of detail, supported by clear and unambiguous rationales explaining the critical thinking applied. The information/records contained within software development and support tools offer the opportunity to demonstrate control in areas where separate documentation was previously considered a necessity, and these should be leveraged [31]. (See also Appendix D9)

Further information can be found in Appendix M12 and the *ISPE GAMP® Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20].

4 Life Cycle Phases

This section further describes the phases of the computerized system life cycle introduced in Chapter 3.

The life cycle approach described is not inherently linear, and is designed to be compatible with a wide range of models and methods, including iterative and incremental (Agile) approaches. References to documentation and deliverables should not be interpreted as always requiring traditional documents. The maintenance of records and information in appropriate and effective software tools is encouraged.

4.1 Concept

Detailed activities in this phase will depend on company approaches to initiating and justifying project commencement. Gaining management commitment to provide appropriate resources is an important pre-project activity. An initial risk assessment or GxP assessment should be performed so that GxP regulated systems are identified before the project phase commences.

4.2 Project

This section describes the following project stages in more detail:

- Planning
- Specification, configuration, and coding
- Verification
- Reporting and release

The key supporting processes of risk management, change and configuration management, design review, traceability, and document management also are described in this section.

Figure 4.1 shows how these project stages and supporting processes form part of the computerized system life cycle. The stages are equally applicable to the project phase and to subsequent changes during operation.

This is a simplified general model that describes a staged approach to the project phase. For example, it covers both configuration and coding, which are required only for certain types of systems. Specific examples of how to use this staged approach for typical types of systems are specified in Section 4.2.6.

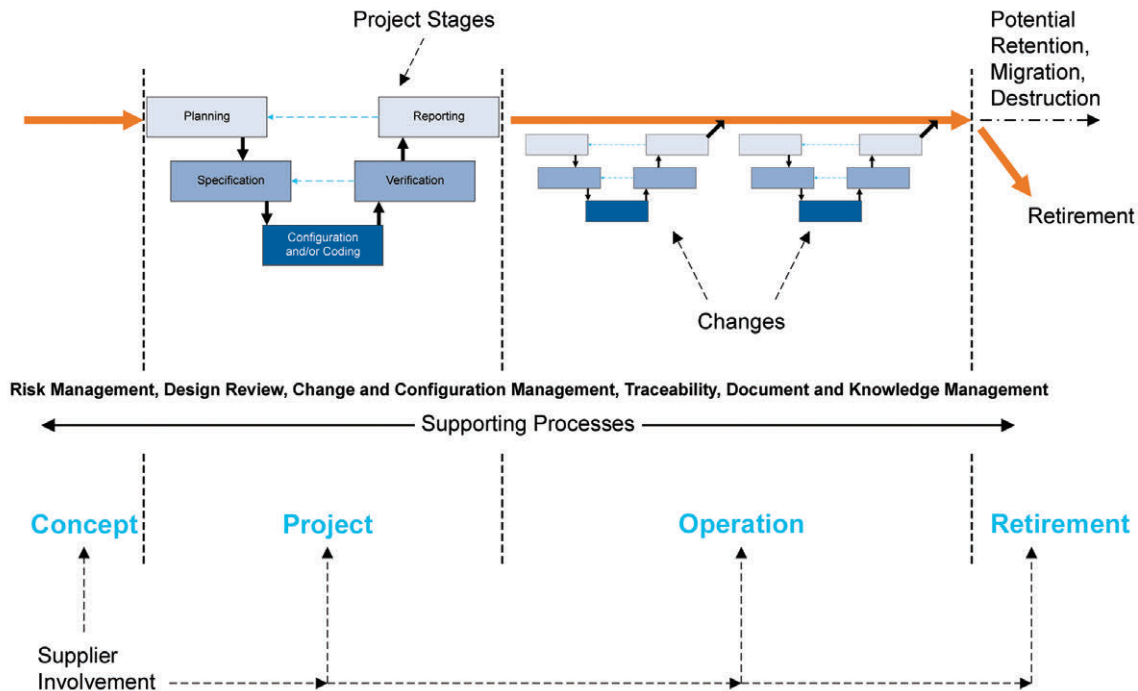
Each project stage typically involves multiple activities; for example, planning may involve:

- Preparing project plans
- Establishing project teams
- Developing requirements
- Conducting supplier assessments
- Defining the extent and formality of verification activities

While the project phase is made up of generally sequential stages, detailed activities often will be performed in parallel or with some overlap. For example, verification activities may occur through several stages. Iterative and incremental approaches may also be used.

Information produced during the execution of these stages provide the evidence that the system is fit for its intended use. Some of this may be used by the regulated company during regulatory inspections to provide justification of the suitability of the system.

Figure 4.1: Project Stages and Supporting Processes within the Life Cycle



4.2.1 Planning

Planning should cover all required activities, responsibilities, procedures, and timelines.

As described in Section 2.1.3, life cycle activities should be scaled according to:

- System impact on patient safety, product quality, and data integrity (risk assessment)
- System complexity and novelty (architecture and nature of system components, including maturity and level of configuration or customization)
- Outcome of supplier assessment (supplier capability)

For further details on computerized system validation planning, see Appendix M1. In some cases, specific computerized system validation plans may not be required as described in Section 3.3.

A clear and complete understanding of the process to be supported and initial user requirements facilitates effective planning. Initial requirements are often gathered during the concept phase and refined and enhanced during the planning stage.

The extent and detail of requirements gathering and specification should be sufficient to support risk assessment, further specification, development of the system, and verification. These activities may be linear or iterative and incremental (Agile). Comparison of available solutions may result in refinement of the requirements.

The approach should be based on product and process understanding and relevant regulatory requirements. Where appropriate, e.g., for process control systems, requirements should be traceable to relevant CPPs and CQAs.

Requirements are the responsibility of the user community and should be maintained and controlled.

See Appendix D1 for further details on Requirements Specifications (RS).

4.2.2 Specification, Configuration, and Coding

The role of specification is to enable systems to be developed, verified, and maintained. The number and level of detail of the specifications will vary depending upon the type of system and its intended use. For example, software design specifications are not expected from the regulated company for non-custom products.

Specifications should be adequate to support subsequent activities, including risk assessment, further specification and development of the system, verification as appropriate, and system maintenance and update.

The requirements for configuration and coding activities depend on the type of system (see Section 4.2.6 for examples).

Any required configuration should be performed in accordance with a controlled and repeatable process. Any required software coding should be performed in accordance with defined standards. The need for code reviews within the organization producing the software should be considered.

Configuration management and version control are intrinsic and vital aspects of controlled configuration and coding.

Figure 4.1 shows specification as a separate activity to configuration and coding. However, specification activities may be distinct from, or tightly linked to, configuration and coding activities depending on the software development method being adopted.

Specifications should be maintained and controlled. See Appendix D1 and Appendix D3 for further details on specification, configuration, and design. See Appendix D4 for further details on the management, development, and review of software.

4.2.3 Verification

Verification confirms that specifications have been met. This may involve multiple stages of reviews and testing depending on the type of system, the development method applied, and its use.

While verification is shown as a single box on Figure 4.1, verification activities occur throughout the project stages. For example, design reviews should verify specifications during the specification stage. See Section 4.2.5 for further details on design reviews.

Terminology used to cover verification activities is described in Section 4.2.6.4.

Testing computerized systems is a fundamental verification activity. Testing is concerned with identifying defects so that they can be corrected, as well as demonstrating that the system meets requirements.

The use of effective and appropriate testing tools is encouraged. Such tools should have documented assessments for their adequacy (refer to EU Annex 11 Clause 4.7 [32]) and be controlled in use; however, validation is not required.

Testing is often performed at several levels depending on the risk, complexity, and novelty. Different types of testing are covered in Appendix D5, Section 6.

Tests may be defined in one or more test specifications to cover hardware, software, configuration, and acceptance.

An appropriate test strategy should be developed based on the risk, complexity, and novelty. Supplier documentation should be assessed and used if suitable. The strategy should define which types of testing are required and the level and purpose of test specifications. The test strategy should be reviewed and approved by appropriate SMEs.

See Appendix D5 for further details on testing and development of an appropriate test strategy.

4.2.4 Reporting and Release

The system should be accepted for use in the operating environment and released into that environment in accordance with a controlled and documented process. Acceptance and release of the system for use in GxP regulated activities should follow a defined process and involve oversight and input of the process owner, system owner, and the appropriate quality function as necessary and applicable. This may be a continuously managed process rather than a one-off event, for example Continuous Integration and Continuous Deployment (CI/CD), and Software as a Service (SaaS).

A computerized system validation report should be produced summarizing the activities performed, any deviations from the plan, any outstanding and corrective actions, and provide a statement of fitness for intended use of the system. See Appendix M7 for further details.

The incorporation of lessons learned/after action review stage gates in the project should be considered (see *ISPE Good Practice Guide: Knowledge Management in the Pharmaceutical Industry*, Appendix 6 [10]).

In some cases, specific computerized system validation reports may not be required, as described in Section 3.3.

A well-managed system handover from the project team to the process owner, system owner, and operational users is a prerequisite for effectively maintaining compliance of the system during operation. See Section 8.6 and Appendix O1 for further details on system handover.

4.2.5 Supporting Processes

4.2.5.1 Risk Management

An appropriate risk-management process should be established.

See Chapter 5 and Appendix M3 for further details on risk management.

4.2.5.2 Change and Configuration Management

Appropriate configuration management processes should be established such that a computerized system and all its constituent components can be identified and defined at any point during its life cycle.

Change management procedures covering software, hardware, and documentation should be established. The point at which change management is introduced should be defined. Appropriate change processes should be applied to both project and operational phases.

Any involvement of the supplier in these processes should be defined and agreed.

See Appendix M8 for further details on project change and configuration management.

4.2.5.3 Design Review

At suitable stages during the life cycle, planned and systematic design reviews of specifications, design, and development should be performed. This design review process should evaluate deliverables to ensure that they satisfy the specified requirements. Corrective actions should be defined and progressed.

The rigor of the design review process and the extent of documentation should be based on risk, complexity, and novelty.

Design reviews are mainly applicable for custom applications and within the supplier product development processes.

See Appendix M5 for further details on design reviews.

4.2.5.4 Traceability

Traceability is a process for ensuring that:

- Requirements are traceable back to business process needs
- Requirements are addressed and traceable to the appropriate functional and design elements in the specifications
- Requirements can be traced to the appropriate verification

As well as demonstrating coverage of design and verification, traceability can greatly assist the assessment and management of change.

Traceability should be focused on aspects critical to patient safety, product quality, and data integrity. The use of traceability tools and automated traceability methods is encouraged.

See Appendix M5 for further guidance on traceability.

4.2.5.5 Documentation Management and Knowledge Management

Management of documentation includes preparation, review, approval, issue, change, withdrawal, and storage. See Appendix M9 for further details on documentation management. Information and records may be maintained in appropriate and effective tools rather than in traditional documents.

Documentation, tools, and systems contain explicit knowledge that is captured and codified, whereas implied knowledge is context-specific knowledge acquired through personal experience or internalization rather than physical media or information systems. Both need to be considered for successful initial and subsequent system deployments.

See the *ISPE Good Practice Guide: Knowledge Management in the Pharmaceutical Industry* [10] for further details on knowledge management.

4.2.6 Practical Examples

This section shows how the general approach may be applied in a scalable manner to three common types of systems, using Categorization of software. Categorization assists in selecting appropriate life cycle activities and deliverables, as described in Appendix M4, based on the nature of the component and likelihood of defects.

Computerized systems are generally made up of a combination of components from different GAMP Categories (as described in the following subsections). For example, core functionality within a computerized system may be Category 3, with Category 4 workflow configuration, and custom interfaces to other systems being Category 5.

Categories 3 to 5 should be viewed as a continuum with no absolute boundaries. The software category is just one factor to consider in a risk-based approach. Life cycle activities should be scaled based on the GxP impact, complexity, and novelty of the system. In all cases, the emphasis should be on ensuring fitness for intended use based on appropriate technical activities following current good practice. Table-driven approaches purely based on categorization decisions should be avoided.

For convenience the activities shown in this section are a linear representation, but iterative and incremental (Agile) approaches are equally acceptable as described in Section 3.2 and Appendix D8.

These examples are intended to be indicative and for illustration purposes only. Actual approaches for specific systems should be based on the results of supplier assessments and risk assessments, in addition to the categorization of system components, as described in Appendices M2, M3, and M4.

The terminology used in these examples is discussed in Section 4.2.6.4.

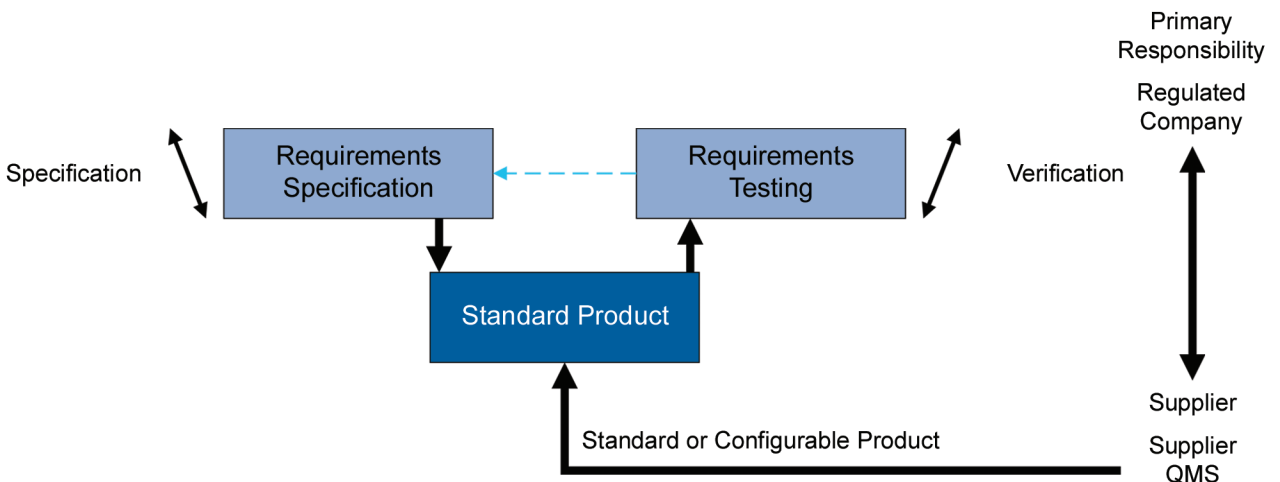
4.2.6.1 Example of a Standard Product

Many computerized systems comprise commercially available software products running on standard hardware components.

Software products that are used off-the-shelf (i.e., standard and not configurable for a specific business process), are typically classified as GAMP Category 3. This includes off-the-shelf components used for business purposes. It includes both those that cannot be configured to conform to business processes, and those that offer defined ranges of factory-provided values or ranges (also called parameterization, as may be found in process control systems and simple laboratory devices). In both cases, configuration to run in the user’s environment is possible and likely (e.g., for printer setup).

In such cases, and based on satisfactory supplier and risk assessments, a simple approach consisting of one level of specification and verification is typically applicable (see Figure 4.2).

Figure 4.2: Approach for a Standard Product (Category 3)



Testing typically covers:

- Correct installation
- Tests that demonstrate fitness for intended use and allow acceptance of the system against requirements
- Any further tests as a result of risk and supplier assessments

Regulated companies typically perform the required specification and verification. While the system is not configured for a business process, there may be some limited configuration such as run-time parameters or printer setup.

Supplier activities typically include supply of the product, and provision of documentation, training, and support and maintenance.

4.2.6.2 Example of a Configured Product

A common type of computerized system involves the configuration of commercially available software products running on standard hardware components. These may be installed on-premise by the regulated company, or increasingly for IT applications such as SaaS, hosted by an IT service provider/supplier.

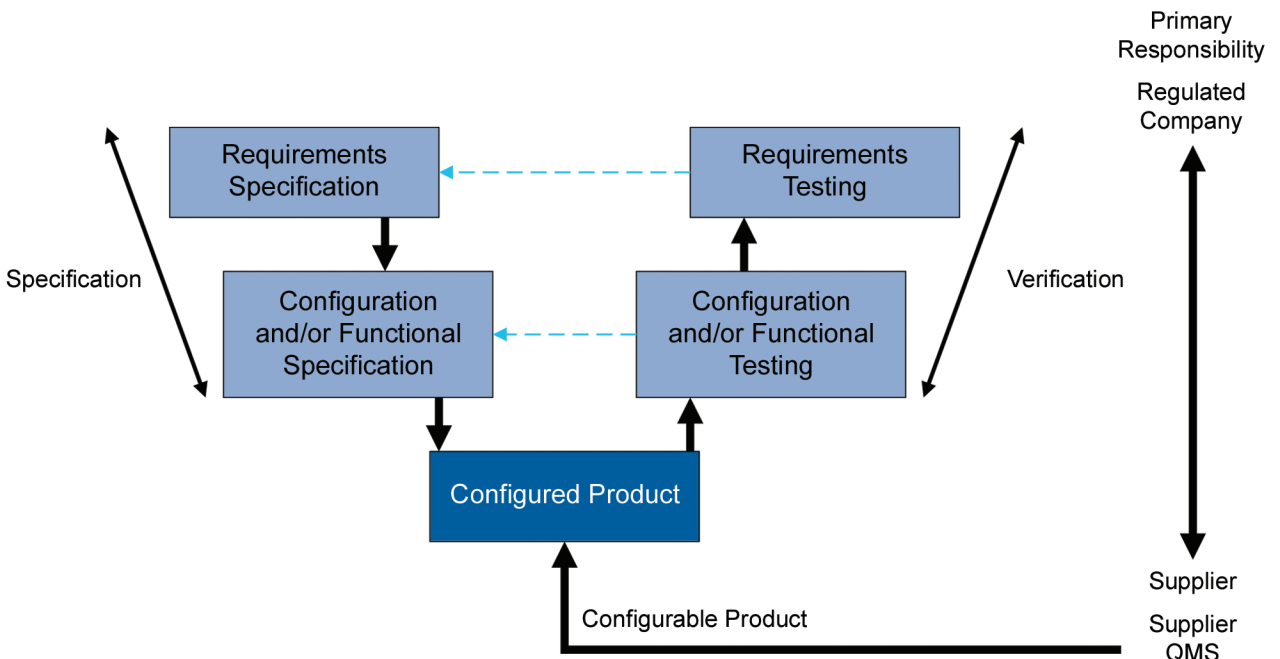
Commercial software products that are configured for a specific business process are typically classified as GAMP Category 4.

In such cases, and based on satisfactory supplier and risk assessments, a flexible approach to specification and verification may be applied. The number of deliverables required to cover the required specification and verification of functionality delivered via configuration will depend on the size, complexity, and technical architecture of the system (see Figure 4.3). For example, in many cases, functionality and configuration aspects may be combined into one specification, or the definition of configuration to meet defined requirements is all that is required.

In some cases, for example, complex on-premise solutions such as process control systems, but not typically for commercially available IT products or SaaS solutions, a project-specific functional specification and subsequent functional testing may be defined and agreed between the regulated company and the supplier.

The illustrative diagrams in this section show the project phase activities from the regulated company perspective. Reference to functional and other project specifications should not be confused with the supplier's internal core product functional specifications, technical, architectural, and design specifications that are maintained within the supplier QMS as part of their product life cycle information.

Figure 4.3: Approach for a Configured Product (Category 4)



Testing typically covers:

- Correct installation
- Configuration of the system

- Functionality that supports the specific business process based on risk and supplier assessments (this is an area where supplier activities may and should be leveraged as much as possible, see Section 8.3)
- Tests that demonstrate fitness for intended use and allow acceptance of the system against requirements
- Any further tests as a result of risk and supplier assessments

Regulated companies should decide upon the required levels of specification and verification, and many of the project phase activities and documents may be delegated. Since the system is configured for a business process, testing should be focused on this configuration.

Supplier activities typically include:

- Supply of the product
- Production of specifications and test specifications, as required, on behalf of the regulated company
- Support during configuration and testing
- User documentation
- Training
- Support and maintenance activities

The supply of the product and configuration may be performed by different suppliers.

Note that projects that are predominantly Category 4 may also require development of new custom software components, such as interfaces.

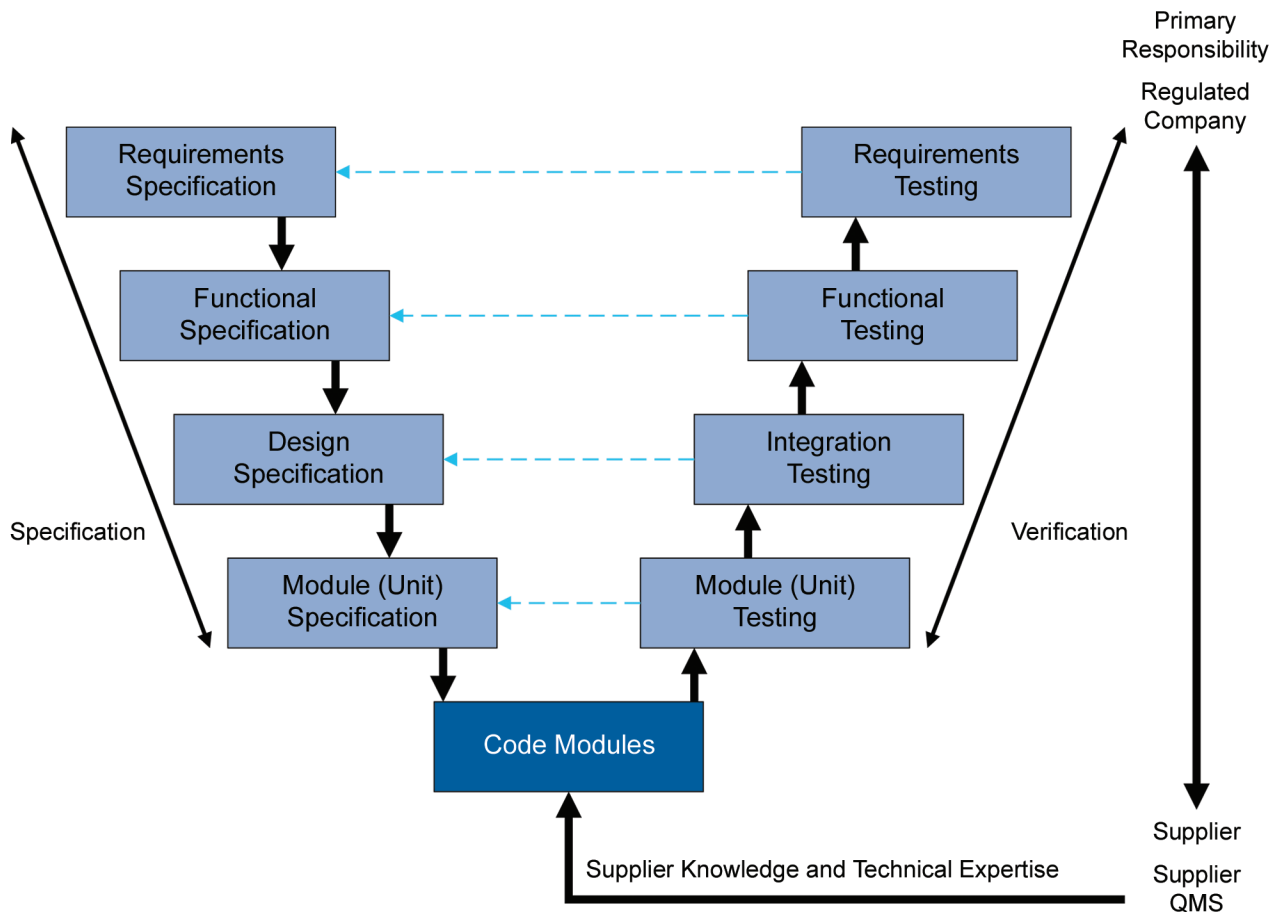
4.2.6.3 Example of a Custom Application

Some computerized systems are developed to meet individual user requirements, where no commercially available solution is suitable. These may be developed using a linear (waterfall) approach or by using iterative and incremental (Agile) methods. (See Figure 4.4)

The software developed for such systems is classified as GAMP Category 5.

In complex cases an approach consisting of various levels of specification and verification may be applied. The number of deliverables required to cover these levels will depend on the complexity and impact of the system. For example, for a small simple system the specification and design specifications may be combined. References to documentation and deliverables should not be interpreted as always requiring traditional documents, and the maintenance of records and information in appropriate and effective software tools is encouraged. For automated process equipment, computer system specification and verification should be part of an integrated engineering approach to ensure compliance and fitness for intended use.

Figure 4.4: Approach for a Custom Application (Category 5)



Testing typically covers:

- Correct installation
- Functionality and design
- Tests that demonstrate fitness for intended use and allow acceptance of the system against requirements
- Any further tests as a result of risk assessments and supplier assessments

Regulated companies should decide upon the required levels of specification and verification, and many of the project phase activities and deliverables may be delegated. Since the system is new, rigorous testing should be performed at both functional and design levels.

Supplier activities typically include production of specifications and test specifications on behalf of the regulated company, development of new software, testing, user documentation, training, and support and maintenance activities.

Complex systems may require a further hierarchy of specifications covering hardware design specifications and configuration specifications.

4.2.6.4 Terminology

The specific terminology used to describe life cycle activities and deliverables varies from company to company and from system type to system type. There are a number of reasons for this, including:

- Regulated companies having different approaches
- Difference in emphasis of GLP, GCP, GMP, GVP, and medical devices
- Difference in emphasis of various regulatory agencies
- Different international or local standards being followed
- Different types of computerized systems (e.g., IT, manufacturing, and laboratory systems)
- Suppliers using a range of different development models and approaches

This Guide aims to be flexible and does not intend to prescribe any one set of terms to the exclusion of others.

This section describes how qualification terminology, as traditionally used, relates to the activities described in this Guide. See Table 4.1. This will assist readers who use this terminology with the application of this Guide.

This Guide does not use traditional pharmaceutical process validation terminology such as IQ/OQ/PQ to describe system life cycle activities. This aligns with the US FDA, who also chose not to use the terminology in General Principles of Software Validation; Final Guidance for Industry and FDA Staff [33] because the terminology may not be well understood among many software professionals. This qualification terminology is also not well aligned with current good software engineering and IT/IS practice, or typical software supplier practices, and implies an inherently linear approach. Such terminology may, however, be appropriate for the qualification of, for example, simpler stand-alone manufacturing or laboratory equipment.

Whatever terminology is used for verification activity, the overriding requirement is that the regulated company can demonstrate that the system is compliant and fit for intended use.

Table 4.1: Relationship between Traditional Qualification Terminology and GAMP 5 Activities

Traditional Term	Description	GAMP 5 Verification Activity
Design Qualification (DQ)	Documented verification that the proposed design of facilities, systems, and equipment is suitable for the intended purpose	Design review (See Section 4.2.5 for further details)
Installation Qualification (IQ)	Documented verification that a system is installed according to written and approved specifications	Checking, testing, or other verification to demonstrate correct: <ul style="list-style-type: none"> • Installation of software and hardware • Configuration of software and hardware (See Appendix D5 for details)
Operational Qualification (OQ)	Documented verification that a system operates according to written and approved specifications throughout specified operating ranges	Testing or other verification of the system against specifications to demonstrate correct operation of functionality that supports the specific business process throughout all specified operating ranges. (See Appendix D5 for details)

Table 4.1: Relationship between Traditional Qualification Terminology and GAMP 5 Activities (continued)

Traditional Term	Description	GAMP 5 Verification Activity
Performance Qualification (PQ)	Documented verification that a system is capable of performing the activities of the processes it is required to perform, according to written and approved specifications, within the scope of the business process and operating environment	Testing or other verification of the system to demonstrate fitness for intended use and to allow acceptance of the system against specified requirements. (See Appendix D5 for details)

Note: The use of qualification terminology in relation to computerized systems and the relationship between OQ and PQ in particular, varies from company to company. The comparisons in Table 4.1 provide a general interpretation only and are not intended to be prescriptive.

Regulated companies should decide on a verification approach appropriate to a specific system. Testing activities should be selected based on the risk, complexity, and novelty of the system as described in Section 4.2.3.

The examples in Section 4.2.6 illustrate the typical different levels of testing applied to different categories of system, such as:

- Module testing
- Integration testing
- Configuration testing
- Functional testing
- Requirements testing

Table 4.1 lists various GAMP verification activities. There is no one-to-one relationship between the levels of testing and the GAMP verification activities, as the following examples show. Note that these examples are intended as illustrative and not prescriptive, and that critical thinking and technical experience and knowledge should be applied to the selection of the appropriate life cycle approach.

- For a typical Category 3 system, testing of both installation and configuration are covered by requirements testing.
- For a typical Category 3 system, tests are executed to demonstrate fitness for intended use and to allow acceptance of the system against user requirements. There is typically no need for further testing to demonstrate correct operation of standard functionality of the product.
- For a typical Category 4 system, while testing of configuration is covered by configuration testing, testing of installation may occur at any of the testing levels depending on the project.
- For a typical Category 5 system, correct operation of functionality that supports the specific business process may be covered by module testing, integration testing, and functional testing, and may be supplemented by pre-delivery testing.

The relationship between the required verification and the different levels of testing, particularly for GAMP Category 4 and GAMP Category 5 systems, may be complex. The verification or test strategy for a particular system should ensure that the required verification activities are adequately covered.

Acceptance of the system by the regulated company as being fit for release for operational use includes satisfactory completion of an agreed set of verification activities. For some systems, this may occur in stages including the leveraging, wherever possible, of testing or other acceptance activities performed prior to, and after, delivery. Commonly used terms within such a process include Factory Acceptance Testing, Site Acceptance Testing, and System Acceptance Testing. Verification activities should not be duplicated unnecessarily.

Whichever terms are used, the verification strategy should clearly define which activities should be satisfactorily completed to allow acceptance of the system for release into operational use by the regulated company.

See Appendix D5 for further guidance on aspects of testing.

4.3 Operation

This section provides guidance on system operation. Not all the activities will be directly relevant to all systems. The approach and required activities should be selected and scaled according to the nature, risk, and complexity of the system in question through the application of critical thinking.

For more detailed information on system operation topics see the *ISPE GAMP® Good Practice Guide: A Risk-Based Approach to Operation of GxP Computerized Systems* [34].

As part of preparing for final acceptance and formal handover for live operation, the regulated company should ensure that appropriate operational processes, procedures, and plans have been implemented, and are supported by appropriate training. These procedures and plans may involve the supplier in support and maintenance activities.

Depending on the deployment model and degree of outsourcing, the supplier (or service provider) may host and administer, as well as support, the system. Appropriate contracts and agreements are essential to define expectations, responsibilities, and performance measures.

Once the system has been accepted and released for use, there is a need to maintain compliance and fitness for intended use throughout its operational life. This is achieved by the use of established and up-to-date procedures and training that cover use, maintenance, and management.

The operational phase of a system may last many years, and may include changes to services, software, hardware, the business process, and regulatory requirements. The integrity of the system and its data should be maintained at all times and verified as part of periodic review. The use of supporting systems and tools to digitize and automate processes for efficiency, consistency, and process adherence is encouraged.

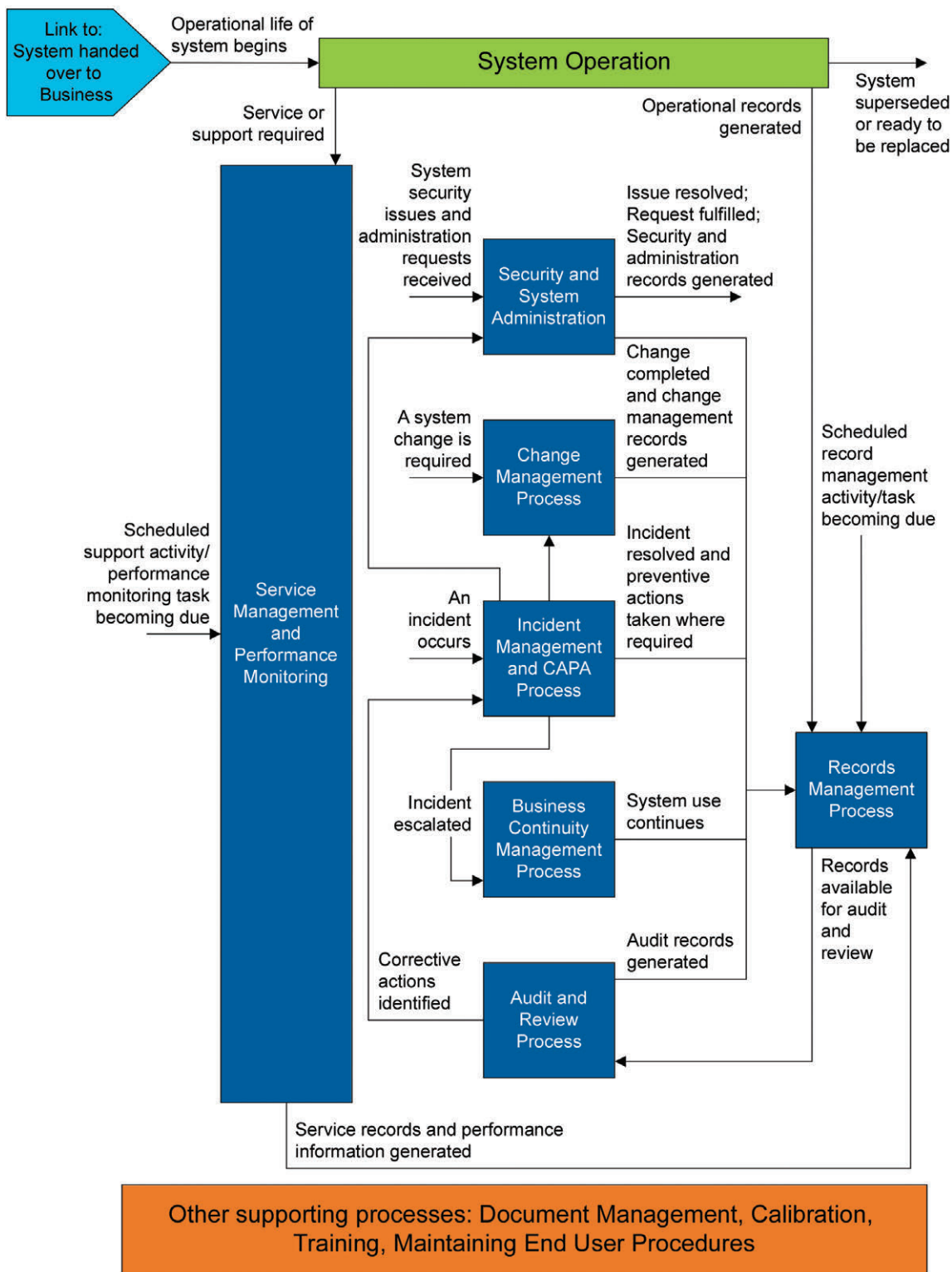
As experience is gained during operation, opportunities for process and system improvements should be sought based on periodic review and evaluation, operational and performance data, and root-cause analyses of failures. Information from the incident and problem management and Corrective and Preventive Action (CAPA) processes can provide significant input to the evaluation.

Change management should provide a dependable mechanism for prompt implementation of technically sound improvements following the approach to specification, design, and verification described in this Guide. The rigor of the approach, including the extent of documentation and verification, should be based on the risk and complexity of the change through the application of critical thinking.

Note that operational GxP computerized systems as discussed in this section are those that support internal regulated company GxP business processes, whereas operational SaMD is typically a product in the hands of patients or health care providers. In these cases further medical device-related requirements such as the need for post-market surveillance also apply.

Maintaining system compliance involves many interrelated activities. Figure 4.5 shows the major relationships between related groups of these activities. Service management and performance monitoring occur throughout the operational life of the system. Other activities, such as change management, occur when triggered.

Figure 4.5: Major Information Flows between Operational Activities



Some of the groups of activities shown in Figure 4.5 contain several individual related processes, procedures, and plans, as described in Table 4.2.

Table 4.2: Grouping of Operational Processes

Group of Processes	Process	Appendix
Handover	Handover	O1
Service Management and System Monitoring	Establishing and Managing Support Services System Monitoring	O2 O3
Incident and Problem Management, Deviation, and CAPA	Incident Management and Problem Management Corrective and Preventive Action	O4 O5
Configuration Management	Operational Change and Configuration Management	O6
Audits and Review	Periodic Review (Internal Quality Audits not covered by GAMP 5)	O8
Continuity Management	Backup and Restore Business Continuity Management	O9 O10
Security and System Administration	Security Management System Administration	O11 O12
Records Management	Archiving and Retrieval	O13

These processes are supported by QMS activities, such as document management, records management, knowledge management, training management, and the maintenance of up-to-date end user procedures. Further information is available in *ISPE GAMP® Guide: Records and Data Integrity* [35] and *ISPE Good Practice Guide: Knowledge Management in the Pharmaceutical Industry* [10]

The individual support and maintenance processes required to maintain the compliance of computerized systems during operation are briefly described below and covered in more detail in the Operational Appendices as shown in Table 4.2.

The use of good IT practices (e.g., ITIL [5]) supported by IT service management tools and automation is encouraged to ensure the efficiency and effectiveness of support processes and an auditable record of support activities.

4.3.1 Handover

Handover is the process for transfer of responsibility of a computerized system from a project team or a service group to the operations team or a new service group.

This is an important process; achieving compliance and fitness for intended use on its own may not be enough to guarantee a successful transfer into the operational phase.

The handover process will typically involve the project team (development group and/or supplier), process owner, system owner, and Quality Unit. The support group should be involved at the earliest opportunity to ensure effective knowledge transfer and establishment of operational procedures. A period of elevated support and maintenance, often referred to as Hypercare Services, may be arranged to facilitate the transfer.

Implied context-specific knowledge acquired through personal experience or internalization, as well as explicit knowledge captured and codified in documentation, tools, and systems should be considered. See *ISPE Good Practice Guide: Knowledge Management in the Pharmaceutical Industry* [10] for further details on knowledge management.

See Appendix O1 for further details.

4.3.2 Service Management and Performance Monitoring

Figure 4.5 shows the major relationships between operational processes. Service management and performance monitoring are shown related to records management due to records generated to demonstrate proper operation and performance of a system. In addition, there is potential interaction with incident and problem management and CAPA and change management when the results of the service or monitoring indicate there are issues that need addressing. For clarity, these interactions are not shown in Figure 4.5.

4.3.2.1 Establishing and Managing Support Services

The support required for each system, and how it will be provided, should be established. Support may be provided by external service providers or internal resources. This process should ensure that service agreements, maintenance plans, SOPs, support systems, and tools are established.

See Appendix O2 for further details.

4.3.2.2 Performance Monitoring

Performance monitoring detects issues that could impact the availability and performance of the system in order to facilitate mitigation before problems occur. Detected issues are managed through the incident management and problem management processes. Monitoring tools and automation are increasingly used to detect potential issues, report issues, and escalate to support organizations for timely intervention and rectification, and are encouraged.

The need for performance monitoring should be considered, and required activities scheduled and documented. This may change during the operational life of a system.

See Appendix O3 and Appendix D9 for further details.

4.3.3 Incident and Problem Management and CAPA

4.3.3.1 Incident Management and Problem Management

The incident management process aims to categorize incidents to direct them to the most appropriate resource or complementary process to achieve a timely resolution; whereas problem management involves analyzing root causes and preventing incidents from happening in the future. There should be a procedure defining how problems related to software, hardware, and procedures should be captured, reviewed, prioritized, progressed, escalated, and closed. This includes the need for processes to monitor progress and provide feedback. Incident and problem management processes may be supported by service management tools that support the incident and problem management process, associated action planning, and traceability of actions taken to address the incident or problem.

See Appendix O4 for further details.

4.3.3.2 Corrective and Preventive Action

CAPA is a process for investigating, understanding, and correcting discrepancies based on root-cause analysis, while attempting to prevent their recurrence.

In the operational environment the CAPA process should feed into the overall CAPA system used for GxP operations. When incidents occur, or when opportunities to reduce process/system failures are identified by other means, CAPA should be identified and processes established to ensure that these are implemented effectively. CAPA can provide a solution to problem control as described in ITIL Problem Management [5].

See Appendix O5 for further details.

4.3.4 Change Management

4.3.4.1 Change Management

Change management is a critical activity that is fundamental to maintaining the proper functioning and controlled status of systems and processes. All changes that are proposed during the operational phase of a computerized system, whether related to software (including middleware), hardware, infrastructure, or use of the system, should be subject to a formal change-control process (see Appendix O6 for guidance on replacements). This process should ensure that proposed changes are appropriately reviewed to assess impact and risk of implementing the change. Regression analysis and regression testing may be required. The process should ensure that changes are suitably evaluated, authorized, documented, tested, and approved before implementation, and subsequently closed.

The process should allow the rigor of the approach, including the extent of documentation and verification, to be scaled based on the nature, risk, and complexity of the change, by application of critical thinking. Some activities such as replacements and routine system administration tasks should be covered by appropriate repair or system administration processes.

Change management should provide a mechanism for prompt implementation of continual process and system improvements based on periodic review and evaluation, operational and performance data, and root-cause analysis of failures.

See Appendix O6 for further details.

4.3.4.2 Configuration Management

Configuration management includes those activities necessary to precisely define a computerized system at any point during its life cycle, from the initial steps of development through to retirement.

A configuration item is a component of the system that does not change as a result of the normal operation of the system. Configuration items should only be modified by application of a change management process. Formal procedures should be established to identify, define, and baseline configuration items, and to control and record modifications and releases of configuration items, including updates and patches.

See Appendix O6 for further details.

4.3.4.3 Repair Activity

The repair or replacement of defective computerized system components, which are often but not exclusively hardware or infrastructure related, should be managed in accordance with a defined process. Such activities should be authorized and implemented within the wider context of the change management process. Many repair activities are emergencies and require rapid resolution, so the incident and change management processes should be designed to allow such activities to occur without delay or increased risk to the operational integrity of the computerized system.

See Appendix O6 for further details.

4.3.5 Periodic Review

Periodic reviews are used throughout the operational life of systems to verify that they remain compliant with regulatory requirements, fit for intended use, and meet company policies and procedures, including those related to data integrity. The reviews should confirm that, for components of a system, the required support and maintenance processes and expected regulatory controls (plans, procedures, and records) are established.

Periodic reviews should be:

- Scheduled at an interval appropriate to the impact and operational history of the system. Risk and other assessments should be used to determine which systems are in scope and the frequency of periodic review.
- Performed in accordance with a predefined process
- Documented with corrective actions tracked to satisfactory completion

See Appendix O8 for further details.

4.3.6 Continuity Management

4.3.6.1 Backup and Restore

Processes and procedures should be established to ensure that backup copies of software, records, and data are made, maintained, and retained for a defined period within safe and secure areas.

Restore procedures should be established, tested, and the results of that testing documented.

See Appendix O9 for further details.

4.3.6.2 Business Continuity Planning

Business Continuity Planning (BCP) is a series of related activities and processes concerned with ensuring that an organization is fully prepared to respond effectively in the event of failures and disruptions, covering local and global infrastructure, data, and the application.

Critical business processes and systems supporting these processes should be identified, and the risks to each assessed. Plans should be established and exercised to ensure the timely and effective resumption of these critical business processes and systems.

A business continuity plan defines how the business may continue to function and handle data following failure. It also defines the steps required to restore business processes following a disruption and, where appropriate, how data generated during the disruption should be managed. Plans should be prioritized based on patient and business risk, in case of disruption to multiple systems.

The BCP also identifies the triggers for invoking the business continuity plan, roles and responsibilities, and required communication.

See Appendix O10 for further details.

4.3.6.3 Disaster Recovery Planning

As a subset of BCP, plans should be specified, approved, and rehearsed for the recovery of specific systems in the event of a disaster. These plans should detail the precautions taken to minimize the effects of a disaster, allowing the organization to either maintain or quickly resume critical functions. There should be a focus on disaster prevention, e.g., the provision of redundancy for critical systems. Disaster Recovery (DR) plans often require a shared responsibility model between internal organizations of the regulated company (e.g., business, IT) and external service providers (e.g., cloud service providers).

See Appendix O10 for further details.

4.3.7 Security and System Administration

4.3.7.1 Security Management

Computerized systems and data should be adequately protected against willful or accidental loss, damage, or unauthorized change. Procedures for managing secure access, including adding and removing privileges for authorized users, virus management, password management, and physical security measures should be established before the system is approved for use.

Role-based security should be implemented, if possible, to ensure that sensitive data and functions are not compromised. Security management procedures should apply to all users, including administrators, superusers, users, and support staff (including supplier support staff).

Security provisions should ensure that data is protected against unauthorized intrusions including cyber security attacks. Intrusion prevention and detection processes, supported by relevant IT tools and automation, should be in place.

See Appendix O11 for further details.

4.3.7.2 System Administration

Once a system is in operation the users of the system will require support. The system administration process provides administrative support for systems, including performance of standard administration tasks. The extent of this process varies greatly depending on the nature of the system.

See Appendix O12 for further details.

4.3.8 Record Management

4.3.8.1 Retention

Policies for the retention of regulated records should be established, based on a clear understanding of regulatory requirements and existing corporate policies, procedures, standards, and guidelines.

See Appendix O13 for further details.

4.3.8.2 Archiving and Retrieval

Archiving is the process of taking records and data off-line by moving them to a different location or system, often protecting them against further changes. Procedures and processes for archiving and effective and accurate retrieval of records should be established based on a clear understanding of regulatory requirements including retention periods.

See Appendix O13 for further details.

The *ISPE GAMP Guide: Records and Data Integrity* [35] and *ISPE GAMP® RDI Good Practice Guide: Data Integrity by Design* [36] give further details on record retention, archive, and retrieval.

4.4 Retirement

This activity covers system withdrawal, system decommissioning, system disposal, and migration of required data.

4.4.1 *Withdrawal*

Withdrawal is the removal of the system from active operation, i.e., users are deactivated, interfaces disabled. No data should be added to the system from this point forward. Special access should be retained for data reporting, results analysis, and support.

4.4.2 *Decommissioning*

Decommissioning is a controlled process by which an application or system is removed from use in an organization once it has been retired and/or has become obsolete.

4.4.3 *Disposal*

Data, documentation, software, or hardware may be permanently destroyed. Each may reach this stage at a different time. Data and documentation may not be disposed of until they have reached the end of the record-retention period, as specified in the company's record-retention policy, following an authorized and documented process.

Due to the volume of data and records involved, retirement can be a major task for IT systems in particular. Consideration should be given to:

- Establishing procedures covering system retirement, including withdrawal, decommissioning, and disposal as appropriate
- Documentary evidence to be retained of actions taken during retirement of the system
- GxP records to be maintained, their required retention periods, and which records can be destroyed
- The need to migrate records to a new system or archive, and the method of verifying and documenting this process
- Ability to retrieve these migrated records on the new system

Further guidance is also provided in the *ISPE GAMP Guide: Records and Data Integrity* [35] and *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36].

See Appendix M10 for further details.

4.4.4 *Data Migration*

Data migration may be required when an existing system is replaced by a new system, when an operational system experiences a significant change, or when the scope of use of a system is changed. The migration process should be accurate, complete, and verified.

See Appendix D7 for further details.

5 Quality Risk Management

Chapter 3 introduced the concept of QRM as part of the life cycle approach. This section gives an overview of the QRM process and Appendix M3 provides more detail.

This section is primarily aimed at new computerized systems. It does not imply that formal risk assessments are required for all existing systems. The extent of risk management required for existing systems, including the need for formal risk assessments, should be considered as part of periodic review.

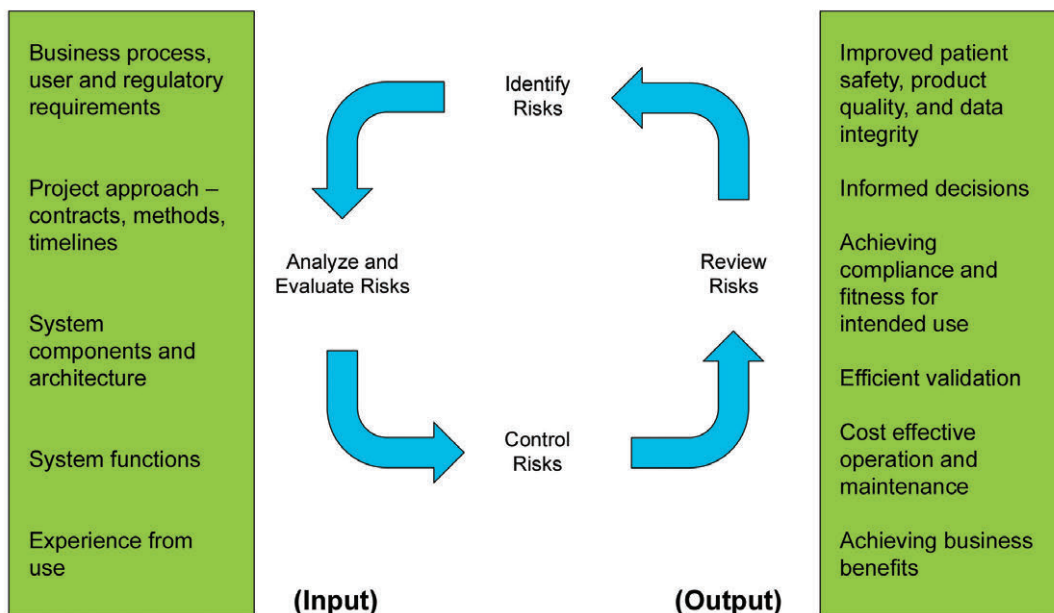
This section focuses on software products and custom applications rather than on infrastructure.

5.1 Overview

QRM is a systematic process for the assessment, control, communication, and review of risks. It is an iterative process used throughout the computerized system life cycle from concept to retirement.

Figure 5.1 indicates key areas for risk management and the benefits of the approach.

Figure 5.1: Overview and Benefits of Risk Management



For a given organization, a framework for making risk-management decisions should be defined to ensure consistency of application across systems and business functions. Terminology should be agreed upon, particularly regarding definitions and metrics for key risk factors.

Such a framework is most effectively implemented when it is incorporated into the overall QMS and is fully integrated with the system life cycle.

5.2 Science-Based Quality Risk Management

Determining the risks posed by a computerized system requires a common and shared understanding of:

- Impact of the computerized system on patient safety, product quality, and data integrity
- Supported business processes
- CQAs for systems that monitor or control CPPs
- User requirements
- Regulatory requirements
- Project approach (contracts, methods, timelines)
- System components and architecture
- System functions
- Supplier capability

The organization also should consider other applicable risks, such as Health, Safety, and Environment (HSE).

Managing the risks may be achieved by:

- Elimination by design
- Reduction to an acceptable level
- Verification to demonstrate that risks are managed to an acceptable level

It is desirable to eliminate risk, if possible, by modifying processes or system design. Design reviews can play a key role in eliminating risk by design.

Risks that cannot be eliminated by design should be reduced to an acceptable level by controls or manual procedures. Risk reduction includes applying controls to lower the severity, decrease probability, or increase detectability.

A systematic approach should be defined to verify that the risk associated with a system has been managed to an acceptable level. The overall extent of verification and the level of detail of documentation should be based on the risk to patient safety, product quality, and data integrity, and take into account the complexity and novelty of the system.

The information needed to perform risk assessments may become available, and should be considered, at different stages in the life cycle. For example, the high-level risks associated with a business process need to be understood before the risks associated with specific functions of computerized systems can be assessed.³

The criticality of a business process is independent of whether it is manually processed, semi-automated, or fully automated. Systems that support critical processes include those that:

- Generate, manipulate, or control data supporting regulatory safety and efficacy submissions

³ CQAs of drug development and manufacture will influence the understanding of the impact of the business process, while CPPs will influence the impact of specific computerized functions.

- Control critical parameters and data in pre-clinical, clinical, development, and manufacturing
- Control or provide data or information for product release
- Control data or information required in case of product recall
- Control adverse event or complaint recording or reporting
- Support pharmacovigilance

5.3 Quality Risk Management Process

The ICH Q9 [14] describes a systematic approach to QRM intended for general application within the pharmaceutical industry. It defines the following two primary principles of QRM:

- *“The evaluation of the risk to quality should be based on scientific knowledge and ultimately link to the protection of the patient; and*
- *The level of effort, formality and documentation of the quality risk-management process should be commensurate with the level of risk.”*

In the context of computerized systems, scientific knowledge is based upon the system specifications and the business process being supported.

This Guide uses the following key terms from ICH Q9 [14]:

“Harm: *Damage to health, including the damage that can occur from loss of product quality or availability.”*

“Hazard: *The potential source of harm.”*

“Risk: *The combination of the probability of occurrence of harm and the severity of that harm.”*

“Severity: *A measure of the possible consequences of a hazard.”*

This Guide applies the general principles of ICH Q9 [14] to describe a five-step process for risk management as an integral part of achieving and maintaining system compliance. For simple or low-risk systems, some of these steps may be combined. See Appendix M3 for further details on the QRM process.

This process is focused on managing risks during the project phase. Risk management should also be used appropriately both within specific activities and during the operational phase. Examples include:

1. Determining the need for supplier audit as part of supplier assessment
2. Determining the rigor and extent of testing
3. Determining corrective actions arising from test failures
4. Determining the impact of proposed changes as part of change management
5. Determining the frequency of periodic reviews

Application of risk management to the above activities is covered in the appropriate sections of this Guide.

Organizations may have established risk-management processes, including the use of methods such as those listed in Appendix M3. While this Guide describes one suggested approach, it does not intend or imply that these existing methods should be discarded, rather that they continue to be used, as appropriate, within the context of an overall QRM framework consistent with ICH Q9 [14].

Process Risk Assessment

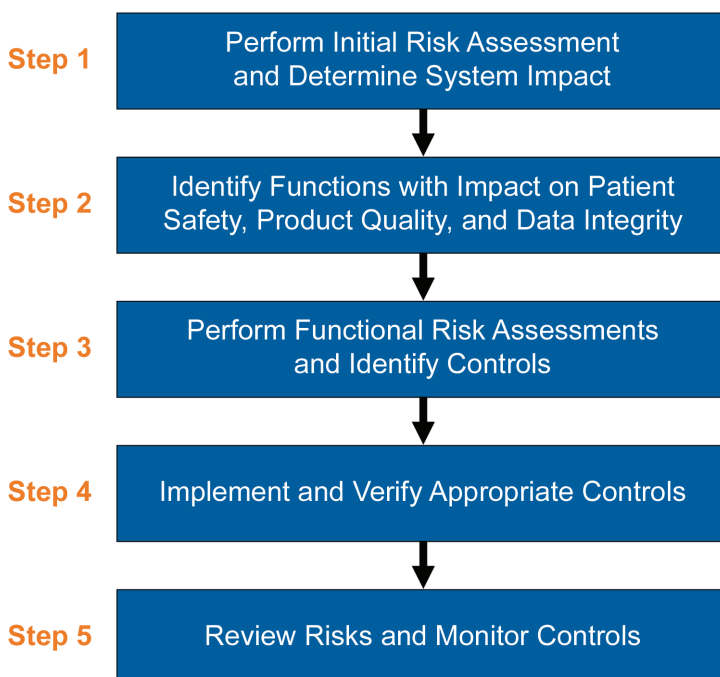
Some records and data may reside on more than one system during their life cycle, and QRM activities should start at the business process level, at a level higher than individual systems. A process risk assessment (also known as business process risk assessment) is a non-system-specific high-level assessment of the business process or data flow, which may occur before system-specific QRM activities. An equivalent risk assessment from a data flow (rather than business process flow) perspective may be performed, using the same approaches and techniques, and with the same benefits.

The process risk assessment is aimed at identifying key high-level risks to patient safety, product quality, and data integrity, and identifying the required controls to manage those risks. Typically, at this stage no assumptions are made about the nature or exact functionality and design of the computerized system(s) that will support the process.

The process risk assessment provides valuable input to subsequent QRM activities. Typical inputs to the process risk assessment include:

- Defined business process scope
- Process descriptions and/or diagrams
- Identified regulatory requirements for the proposed process scope
- Identified company quality requirements

Figure 5.2: Quality Risk-Management Process



Step 1 – Perform Initial Risk Assessment and Determine System Impact

An initial risk assessment should be performed based on an understanding of business processes and business risk assessments, user requirements, regulatory requirements, and known functional areas. A system cannot have a higher impact than the business process it supports. Any relevant previous assessments may provide useful input, and these should not be repeated unnecessarily.

The results of this initial risk assessment should include a decision on whether the system is GxP regulated (i.e., GxP assessment). It also should include an overall assessment of system impact. The scope and objectives of any further risk assessments should be defined.

Based on this initial risk assessment and resulting system impact, it may not be necessary to perform the subsequent steps of the process, as the level of risk may already be at an acceptable level.

The specific level of effort, formality, and documentation of any subsequent steps should be determined based on level of risk and system impact. See Appendix M3 for further details.

If relevant, regulated electronic records and signatures should be identified. Again, existing assessments may provide useful input and should not be repeated. A detailed approach and specific guidance is provided in the *ISPE GAMP Guide: Records and Data Integrity* [35]

Step 2 – Identify Functions with Impact on Patient Safety, Product Quality, and Data Integrity

Functions that have an impact on patient safety, product quality, and data integrity should be identified by building on information gathered during Step 1, referring to relevant specifications, and considering project approach, system architecture, and categorization of system components. Individual functions cannot have a higher impact than the system as a whole.

Step 3 – Perform Functional Risk Assessments and Identify Controls

Functions identified during Step 2 should be assessed by considering possible hazards, and how the potential harm arising from these hazards may be controlled.

It may be necessary to perform a more detailed assessment that analyzes further the severity of harm, likelihood of occurrence, and probability of detection. See Appendix M3 – Section 11.5 for an example detailed assessment process.

The judgment as to whether to perform a detailed assessment for specific functions should be dealt with on a case-by-case basis and the criteria can vary widely. The criteria to be considered include:

- Criticality of the supported process
- Specific impact of the function within the process
- Nature of the system (e.g., complexity and novelty)

Appropriate controls should be identified based on the assessment. A range of options is available to provide the required control depending on the identified risk. These include, but are not limited to:

- Modification of process design
- Modification of system design
- Application of external procedures

- Increasing the detail or formality of specifications
- Increasing the number and level of detail of design reviews
- Increasing the extent or rigor of verification activities

Where possible, elimination of risk by design is the preferred approach.

Step 4 – Implement and Verify Appropriate Controls

The control measures identified in Step 3 should be implemented and verified to ensure that they have been successfully implemented. Controls should be traceable to the relevant identified risks.

The verification activity should demonstrate that the controls are effective in performing the required risk reduction.

The effort, formality, and documentation of the verification activity should be commensurate with the level of risk.

Step 5 – Review Risks and Monitor Controls

During periodic review of systems, or at other defined points, an organization should review the risks. The review should verify that controls are still effective, with corrective action taken under change management if deficiencies are found. The organization also should consider whether:

- Previously unrecognized hazards are present
- Previously identified hazards are no longer applicable
- The estimated risk associated with a hazard is no longer acceptable
- The original assessment is otherwise invalidated (e.g., following changes to applicable regulations or change of system use)

Where necessary, the results of the evaluation should be fed back into the risk-management process. If there is a potential that the residual risk or its acceptability has changed, the impact on previously implemented risk control measures should be considered, and the results of the evaluation documented. It should be noted that some changes may justify relaxation of existing controls.

The frequency and extent of any periodic review should be based on the level of risk and should consider previous findings and operational history.

6 Regulated Company Activities

Responsibility for the compliance of computerized systems lies with the regulated company. This involves activities at both the organizational level and at the level of individual systems.

Therefore, this section is divided into:

- Governance for Achieving Compliance
- System-Specific Activities

6.1 Governance for Achieving Compliance

Achieving robust, cost-effective compliance requires strong governance. Key elements of successful governance include:

- Establishing computerized systems compliance policies and procedures
- Identifying clear roles and responsibilities
- Training
- Managing supplier relationships
- Maintaining a system inventory
- Planning for validation
- Continual improvement activities
- Data governance

Effective governance is achieved by integrating these activities into the management of the organization. Each activity is described further in the following subsections.

6.1.1 *Computerized Systems Policies and Procedures*

Regulated companies should have a defined policy for ensuring that computerized systems are compliant and fit for intended use. The policy should typically include a commitment to:

- Identify and comply with all applicable GxP requirements
- Integrate life cycle activities into the regulated company's QMS
- Identify and assess each system
- Ensure GxP regulated systems are compliant and fit for intended use according to established SOPs
- Follow a validation framework, including the use of validation plans and validation reports as necessary
- Maintain compliance throughout the life of a system

Further details should be documented, e.g., in more detailed policies or in SOPs, which may be supplemented by guidance and templates. These documents will typically address:

- Maintaining the system inventory
- Determining the impact of systems on patient safety, product quality, and data integrity
- Defining roles and responsibilities
- The computerized system life cycle approach
- Planning, supplier assessment, risk management, specification, verification and reporting activities, and documents
- System operation and management, including up-to-date operating procedures for end users and administrators, and all operational processes described in Section 4.3
- Record and data management
- Security management

Policies and procedures should be developed taking into account existing policies, procedures, and practices.

6.1.2 Identifying Clear Roles and Responsibilities

Roles and responsibilities for activities should be documented, allocated, and communicated.

The roles of process owner, system owner, data owner, quality unit, SME, end user, and supplier are particularly important and are covered separately and in more detail in Section 6.2.3. The appropriate and timely involvement of these key roles should be ensured.

Key responsibilities include:

- Defining, approving, and maintaining policies and SOPs
- Compiling and prioritizing the system inventory
- Producing plans and reports
- Managing compliance and validation activities
- Maintaining compliance during operation

6.1.3 Training

Training is the process that ensures that persons who develop, validate, maintain, support, or use computerized systems have the education, training, and experience to perform their assigned tasks.

Procedures for training covering responsibilities, plans, and records should be established. The process owner is typically responsible for ensuring that all users are adequately trained; however, maintenance staff may be the responsibility of the system owner, and development staff may be the responsibility of a project manager.

Persons in responsible positions should have the appropriate training for the management and use of computerized systems within their field of responsibility. This should include specification, verification, installation, and operation of computerized systems.

All users and support staff of a GxP regulated system, including contracted staff, should be given appropriate training, including basic GxP training. They also should be given specific training covering regulatory aspects of using the computerized system, e.g., security aspects, or the use of electronic signatures.

For computerized systems, the regulated company should therefore:

- Establish the necessary training needs, including users, suppliers, data centers, IT departments, engineering, maintenance
- Provide training to satisfy these needs
- Evaluate the effectiveness of the training
- Ensure that staff are aware of the relevance and importance of their activities, e.g., GxP
- Ensure that supplier staff are adequately trained, e.g., as part of supplier assessment
- Maintain appropriate training records
- Ensure training is maintained up-to-date, e.g., following system changes

A risk-based approach should be used to determine the rigor of training required, measuring the effectiveness of training, the frequency of training, and the approach to retention of training records.

6.1.4 *Managing Supplier Relationships*

All phases of the computerized systems' life cycle require cooperation between the regulated company and external and internal suppliers, including IT and engineering. Both regulated companies and suppliers have important roles to play in ensuring that suitable computerized systems are deployed as part of regulated activities. Responsibility for activities may be with the suppliers, but in all cases regulatory accountability lies with the regulated company.

The regulated company must have defined roles and responsibilities for acceptance and release of GxP computerized systems. When outsourcing or delegating activities, there should be no resultant decrease in product quality, process control, or quality assurance. There should be no increase in the overall risk of GxP processes. The competence and reliability of service providers must be ensured.

Even though regulated companies cannot delegate their regulatory accountabilities to a supplier, they may leverage the knowledge, experience, activities, and artifacts of an IT/IS service provider through risk-based assessment, management, and governance processes.

Regulated companies should ensure that internal and external suppliers are made aware of the need for regulatory compliance on the part of the regulated company. The regulated company should ensure that the supplier has an understanding of the regulations that their customers must comply with, and how the supplier can help in achieving this compliance. The regulated company should verify, prior to contract placement, that the supplier has adequate expertise and resources to support user requirements and expectations. The most common mechanism for this is the supplier assessment, which may include an audit depending on risk, complexity, and novelty.

It should be noted that some suppliers, e.g., suppliers of commercially available software products or systems, will have fulfilled a significant part of their responsibilities before any relationship is established with individual regulated companies, and that this will have a major influence on any ensuing cooperation.

The regulatory expectation is that systems are fit for intended use and maintained in a state of control and compliance. Activities performed by, and information maintained by, suppliers following their own methods and approaches, and under their own QMS, may assist in achieving this objective.

Supplier activities are covered in Chapter 7.

6.1.5 Maintaining the System Inventory

Regulated companies should maintain an inventory of computerized systems, showing those that are GxP regulated (see Section 5.3). The inventory should provide summary information such as the validation status, ownership, impact, current system version, and supplier. Automated equipment may be listed separately and duplication should be avoided.

The inventory should be at the level of systems that support business processes, rather than individual items of hardware, such as keyboards and routers that would be covered by local IT procedures.

The system inventory may be used for planning periodic reviews.

6.1.6 Planning for Validation

Computerized system validation within a business unit is typically performed using a hierarchical framework of plans covering GxP regulated computerized systems.

Computerized system validation plans describe how to ensure compliance and fitness for intended use of specific systems. They specify scope, approach, resources, roles and responsibilities, and the types and extent of activities, tasks, and deliverables.

See Section 3.3 for further details on the computerized system validation framework. See Appendix M1 for more detailed guidance.

6.1.7 Continual Improvement Activities

Improving the processes used to achieve and maintain compliance and fitness for intended use and making them more effective and efficient is highly desirable. The risk of non-compliance is also reduced.

Suppliers also benefit from improving their processes for product development and support, and for other services provided (see Section 7.3).

Achieving improvements depends on an understanding of the effectiveness of the current processes and obtaining relevant and objective measures of the quality of both the process and the product.

6.1.7.1 Understanding

Understanding the effectiveness of the current processes is best gained by considering current levels of conformance to the processes (e.g., established by audit and trending performance), by reviewing current processes against recognized good practices, and applying critical thinking. This understanding should assist with identifying areas of the QMS that may require improvement. For example, persistent non-conformities in a particular process may be caused by problems within the process. Alternatively, a review of current processes may find scope for streamlining these processes based on developments in recognized good practice.

6.1.7.2 Metrics

Metrics may be gathered throughout the system life cycle, including:

- Design and development metrics (e.g., from design and code reviews)
- Testing metrics (e.g., from analysis of test failures and resulting actions)
- Operation and maintenance metrics (e.g., from incident management, change management, backup and restore)

Metrics should be collected only for a clear purpose. They typically provide information on one aspect of the operation of the QMS, and may assist with determining improvements to the QMS or to its use.

6.1.8 Data Governance

Data governance may be defined as [37]):

“The arrangements to ensure that data, irrespective of the format in which they are generated, are recorded, processed, retained and used to ensure the record throughout the data lifecycle.”

Data governance ensures formal management of records and data throughout the regulated company. It encompasses the people, processes, and technology required to achieve consistent, accurate, and effective data handling. Data governance provides the structure within which appropriate decisions regarding data-related matters may be made according to agreed models, principles, processes, and defined authority. For further details see *ISPE GAMP Guide: Records and Data Integrity*, Chapter 3: Data Governance Framework [35].

6.2 System-Specific Activities

Table 6.1 shows the typical regulated company activities required for a configurable computerized system.

Note: This table is indicative only. Activities required for a specific system should be determined based on risk, complexity, and novelty, by applying critical thinking.

This section provides further details on each task.

Table 6.1: Typical Activities for a Configurable Computerized System

Step	Task	Description
1.	Identify Compliance Standards	Compliance activities should be performed in accordance with applicable company policies and procedures.
2.	Identify System	The system should be added to an inventory of systems in accordance with documented procedures.
3.	Identify Key Individuals	These include Process Owner, System Owner, Quality Unit, SME, Supplier, End User.
4.	Produce Requirements Specification (RS)	The RS should define clearly and precisely what the regulated company wants the system to do, state any constraints, and define regulatory and documentation requirements.
5.	Determine Strategy for Achieving Compliance <ul style="list-style-type: none"> • Risk Assessment • Assessment of System Components • Supplier Assessment 	<p>An initial risk assessment should be performed during planning. Depending on the system, including GxP impact, complexity, and novelty, further assessments may be required as specifications are developed.</p> <p>System components should be assessed to determine the approach required, taking into account architecture, complexity, and novelty, including maturity and level of configuration or customization.</p> <p>The quality capability of a supplier should be formally assessed as part of the process of selecting a supplier and planning for achieving compliance. The decision whether to perform a supplier audit should be documented and based on a risk assessment and categorization of the system components.</p>

Table 6.1: Typical Activities for a Configurable Computerized System (continued)

Step	Task	Description
6.	Plan	Activities including risk assessments, deliverables, procedures, and responsibilities for establishing the adequacy of the system should be defined in a plan.
7.	Review and Approve Specifications	The regulated company should review and approve specifications, as appropriate. This could involve one or several specifications depending on the system. Design reviews may be used where appropriate.
8.	Develop Test Strategy	The regulated company should determine what testing is required after considering the existing documentation available. Depth and rigor of testing should consider intended use, impact, and risk.
9.	Test	The regulated company should ensure that testing defined in the test strategy is completed and ensure review of test results.
10.	Report and Release	A report should provide evidence that all planned deliverables and activities have been completed and that the system is fit for intended use. Any deviations, or outstanding or corrective actions, and any associated risk, should be managed and justified, and followed up as required by the regulated company. There should be a formal process covering release of the system for operational use by end users.
11.	Maintain System Compliance during Operation	The regulated company should establish adequate system management and operational procedures. See Section 4.3 for further details.
12.	System Retirement	The regulated company should manage the withdrawal of the computerized system from use, including migration of data to a new system, if applicable.

Table 6.1 shows typical regulated company activities for a configurable system. For a custom system, and based on risk, there may be increased levels of specification, review, and testing required.

6.2.1 Identify Compliance Standards

Compliance and fitness for intended use should be achieved in accordance with applicable company policies and procedures. National and international standards may be referenced. Industry guidance, such as ISPE GAMP guidance [38], may also be used as supporting information, in the context of appropriate company policies and procedures.

6.2.2 Identify System

The system should be assessed to determine whether it is GxP regulated and added to the system inventory in accordance with documented procedures (see Section 6.1.5).

6.2.3 Identify Key Individuals

This section describes key roles and responsibilities when achieving compliance. Designated individuals should have sufficient experience and training to perform their respective roles.

Specific activities may be delegated to appropriate representatives.

6.2.3.1 *Process Owner*

This is the owner of the business process or processes being managed. The process owner is ultimately responsible for ensuring that the computerized system and its operation are in compliance and fit for intended use in accordance with applicable SOPs. The process owner also may be the system owner. The process owner may be the de facto owner of the data residing on the system (data owner) and therefore, ultimately responsible for the integrity of the data. Process owners are typically the head of the functional unit using the system.

Specific activities may include:

- Approving key documentation as defined by plans and SOPs
- Providing adequate resources (personnel, including SMEs, and financial) to support development and operation of the system
- Ensuring adequate training for end users
- Ensuring that SOPs required for the operation of the system exist, are followed, and are reviewed periodically
- Ensuring changes are approved and managed
- Reviewing assessment/audit reports, responding to findings, and taking appropriate actions to ensure GxP compliance
- Coordinating input from other groups (e.g., finance, information security, HSE, legal)

6.2.3.2 *System Owner*

The system owner is responsible for the availability, and support and maintenance of a system, and for the security of the data residing on that system. The system owner is responsible for ensuring that the computerized system is supported and maintained in accordance with applicable SOPs. The system owner also may be the process owner (e.g., for IT infrastructure systems or systems not directly supporting GxP). For systems supporting regulated processes and maintaining regulated data and records, the ownership of the data resides with the GxP process owner, not the system owner.

The system owner acts on behalf of the users. For larger systems, the system owner will typically be from the IT or engineering functions. Global IT systems may have a global system owner and a local system owner to manage local implementation.

Specific activities may include:

- Approving key documentation as defined by plans and SOPs
- Ensuring that SOPs required for maintenance of the system exist, are followed, and are reviewed periodically
- Ensuring adequate training for maintenance and support staff
- Ensuring changes are managed
- Ensuring the availability of information for the system inventory and configuration management
- Providing adequate resources (personnel, including SMEs, and financial) to support the system
- Reviewing audit reports, responding to findings, and taking appropriate actions to ensure GxP compliance

- Coordinating input from other groups (e.g., finance, information security, HSE, legal)
- Managing the system life cycle, including system upgrade and replacement planning
- Ensuring that systems are supported and maintained such that they are fit for the intended business use, and support data integrity
- Ensuring that data integrity risks are identified and controlled to acceptable levels

6.2.3.3 Quality Unit

The term quality unit is used here as an encompassing term that includes many quality-related roles that are important to developing and managing regulated computerized systems. The manner in which a quality unit addresses the responsibilities noted may vary based on the applicable regulations. For example, the strict interpretation of the GLP requirement for separation of duties may lead some companies to interpret this as requiring that the GLP quality unit audit a validation document rather than approve it. Regardless of the mechanism, the intent of achieving acceptance from the quality unit is the same.

The quality unit has a key role to play in successfully planning and managing the compliance and fitness for intended use of computerized systems, and provides an independent role in:

- Approving or auditing key documentation such as policies, procedures, acceptance criteria, plans, and reports
- Focusing on quality critical aspects
- Involvement of SMEs
- Approving changes that potentially affect patient safety, product quality, or data integrity
- Auditing processes and supporting documentary evidence to verify that compliance activities are effective

The role may be split into corporate and operational quality depending on the organization. The following example is indicative only and titles and details of responsibility may vary between organizations.

6.2.3.4 Corporate Quality

This group operates at the corporate level and is responsible for:

- Setting policy
- Maintaining oversight of company standards
- Auditing for compliance
- Reviewing effectiveness of quality systems and processes

Regulatory authorities require the corporate quality unit to be independent of the business activities.

6.2.3.5 Operational Quality

This, typically, is the quality unit of a division or business unit. This group is involved in the compliance of GxP regulated computerized systems within the division or business unit and typically covers:

- Implementing quality standards and procedures for the development and operation of computerized systems

- Reviewing risk assessment and control activities
- Supporting project phase activities as defined in computerized system validation plans
- Supporting life cycle processes such as change control and document management
- Training related to computerized systems quality, compliance, and data integrity
- Agreeing on the approach to managing deviations with approval of any supporting rationales
- Managing the quality of external service and application providers (e.g., contractors, outsourcing organizations, etc.)

Specific operational quality activities can be delegated to a variety of functions provided that independence can be demonstrated. For example:

- IT departments may have their own quality management function
- Engineering departments may have their own standards groups
- Suppliers and consultants may be authorized to assist

Any such delegation should be clearly defined, and their respective roles agreed and documented as part of planning. In all circumstances, the quality unit retains ultimate responsibility and accountability for compliance with regulations.

6.2.3.6 Subject Matter Expert

Qualified and experienced SMEs have a key role in performing reviews and assessments, and taking technical decisions, based on science-based process and product understanding, and sound engineering principles. Different SMEs may be involved with different activities, e.g., during specification and verification.

SMEs should take the lead role in the verification of the computerized system. Responsibilities include planning and defining verification strategies, defining acceptance criteria, selection of appropriate test methods, execution of verification tests, and reviewing results.

SMEs may come from a wide range of backgrounds as required, including business process, engineering, IT, supplier, quality, and validation.

6.2.3.7 Supplier

Suppliers (including internal suppliers such as IT or engineering) play an important support role in achieving and maintaining system compliance and fitness for intended use. Specific activities may include:

- Provisioning existing documentation (see Section 8.3)
- Preparing and reviewing documentation
- Acting as SME for technical aspects such as configuration and design options
- Performing and supporting testing
- Managing changes and configuration

- Supporting processes geared toward maintaining system compliance, e.g., by providing software patches, second tier support for problem resolution processes, etc.

See Chapter 7 for further details on supplier activities.

6.2.3.8 End User

In addition to using the system in accordance with approved procedures, end users may be involved in the following activities throughout the life cycle:

- Providing input to user requirements and specifications
- Evaluating prototypes
- Testing
- Acceptance of system and handover
- Providing input to the development of SOPs for system use
- Reporting defects
- Identifying opportunities for improvement

6.2.4 Requirements Specification

The RS describes what the system should do. The RS is the responsibility of the regulated company but may be written by a third party or supplier. It should be adequately reviewed by SMEs and approved by the process owner. Other approvers may include the system owner and quality unit representative.

See Appendix D1 for further details on RS.

6.2.5 Determine Strategy for Achieving Compliance and Fitness for Intended Use

6.2.5.1 Risk Assessment

An initial risk assessment should be performed during planning to determine whether the system is GxP regulated, the impact of the system, and the need for further risk assessments. This process is described in Chapter 5.

See Appendix M3 for further details on risk assessment.

6.2.5.2 Assessment of System Components

The process of assessing system components applies the GAMP software categories and hardware categories as input to establishing the required activities, based on how the system is constructed or configured. This should take into account architecture, complexity, and novelty, including maturity and level of configuration or customization. Categorization should, however, be regarded as only part of the process of defining the required life cycle strategy based on critical thinking.

See Appendix M4 for further details on categories of software and hardware.

6.2.5.3 Supplier Assessment and Education

The regulated company should formally assess each supplier to establish their quality capability. This is typically performed by an SME and may involve an audit of the supplier depending on risk, complexity, and novelty. The assessment may find that a supplier has either a well-established, formal QMS, or has attained a recognized third-party certification such as ISO 9001 [39]. The strategy should take account of assessment conclusions. Other independent third-party assessment reports may also be relevant and useful. Examples include SOC 2® reports [40], which assess security, availability, and processing integrity of the systems that service organizations use to process users' data, and the confidentiality and privacy of the information processed by these systems.

If another regulated company has already assessed the supplier for the same reason, then subject to that company agreeing to share that information, an additional assessment may not be necessary. The justification for not assessing a specific supplier should be formally documented.

Regulated companies should be prepared to assist in the education and training of suppliers, either by direct involvement or by providing advice on training requirements, sources of information, and sources of specialist training and education, such as ISPE [41]. It may be beneficial to supply example documents, where possible, to establish the correct content and level of detail in the key documents.

See Appendix M2 for further details on supplier assessment.

6.2.6 Planning

Planning is an essential activity for any system development and should address all aspects, including activities that demonstrate compliance and fitness for intended use. Responsibilities, deliverables, and procedures to be followed should be defined. Since the supplier may provide deliverables or directly support these activities, planning provides the opportunity to decide how best to leverage supplier activities and documentation to avoid unnecessary duplication.

See Appendix M1 and Section 3.3 for further details on validation planning.

6.2.7 System Specifications

There are a number of types of specifications that may be required to adequately define a system. These may include functional specifications, configuration specifications, and design specifications. Particularly in the case of Agile development methods, the types and naming of deliverables may differ from these more traditional examples. See also Appendix D8. Specification information may be maintained in tools rather than traditional documents. See also Appendix D9.

The applicability of, and need for, these different specifications depends upon the specific system and should be defined during planning. See Section 4.2.6 for typical examples of the level of specification required for standard products, configured products, and custom applications.

Project or individual customer-system-specific functional specifications (as opposed to core product functional specifications) are normally written by the supplier and describe the detailed functions of the system, i.e., what the system will do to meet the requirements. The regulated company should review and approve such functional specifications when they are produced for a custom application or configured product. In this situation, they are often considered to be a contractual document.

See Appendix D1 for further details on functional specifications.

Configuration specifications are used to define the required configuration of one or more software packages that comprise the system. The regulated company should review and approve configuration specifications and have access to the information contained in them.

Design specifications for custom systems should contain sufficient detail to enable the system to be built and maintained. In some cases, the design requirements can be included in the functional specification. Relevant technical SMEs should take a lead role in the review and acceptance of design specifications.

See Appendix D3 for further details on configuration and design.

A current system description should be available for regulatory inspection and training. This may be covered by the requirements or functional specifications, or by a separate deliverable.

See Appendix D6 for further details on system descriptions.

6.2.7.1 Design Reviews

Design reviews evaluate deliverables against standards and requirements, identify issues, and where applicable propose required corrective actions. Design reviews assist with ensuring that computerized systems are fit for intended use, and that proposed functionality is appropriate, consistent, and meets defined standards.

They are planned and systematic reviews of specifications, design, and development, and should be planned to occur at suitable stages during the life cycle. They are an important part of the verification process.

Design reviews should be performed by SMEs and involve others as required.

For standard products (GAMP Category 3), design reviews by the regulated companies are not typically required. For configured products (GAMP Category 4), design review activities by regulated companies should focus on the configuration and any customization activities to meet user requirements. For complex custom applications (GAMP Category 5), design reviews may be conducted at various levels of specification.

Supplier development activities, including reviews, should be verified during the supplier assessment.

See Appendix M5 for further details on design review.

6.2.8 Development and Review of Software for Custom Applications

Code reviews by the regulated company are not required for standard and configurable software products. Custom applications and custom software for configured products (e.g., interfaces, macros, and report generation) should be developed in accordance with defined standards.

The need for, and extent of, reviews of new software during development should be based on risk, complexity, and novelty. Such reviews should be performed by an appropriate SME, typically from the organization developing the software. The regulated company should ensure that corrective actions resulting from such reviews are tracked to satisfactory completion.

See Appendix D4 for further details on management, development, and review of software.

6.2.9 Test Strategy and Testing

Section 4.2.3 describes the use of testing as a fundamental part of verification activity, including the development of a test strategy.

The regulated company is responsible for ensuring that the test strategy will demonstrate compliance and fitness for intended use. The number and types of tests should be based on risk, complexity, and novelty as described in Section 4.2.3. The role of the supplier, including use of existing supplier documentation, should be considered when developing the strategy.

The results of testing should be documented against defined acceptance criteria based on specifications. Test failures should be captured, reviewed, documented, and managed.

See Appendix D5 for further details on testing of computerized systems. See Section 8.5 for details on efficient testing practices.

6.2.10 Reporting and Release

At the conclusion of the project, a computerized system validation report should be produced summarizing the activities performed, any deviations from the plan, any outstanding and corrective actions, and providing a statement of fitness for intended use of the system. See Appendix M7 for further details.

In some cases, specific computerized system validation reports may not be required (see Section 3.3).

Release of the system into the operating environment in accordance with a controlled and documented process is discussed in Section 4.2.4.

6.2.11 Maintaining System Compliance During Operation

The regulated company is responsible for maintaining system compliance during operation (see Section 4.3).

6.2.12 System Retirement

System retirement is described in Section 4.4.

7 Supplier Activities

Although the responsibility for compliance with GxP regulations lies with the regulated company, the supplier may have considerable involvement in the process (see Section 6.1.4).

Regulated companies may be able to leverage supplier knowledge and documentation, subject to suitability following formal assessment. This may involve an audit, depending on risk, complexity, and novelty.

This section is written specifically to help suppliers to meet the requirements and expectations of the regulated company. Some information from previous sections is included to give suppliers a more complete picture.

7.1 Supplier Products, Applications, and Services

Suppliers provide a range of products, applications, and services for hardware, software, and related technologies including the provision of cloud-computing services. The relationship between supplier and regulated company will vary significantly depending upon the product, application, or scope of service being provided. Consultants, for instance those acting as implementation partners, should have sufficient education, training, and experience to advise on the subject for which they are retained.

7.1.1 *Standard Product (GAMP Category 3)*

If the product is purchased off-the-shelf and does not require configuration to support business processes, or only offers defined ranges of factory-provided values or ranges⁴, supplier involvement with the regulated company is, typically, limited to the provision of documentation, training, support, and maintenance. The product should be developed and maintained by the supplier in accordance with good practices (see Section 7.2).

7.1.2 *Configured Product (GAMP Category 4)*

If the product requires configuration to support specific business processes, supplier involvement with the regulated company will, typically, include support with specification, configuration, verification, and operation of the system (see Chapter 4).

Procedures to follow should be agreed between the regulated company and the supplier and be documented in the appropriate plan. Procedures adopted may be those of the regulated company or from the supplier QMS (see Section 7.2).

The product itself should be developed and maintained by the supplier in accordance with good practices (see Section 7.2).

7.1.3 *Custom Application (GAMP Category 5)*

For a custom application, the regulated company typically contracts a supplier to develop the application based on defined requirements. Therefore, the supplier will be involved during the full project life cycle of the system, and also to provide support during system operation as described in Chapter 4. Procedures to follow should be agreed between the regulated company and the supplier and be documented in the appropriate plan.

Procedures adopted may be those of the regulated company or from the supplier QMS (see Section 7.2).

⁴ This is also called parameterization, as may be found in process control systems and simple laboratory devices.

7.1.4 Service Provision

Suppliers that provide IT/IS services (including the provision of cloud-computing services) should operate within a QMS (see Section 7.2). Quality planning should define the activities, procedures, deliverables, and responsibilities for establishing delivery and monitoring of the service. Such a plan is a contractual document, and as such, should be approved for use by both the supplier and the regulated company.

The required information may be satisfactorily covered by other contractual documents such as a service level agreement, in which case a separate plan would not be required.

The extent to which the good practices described in Section 7.2 apply to the provision of services will vary considerably depending on the scope and nature of the service and should be defined as part of the supplier QMS.

Further guidance specific to IT/IS service providers and cloud service providers is available in the ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management, Chapter 4: IT Service Management [20].

7.2 Supplier Good Practices

Table 7.1 lists good practice activities that apply to product and application development, support and service provision. These are further described in this section.

Table 7.1: Supplier Good Practices

Step	Practice	Description
1.	Establish QMS	The supplier QMS should: <ol style="list-style-type: none"> 1. Provide a documented set of procedures and standards 2. Ensure activities are performed by suitably competent and trained staff 3. Provide evidence of conformance with the defined procedures and standards 4. Enable and promote continual improvement, including adoption of current software methods, good practices, and appropriate tools and automation
2.	Establish Requirements	The supplier should ensure that clear requirements are defined or provided by the regulated company.
3.	Quality Planning	The supplier should define how their QMS will be implemented for a particular product, application, or service.
4.	Assessments of Sub-Suppliers	Suppliers should formally assess their sub-suppliers as part of the process of selection and quality planning.
5.	Produce Specifications	The supplier should specify the system to meet the defined requirements.
6.	Perform Design Review	The design of the system should be formally reviewed against requirements, standards, and identified risks to ensure that the system will meet its intended purpose and that adequate controls are established to manage the risks.
7.	Software Production/ Configuration	Software should be developed in accordance with defined standards, including the use of code review processes. Configuration should follow any defined rules or recommendations and should be documented.

Table 7.1: Supplier Good Practices (continued)

Step	Practice	Description
8.	Perform Testing	The supplier should test the system in accordance with approved test plans and test specifications.
9.	Commercial Release of the System	System release to customers should be performed in accordance with a formal process. Note: This is not release into GxP environment, which is a regulated company activity.
10.	Provide User Documentation and Training	The supplier should provide adequate system management documentation, operational documentation, and training in accordance with agreed contracts.
11.	Support and Maintain the System in Operation	The supplier should support and maintain the system in accordance with agreed contracts. The process for managing and documenting system changes should be fully described.
12.	System Replacement and Retirement	The supplier should manage the replacement or withdrawal of products/ services in accordance with a documented process and plan. The supplier also may support the regulated company with the retirement of computerized systems in accordance with regulated company procedures.

7.3 Quality Management System

It is recommended that suppliers follow a QMS, preferably based on recognized standards. The QMS should define:

- The process being followed to deliver and support the product, application, or service
- The process being followed to understand customer business process needs, and capture customer requirements (including business, quality, and regulatory requirements)
- Responsibilities, including clear separation of authority between Quality Assurance (QA) and other groups such as product development, product support, finance, or marketing
- Deliverables
- Management of documentation, records, and information
- Planned reviews of the QMS and internal audits
- Management of incidents/problems/non-conformities
- Approach to continual improvement, including the adoption of current software development models, methods and good practices, and associated selection of supporting tools and automation

The QMS should be based on a life cycle concept for the development and subsequent support of the computerized system. There are many equally valid life cycle approaches that may be used by suppliers, including Agile. This Guide does not recommend any particular approach, but rather highlights those activities expected of suppliers to support the regulated company in achieving and maintaining compliance.

The QMS should include procedures covering the activities that support system development and support, such as:

- Software management, control, and release
- Development change control
- Configuration management
- Traceability
- Training of supplier staff
- Document, records, and information management
- Backup and restore

Many systems developed today are based on software products and packages, which are configured to meet user requirements. Such products, normally, will come with supporting documentation, and where possible this documentation should be used in the system life cycle. Further modules of custom software may be required to provide specific functionality, such as interfaces and reports. The design and development of such software should be fully documented.

The QMS should cover the approach to continual improvement. For example, CMMI [3] provides an approach based on a framework for assessing and improving organizational capability and maturity. The use of metrics for measuring and improving the quality of software and hardware should be considered as part of the approach to improvement.

The use of infrastructure and IT service delivery approaches such as ITIL [5] is encouraged. Industry guidance, such as *ISPE GAMP* [38], should be treated as supporting information and should not override the supplier's established QMS.

7.4 Requirements

Requirements are captured, defined, and developed internally by the supplier (in the case of product development), and/or may be provided by the customer (for a configured product, custom application, or a service).

The requirements should define clearly and precisely what the system should do and state any constraints. Requirements should be developed, reviewed, and approved in accordance with the development method implemented by the supplier whether it be linear or iterative.

Changes to requirements should be controlled. Changes to subsequent specification documents that affect the requirements should lead to an update of the requirements.

Regulated companies wish to maximize the use of supplier testing to support their compliance activities. Therefore, requirements should be written such that they can be tested. Individual requirements should be traceable through the life cycle.

For configured products and custom applications, the regulated company should describe the business processes to be automated. In the case of configured products, these processes should be aligned with the functionality of the product to be used. This may require significant process reengineering.

The use of appropriate and effective software tools should be considered by suppliers to manage information/records that may otherwise have been managed via paper documentation.

See Appendix D1 for further details on RS. See Appendix D8 and Appendix D9 for details on managing requirements using an Agile approach.

7.5 Supplier Quality Planning

The supplier should define how the QMS will be implemented for a particular product, application, or service.

This should include defining the life cycle model being followed and the project organization, activities, procedures, deliverables, and responsibilities for establishing the fitness for intended use of the system. The approach may include prototyping or other software development techniques. The role of supplier QA should be clearly defined.

These supplier quality requirements may be captured in a separate document entitled quality plan or other supplier documentation. In each case, the quality requirements should be clearly documented, reviewed, approved, accessible, and followed.

See Appendix M6 for further details on quality and project planning.

7.5.1 Prototyping

Prototyping methods may be used to clarify user requirements or to evaluate areas of risk. Typically, a prototype is used to evaluate the acceptability of a user interface, the performance of critical algorithms, suitability of the overall solution, or aspects of system performance such as capacity and speed.

To be effective, the aims and objectives of the prototype should be clearly defined, and the prototype evaluated against these to ensure the objectives are met. Suppliers should define how information gained can be incorporated into the product in a controlled manner. Prototyping is often used as part of Agile software development where control is achieved through processes such as the Agile ceremonies (e.g., daily Scrum), control of the sprint backlog, use of tools, and sprint reviews and retrospectives (see Appendix D8 for more information).

7.6 Sub-Supplier Assessments

Suppliers should formally assess their sub-suppliers as part of quality planning. They also should be periodically reassessed in accordance with the QMS.

The decision whether to perform an audit of their sub-suppliers should be documented and based on a risk assessment.

In some cases, such as those sub-suppliers related to the location and hosting of GxP regulated data, sub-supplier management may have a direct impact on regulated company compliance, and should be made transparent.

See Appendix M2 for further details on supplier assessments.

7.7 Specifications

For product development, the supplier should document the functionality and design of the system to meet the defined requirements. This should cover software, hardware, and configuration.

References to documentation and specifications should not be interpreted as requiring traditional documents, and the maintenance of records and information in appropriate and effective software tools is encouraged. The illustrative terminology used in this section and elsewhere is not prescriptive and not intended to imply that use of modern terminology associated with, for example Agile methods, is not acceptable.

Functional specifications should clearly and completely describe what the product will do. They should be produced such that objective testing can be subsequently performed.

Design specifications should be based on the functional specifications and should be sufficiently detailed so that the product can be developed.

Specifications may be covered by one or more documents depending on the complexity and risk of the product.

Specifications should be reviewed and approved with traceability established between related documents. They should be managed under change control with the awareness that change to one document may lead to a change being required in others.

It is recognized that not all suppliers use the specification terms described in this Guide, but may still meet the objective of providing adequate specifications through the provision of other documentation or other specification methods.

See Appendices D1 and D3 for further details on specifications.

If the supplier is involved in configuring a product, service, or developing a custom application, the number and level of specifications can vary considerably and should be agreed with the regulated company (see Section 6.2.7). Section 4.2.6 provides examples of specification requirements for configured products and custom applications.

7.8 Design Reviews

Design reviews evaluate deliverables against standards and requirements, identify issues, and propose required corrective actions. They should be planned and systematic reviews of specifications, design, and development, and should be planned to occur at suitable stages during the life cycle, based on risk, complexity, and novelty. Design reviews aim to identify and eliminate issues that would otherwise lead to changes at a later stage.

See Appendix M5 for further details on design reviews. See also Appendix D8 for details on managing design reviews using an Agile approach.

7.9 Software Production/Configuration

The supplier should establish and maintain a formal system for controlling software production. Appropriate methods and tools should be used and the use of these should be defined. Rules and conventions, such as acceptable languages, coding standards, version control, and naming conventions should be established. The use of code reviews should be considered.

Existing software should be used in accordance with documented build processes and taking into account any changes in the system hardware, interfaces, and peripherals.

If the supplier is involved in configuring a product, it should be performed in accordance with the controlling configuration specification and follow appropriate guidelines and recommendations.

See Appendix D4 for further details on management, development, and review of software.

7.10 Testing

For product development, the supplier should test the product in accordance with approved test plans and test specifications.

The test specifications, when executed, should demonstrate that all requirements, functionality, and design have been met.

This may involve one or many stages of testing, depending on the nature of the product. For example, a simple product may only need one test specification while a complex product may have:

- Module (unit) testing
- Integration testing
- System testing

Test records for each stage should be reviewed and approved, and retained for a period defined in the QMS (not to be shorter than the supported lifetime of the current software version).

Test failures should be managed in accordance with a formal documented process.

See Appendix D5 for further details on testing of computerized systems.

If the supplier is involved in configuring a product or developing a custom application, the number and level of test specifications can vary considerably and should be agreed with the regulated company (see Section 6.2.9).

7.11 Commercial Release

System release to customers should be performed in accordance with a formal process that describes the criteria for release, responsibilities, records to be retained, and items to be released, including software, hardware, and documentation. Clear policies and criteria should be established for the acceptability or otherwise of product release, based on the number and severity of known defects. Release notes defining fixes, changes, known problems, and new features should accompany each release, including minor releases and patches.

This activity is particularly applicable to commercially available products and services. For custom applications, the regulated company would typically accept the system following regulated company procedures.

Note that commercial release by a supplier is not a release into the GxP environment, which is a regulated company activity (see Section 6.2.10).

7.12 User Documentation and Training

The supplier should provide adequate system management documentation, operational documentation, and training for both maintenance and operation in accordance with agreed contracts.

7.13 System Support and Maintenance During Operation

The supplier should support and maintain the system in accordance with agreed contracts. Formal procedures should be followed, typically covering areas such as:

- Operational change control
- Configuration management
- Patch management
- Incident management
- Documentation management
- Backup and restore
- Business continuity
- Disaster Recovery (DR)
- Managing software product releases
- Training of supplier staff
- System maintenance
- Security management

These topics are covered by separate sections and appendices in this Guide.

7.14 System Replacement and Retirement

The supplier should manage the replacement or withdrawal of products or service in accordance with a documented process and plans. Sufficient notice of the retirement of a system or version should be given to regulated companies to allow them to plan for their required activities.

The supplier also may support the regulated company with the retirement of computerized systems.

8 Efficiency Improvements

This Guide provides a flexible framework for achieving compliant computerized systems that are fit for their intended use. The benefits will be obtained only if the framework is applied effectively in the context of a particular organization.

In this Second Edition of the Guide many examples of efficiency improvements are also discussed in the context of critical thinking (see Section 3.4 and Appendix M12).

Aspects that can assist efficiency include:

- Establishing verifiable and appropriate user requirements
- Making risk-based decisions
- Leveraging supplier input
- Leveraging existing records and information
- Using efficient testing practices
- Employing a well-managed handover process
- Managing changes efficiently
- Anticipating data archiving and migration needs
- Using tools and automation

8.1 Establishing Verifiable and Objective User Requirements

Requirements should be analyzed to ensure that they are fully defined and are verifiable and objective. For example:

Incomplete Requirement: Room shall be controlled at 20°C.

Complete Requirement: Room shall be controlled at 20°C ± 2°C. Excursions of no greater than 7°C are permitted for < 10 minutes.

The level of detail is dependent on the novelty and complexity of the processes and system being implemented.

Table 8.1 lists some aspects to consider when developing verifiable and objective requirements.

Table 8.1: Considerations When Developing Requirements

Aspect	Purpose
Process Knowledge	In order to identify key requirements of the system that are related to the business or manufacturing process
Business Knowledge	To ensure that requirements are challenged against business need and benefits can be realized
Ownership	To ensure clarity and understanding of the stated requirements
Analytical	To ensure that requirements are challenged to confirm they are complete and accurate
Technical/Product	To ensure that requirements are practical in terms of available technology
Process/Product Impact	To ensure requirements that impact the process or product are clearly identified (including those related to a CQA or CPP)
Technical Authorship	To ensure that requirements are written in concise, correct, and unambiguous language

8.2 Making Risk-Based Decisions

Risk management provides an opportunity to scale life cycle activities. However, benefit may also be achieved if organizations use risk assessments as input to decisions to omit or include an activity. Examples of areas where risk assessments may help with scaling include:

- Number and depth of design reviews required
- Need for, and extent of, source-code review
- Rigor of supplier assessment
- Depth and rigor of testing

Similar opportunities exist during system operation, for example:

- Extent and level of specification and verification of changes
- Rigor of the backup and restore process
- Level of business continuity required
- Frequency and level of DR
- Degree to which identity checks are completed prior to providing access rights
- Scope and frequency of periodic reviews

The benefits of risk-based decision-making can be maximized only if the conclusions and decisions can be leveraged. Therefore, there should be a practical, searchable means of access to conclusions and decisions available to those involved in decision-making, e.g., during subsequent assessments or reviews, during change management, and on subsequent projects. Organizations may use a risk register to achieve this.

The risk-based approach should be focused and resourced for maximum effectiveness and efficiency in managing risk to an acceptable level. Organizations should not invest more effort and time into the risk-management process than is commensurate with the potential impact on the supported business processes, patient safety, product quality, and data integrity.

8.3 Leveraging Supplier Input

Supplier activities, including testing, may be used by the regulated company as part of verification, provided the regulated company has assessed the supplier processes as suitable. This assessment may include a supplier audit, depending on the risk, complexity, and novelty of the system.

The regulated company should assess the supplier for evidence of:

- An acceptable supplier quality system
- Supplier technical capability
- Supplier application of good practice such that activities performed by and information obtained from the supplier will be complete, accurate, and suitable to meet the purpose of verification

Supplier documentation, information, and records should be assessed for content and quality. Regulated company procedures and processes should be flexible regarding acceptable format and structures so that supplier documentation may be leveraged. Regulated companies need to apply critical thinking during the assessment of suppliers and be familiar with current approaches to IT/IS service and solution delivery.

If inadequacies are found in the supplier quality system, technical capability, application of good practice, or documentation, then the regulated company may choose to manage potential risks by applying specific, targeted additional verification checks or other controls, rather than repeating supplier activities and replicating supplier documentation.

The decision and justification to use supplier activities to support the verification of the computerized system should be based on the intended use of the system and should be documented and approved by SMEs, which may include the quality unit or other quality function as relevant, for aspects critical to patient safety, product quality, and data integrity.

Suppliers also may have tools or techniques unique to the specification and testing of their product or used in their QC, which may be leveraged by the regulated company during specification and testing.

See also *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management*, Section 4.4: Leveraging Supplier Effort [20].

8.4 Leveraging Existing Information

In addition to the use of supplier activities and information, regulated companies should also leverage their own activities and information related to existing systems when introducing new, similar systems. Examples include:

- Laboratory equipment
- Secondary manufacturing equipment
- Packaging equipment

Relevant information to reuse may include risk assessments, RS, various plans, test specifications, test results, and design reviews.

The new system should be assessed and any differences with the existing system identified and managed by the introduction of appropriate specification and verification as required. A review of existing information should determine which may be used as is or updated as required. These conclusions should be documented.

The new system should be subject to installation and verification based on user requirements. The validation report should explain the rationale for reuse of documentation.

8.5 Using Efficient Testing Practices

Testing is a major, time-consuming exercise. It perhaps offers the greatest opportunity for efficiency savings.

8.5.1 Reuse of Test Results

Many systems have large amounts of test results available due to suppliers following a QMS independently of a regulated company. On a project, there may be pre-delivery testing, which may include factory acceptance testing. Wherever possible, regulated companies should clearly communicate to suppliers the testing and document requirements in advance such that supplier test documentation is of the required standard to support compliance activities.

Testing also may take place to meet other business or legal requirements, such as HSE, and finance such as SOX [19]. If so, unnecessary duplication of testing should be avoided. This is particularly true for large business systems.

8.5.2 Extent of Required Testing

The level and type of testing should be risk-based and based on the nature of the component involved (for example, whether they are standard, configurable, or custom). Types of testing include:

- Normal case (positive)
- Invalid case (negative)
- Repeatability
- Performance
- Volume/load
- Regression
- Structural testing

See Appendix D5 for further details. The choice of controls to manage identified risks may result in some of these types of testing being required.

While user requirements should be verified by the regulated company through performing installation and acceptance tests demonstrating fitness for intended use, other required tests should be identified based on risk, complexity, and novelty. A review of the existing tests and results can then determine what, if any, further testing is required by the regulated company. The regulated company's procedures should allow for the use of such existing test evidence subject to a documented and justified review and approval by an SME, which may include the quality unit or other quality function as relevant, for aspects critical to patient safety, product quality, and data integrity.

8.5.3 Secondary Test Evidence

The generation and retention of secondary supporting hard copy or image evidence such as screenshots, in addition to the primary test result or output, is unnecessary in most cases and does not add value. It often adds significant cost without associated benefit, as well as adding unnecessary complexity. Such additional evidence should be generated and maintained only where value-added and necessary for effective testing.

Examples where additional test evidence may be useful include:

- Complex results, which may be difficult or time consuming to record manually, or where printouts are more efficient than manual recording
- When the result is something essentially visual or pictorial and easier to review in that format
- Where there is a need for before and after comparisons

Test evidence may be retained electronically provided adequate security and retention mechanisms are established.

Also, systems may also have data audit trails or other system logs that capture much of the information that may have been traditionally captured by screen prints.

Test results should be sufficient for objective review by the SME.

The application of various test methods to achieve the objectives of testing is discussed in *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design*, Appendix S2: Computer Software Assurance [36]. The involvement of trained, experienced, qualified testers and other software professionals will allow the most appropriate test methods and techniques to be used.

8.5.4 Use of Test Witnesses

The use of witnesses during testing is not a requirement and involves a significant overhead, without added benefit. A regulated company may, however, choose to witness some tests, e.g., site or factory acceptance testing, for commercial, contractual, communication, education, or training reasons. It is disproportionate and unnecessary to require test witnessing or requiring the tester to initial every test step to affirm they followed the instructions.

Decisions to use witnesses should consider:

- Knowledge and experience of testers:
 - Trained testers with sufficient knowledge of the system should be used, for example, nominated end users who have been given appropriate training in testing
- Practical issues:
 - Systems, e.g., process control systems, may require two people; one in a control room and one operating/observing equipment on-site
- Level and degree of review by an SME:
 - Use of independent witnesses may form part of the review process
- Degree of automation of the tests and the resulting test evidence, e.g., audit trails or system logs

See Appendix D5 for further details on testing of computerized systems.

8.6 Employing a Well-Managed Handover Process

System handover (from the project team to the process owner, system owner, and operational users) should be well-managed. It is a prerequisite for the effective maintenance of system compliance during operation. Handover should be planned in accordance with agreed criteria and should consider:

- Support requirements for maintenance, as defined by IT or engineering
- Outstanding problems or deficiencies
- How long business processes can be stopped to enable handover
- Ability and steps required to roll back to a previous operational state
- Situations where updates are not optional, e.g., updates pushed to SaaS applications
- Information required at handover (e.g., specification and verification documentation, user and maintenance manuals or guides), or new or updated user procedures
- Training needs (user, support, and maintenance)
- Clear communication between groups, e.g., with application support and client service groups who may need to provide help desk support
- The impact, e.g., on the change-control process to apply during the handover period, when handover is to be phased
- Responsibilities during handover, e.g., for accepting the system and for assessing the severity of outstanding problems or deficiencies
- Potential need of a period of elevated support and maintenance, often referred to as hypercare
- Preservation of implied as well as explicit knowledge, e.g., through SME lists and recording of lessons learned

8.7 Managing Changes Efficiently

Efficient change management should be executed in conjunction with configuration management. Key elements include:

- Documented description and business benefit of the change
- Confirmation of availability of resource
- Assessment of the impact of the change on the application, the underlying infrastructure, the people (users and engineering support staff), and the documentation
- Leveraging the risk assessment information from the original project and assessing any new risks introduced by the change to define the strategy for maintaining compliance - this includes the need for any regression testing
- Evaluation of the change from the financial, technical (IT or engineering), and compliance perspectives at the lowest technically competent level prior to management approval

- Establishing and maintaining the distinction between changes at the pharmaceutical quality system level (impacting the medicinal product life cycle) versus changes at the IT delivery and service management level
- Minimizing the number of approval points in the process
- Documentation and communication of the decision
- Execution and verification of the change, using traceability to identify existing applicable tests
- Closing the change record in a timely manner

Weaknesses in change management systems that may lead to inefficiencies include:

- Lack of scalability, e.g., for minor changes or for standard infrastructure components that change regularly
- Failure to execute change management steps in the appropriate sequence
- Failure in scheduling or in identifying dependencies
- Abuse or misapplication of emergency change processes
- An inability to prevent unnecessary changes
- Failure to keep specifications current
- Failure to leverage existing documentation relating to risk assessment and control, traceability matrices, or protocols
- Lack of follow-up processes to close a change record
- Independent change processes leading to duplication of effort for processes, equipment, and computer systems
- The inappropriate application of like-for-like principles in change management (see Appendix O6 for further details)
- Inadequate management of changes conducted by a supplier, leading to life cycle documents and configuration management records that are out-of-date
- Lack of adequate follow-up after emergency changes (see Appendix O6 for further details)

See Appendix O6 for further details on change management. See also ITIL [5] for further details on change management within an IT service environment.

8.8 Anticipating Data Archiving and Migration Needs

Data archiving and migration requirements should be considered to ensure that data structures and formats are efficient. For detailed discussion see also the *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design*, Chapter 3: Retention Strategy [36].

8.8.1 Different Retention Periods

It may be difficult to archive data with different retention periods that share the same data structures.

It may be difficult to destroy retained data that is no longer required, e.g., to reduce risk exposure to lost data and retention costs, where data with different retention periods share the same data structures.

A data structure that separates data by retention period can address the requirements of archiving and data destruction, but may involve a complex database design.

8.8.2 Data Formats

The migration of custom data formats to a replacement system requires special attention and may cause difficulties. The use of standard data formats should assist subsequent data extraction and migration.

8.8.3 Static and Dynamic Data

Data migration may be complicated where static and dynamic data is combined in a form that is difficult to separate.

Refer to Appendix D7 and *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36].

8.8.4 Data Ownership

Lack of clear data ownership can lead to failure to appropriately dispose of data at the end of its retention period or make such processes and decisions unnecessarily difficult.

8.9 Using Tools and Automation

The use of appropriate tools and automation to support IT process and infrastructure management, system life cycle management, and software specification, development, and testing, can bring significant efficiency improvements, while also increasing quality and lowering risk. The use of such tools is discussed in Appendix D9.

9 Appendix M1 – Validation Planning

9.1 Introduction

This appendix covers the production of individual validation plans for systems or projects (computerized systems validation plans), and also gives information on validation policies and Validation Master Plans (VMPs) for background and context.

Computerized system validation plans describe how the validation is to be performed for specific systems. Validation policies define management intent and commitment. VMPs describe the areas of the company where validation is required and provides an overview of validation planning for those areas.

The terms validation policy, VMP, and computerized system validation plan are used for consistency with other sections of this and other GAMP documents, and because they are the most commonly used terms in the industry. It is recognized that some companies use alternative terminology.

It is a regulatory expectation that validation activities are planned; see for example EU Annex 11 [32] Section 4.

9.1.1 Changes from GAMP 5 First Edition

Changes have been kept to a minimum to avoid disruption to companies that have been successfully following GAMP guidance on this topic since First Edition publication. The changes are:

- Take into account the validation of SaaS solutions
- Take into account the validation of systems developed in an incremental or iterative manner (Agile)
- De-emphasize VMP, with the main focus remaining on computerized system validation planning, which is the primary topic of this appendix

9.2 Scope

This guidance may be applied to all GxP regulated computerized systems. The guidelines apply to both new and existing computerized systems, and sites and organizations in which these systems are used.

Where a computer system is regarded as one component of a wider manufacturing process or system, particularly in an integrated QbD environment, specific and separate computerized system validation may not be necessary, and separate computerized system validation plans would not be required. This environment requires both complete product and process understanding and that the CPPs can be accurately and reliably predicted and controlled over the design space. In such a case, the fitness for intended use of the computer system within the process may be adequately demonstrated by documented engineering or project activities together with subsequent process validation or continuous quality verification of the overall process or system. The same principle applies to the adoption of PAT.

For automated manufacturing equipment, separate computer system validation should be avoided. Computer system specification and verification should be part of an integrated engineering approach to ensure compliance and fitness for intended use of the complete automated equipment.

9.3 Computerized System Validation Plans

9.3.1 General Guidelines

A computerized system validation plan should be produced for each GxP regulated computerized system focusing on aspects related to patient safety, product quality, and data integrity. It should summarize the system and/or project, identify measures for success, and clearly define criteria for final acceptance and release of the system.

The plan should reflect the requirements of the regulated company QMS, but should interpret such requirements, taking into account the GxP processes to be supported, identified quality risks, and the parties involved, to define a specific strategy for achieving compliance and fitness for intended use. As always, critical thinking should be applied.

The plan should define:

- What activities are required
- How they will be performed and who is responsible
- What the output will be
- What the requirements are for acceptance
- How compliance will be maintained for the lifetime of the system

The level of detail in the plan should reflect the risk, complexity, and novelty of the system. For simple or low-risk systems a separate plan may not be needed; applicable aspects of planning may be covered within another document or process.

A generic or common plan may be used for similar systems (e.g., in a laboratory), but should adequately reflect the characteristics of specific systems. Where customization is performed or where supplier resources are to be leveraged, requirements should be communicated to the supplier at the start of the project so that the supplier may contribute to the content of the plan.

The plan defines how compliance and fitness for intended use is to be achieved and how the process is to be controlled and reported. In some cases it may be convenient for a series of reports to be produced throughout the project. The plan should take this requirement into account and indicate the different types of report to be produced: covering progress made, issues raised, and acceptance of different phases of the project. Software development often makes use of tools for aspects such as requirements development, code review/analysis, testing and performance monitoring, and summary information from these may be incorporated into reports.

Planning should commence as early as possible, ideally no later than during the development of the initial user requirements.

The plan may require modification during the project if there is a significant change in strategy or scope following initial approval, in which case project change control should be applied and the plan updated accordingly.

The plan, along with the associated report, may be one of the first documents offered during an inspection or audit to demonstrate regulatory compliance. It should, therefore, be written at a level suitable to be understood by a wide readership. Jargon and technical detail should be avoided.

See Appendix M7 for details on the related topic of computerized system validation reporting.

9.3.2 Roles and Responsibilities

Responsibility for computerized system validation planning ultimately rests with the process owner. This may be delegated to a project manager and also may involve the system owner.

Typically, the computerized system validation plan is approved by the process owner and quality unit.

The meaning of each approval should be defined.

Even though regulated companies cannot delegate their regulatory accountabilities to a supplier or service provider, they may leverage the knowledge, experience, and activities of the supplier or service provider through risk-based assessment, management, and governance processes. These should be described or referenced in the computerized systems validation plan.

Regulated companies may utilize a diverse range of suppliers and service providers to provide software and services, including Infrastructure, Platform, and Software “as a Service” (collectively referred to as XaaS or cloud computing). In the case of SaaS, the responsibility for much of the system specification and verification activities and operational controls will reside with the service provider. For example, in the case of SaaS, the plan should describe how the regulated company will leverage product specification and verification activities performed by the supplier or service provider as part of product development and performed under the supplier or service provider QMS. The plan should describe the configuration specification and verification activities and user acceptance activities required to ensure fitness for intended use.

The roles and responsibilities of the regulated company and the service provider should be clearly described in the computerized system validation plan. Responsibility for system and related activities may be delegated to suppliers and service providers, but in all cases regulatory accountability lies with the regulated company.

9.3.3 Contents of the Plan

Topics discussed in this section may be included in the plan. The guidance provided is intended to be neither prescriptive nor exhaustive.

The level of detail should reflect the risk, complexity, and novelty of the system. Separate sections may not be appropriate or necessary for all systems.

9.3.3.1 Introduction and Scope

Information provided should include:

- The scope of the system
- The objectives of the validation process
- Applicable main regulations
- Review, maintenance, or update process for the plan itself

9.3.3.2 System Overview

A general description of the system in simple terms should be provided, including:

- Business purpose and intended use for the system
- A description of the system and its data at a high level

- System scope and boundaries and overview of the system architecture

Diagrams are encouraged. If the system is being developed using an incremental and iterative approach (Agile), and requirements will develop over time, then this should be described, and information relevant to the initial release should be provided.

9.3.3.3 *Organizational Structure*

Roles and responsibilities should be described. These may include third parties such as suppliers and service providers. Regulated company roles typically include:

- Project manager
 - Project management and planning
 - Control of project activities, resources, and costs
 - Monitoring progress and initiating corrective action
 - Ensuring issues and project objectives are correctly addressed and resolved
 - Reporting to sponsor or senior management
 - Liaising with the quality unit to ensure compliance
- Quality unit
 - Ensuring compliance with appropriate regulatory and quality requirements and company policies
 - Providing support for the review and approval of deliverables
 - Approving the release of the system for use
- Process owner and/or system owner
 - Implementing and managing the system by the business user community
 - Approving completion of stages/phases

SMEs are those individuals with specific expertise and responsibility in a particular area or field (e.g., quality unit, engineering, automation, development, operations).

SMEs should take a lead role, as appropriate within their area of expertise and responsibility, in ensuring that systems are compliant and fit for intended use.

SME responsibilities may include planning and defining verification strategies, performing reviews, defining acceptance criteria, selecting appropriate test methods, executing verification tests, and reviewing results.

The number of personnel required to approve specific documents should be kept to a minimum.

9.3.3.4 *Quality Risk Management*

The QRM approach to be applied should be described.

An initial risk assessment should be performed based on an understanding of business processes and business risk assessments, initial user requirements, regulatory requirements, and known functional areas. Any relevant previous assessments may provide useful input, and these should not be repeated unnecessarily.

The results of this initial risk assessment should include a decision on whether the system is GxP regulated (i.e., GxP assessment). It also should include an overall assessment of system impact.

The level of effort, formality, and documentation of subsequent risk-management activities should be determined based on level of risk and system impact. Stages at which risk assessment will be performed should be identified. (See Section 5.3 and Appendix M3.)

Large enterprise systems, such as Enterprise Resource Planning (ERP) systems, may have some functionality declared as GxP relevant, while other functionality is declared outside the scope of GxP. In such cases the method by which this decision is made should be described and should consider:

- The requirement for deciding levels of GxP impact
- The procedures for performing the assessment
- The current status of the process (recognizing that the assessment may be repeated and the impact assessment updated)

Any specific QRM procedures or standards to be followed should be defined.

9.3.3.5 Validation Strategy

The strategy for achieving compliance and ensuring fitness for intended use should be described, based on consideration of:

- Risk assessment
- Assessment of system components and architecture
- Supplier assessment

The key conclusions of any assessments performed should be included. Any specific procedures or standards to be followed should be defined.

The validation strategy should describe:

- The life cycle model
- Specification and verification approach, including use of linear, iterative, incremental, or evolutionary (Agile) development methods, as appropriate
- The inputs and outputs required for each stage of the project
- The acceptance criteria
- Approach to traceability
- Approach to design review
- Approach to leveraging supplier activities through appropriate assurance mechanisms

See Appendix M4 for further details on categories of software and hardware, and Appendix M5 for further details on design reviews and traceability.

9.3.3.6 Deliverables

The deliverable items to be produced should be listed, including responsibility for production, review, and approval. Note that deliverables may include records within tools and supporting systems and are not constrained to traditional documentation.

9.3.3.7 Acceptance Criteria

The overall acceptance criteria for the system (e.g., successful completion of defined project phases or stages) should be described. The approach to handling significant deviations should be defined. For development using incremental and iterative approaches, the acceptance criteria will typically be defined in the Minimum Viable Product (MVP) and Definition of Done (DoD).

When applying incremental and iterative models and methods the approach to reporting and acceptance for each release should be considered and defined, e.g., whether an updated validation report or equivalent is required, or whether this may be controlled through another mechanism. Such decisions should be based on the GxP impact, complexity, and novelty of the product outcome of sprints.

9.3.3.8 Change Control

The requirements for project change control should be defined, including reference to relevant procedures and/or tools.

The stage at which operational change control will be applied should be defined.

9.3.3.9 Standard Operating Procedures

The SOPs to be created or updated as a result of the implementation of the system should be defined, and the plan should identify responsibility for their production, review, and approval.

9.3.3.10 Supporting and Operational Processes

Details of relevant supporting and operational processes should be defined or referenced, including, but not limited to:

- training (including project team and user training)
- documentation, record, and knowledge management
- configuration management
- maintaining compliance and fitness for intended use
- operational processes (as described in Section 4.3 and supporting appendices)

9.3.3.11 Glossary

Definitions of any terms and abbreviations that may be unfamiliar to the readership of the document should be included.

9.4 Validation Policies

Regulated companies should have corporate or site-level policy documents that define their overall approach to computerized system quality and compliance. Such documents should define, or make reference to, the following:

- Roles and responsibilities for activities and support
- High-level expectations for deliverables
- Standards, templates, and procedures that are expected to be followed throughout the organization
- Definition of high-level processes, including the process to determine whether a system is GxP regulated
- Requirements for record and knowledge management

These policies should be readily available to all those with responsibilities for verification and validation activities, and should be referred to by relevant planning documents.

9.5 Validation Master Plans

9.5.1 General Guidelines

A VMP may be used to define the overview plan for a given period of time, large project, or program of work under which there may be several individual validation plans.

Computerized system validation is often a subset or part of a VMP covering all of an organization's validation activities. A VMP may be for the entire company, or there may be multiple VMPs for smaller business units.

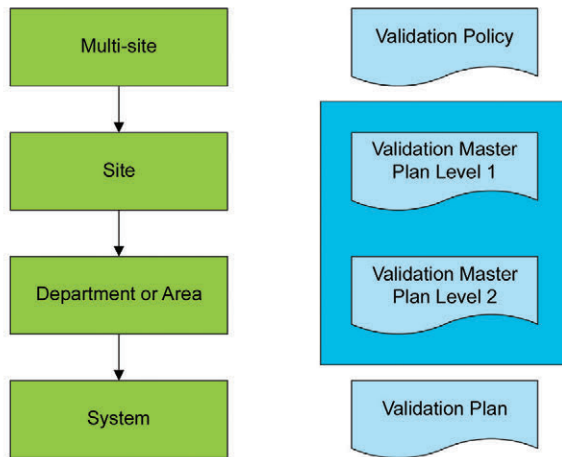
The VMP should be a clear and concise summary document, typically covering:

- Summary of facilities, systems, equipment, or processes in scope, and the respective validation status
- Current status of these facilities, systems, equipment, or processes
- Change-control process to be followed
- Planning and scheduling (including activities for new systems, activities driven by change, and periodic review)

The VMP requires approval by management, and is often subject to regulatory inspection.

The structure of VMPs will depend on the way the regulated company is structured and on company preference and policy. Companies may have a management structure that is organized hierarchically, and some choose successive planning levels that reflect the way that the company itself is organized. Figure 9.1 gives an example planning hierarchy.

Figure 9.1: Planning Hierarchy



Within a site, there may be a single VMP for the site (VMP Level 1) and a number of separate plans for the individual areas on that site (VMP Level 2). For the individual systems within a given area, a detailed plan would define the validation activities for specific systems.

Companies may merge VMP Level 1 and Level 2 into a single plan, or operate with a collection of Level 2 VMPs, and not collate them into higher level plans.

9.5.2 Roles and Responsibilities

Responsibility for creating VMPs rests with senior management. Regardless of who prepares a VMP, senior-management support is essential to ensure adequate resources for the required activities. Facility or area management should approve VMPs.

The quality unit should approve the policies and procedures for validation, including validation planning. The quality unit is responsible for verifying that the proposed approach complies with company quality standards and policies, and meets regulatory requirements.

The meaning of each approval signature should be defined.

9.5.3 Contents of the VMP

The VMP should be a summary document that is brief, concise, and clear. It should cover:

- Scope
- Reference to relevant policies
- Organizational structure
- Summary of facilities, systems, equipment, and processes
- Record types, content, availability, and retention
- Planning and scheduling
- Change control
- Reference to existing documents

10 Appendix M2 – Supplier Assessment

10.1 Introduction

This appendix provides a risk-based approach to performing supplier assessments. Regulated companies should consider formally assessing each supplier of GxP regulated computerized systems and services. The assessment approach of the system/service being provided should be based upon the application of critical thinking to understand the risk to patient safety, product quality, and data integrity. Documented justification should be provided for not assessing suppliers of GxP regulated systems/services.

Topics covered in this appendix include:

- The reasons for carrying out supplier assessments
- The different types of assessment
- The assessment process
- Postal/email audits and on-site/virtual online audits
- Joint and shared audits
- Corporate audits
- Supplier preparation for an audit
- Supplier certification
- International standards and certification

Note that in this appendix the term audit is used to cover both a physical or virtual online visit to the supplier and a formal assessment using a questionnaire (known as a postal or email audit).

This appendix covers both assessments of prospective suppliers of computerized services and systems, including cloud (XaaS) service providers, and existing suppliers who have not yet been assessed.

The material contained within this appendix also may be used by regulated companies to assess the competence of:

- External service providers (e.g., validation, project management, engineering support, maintenance) who support one or more of the various life cycle phases of computerized systems
- Internal functions, such as IT and engineering

Open-Source Software (OSS) is a developing area and requires special consideration [3].

Example checklists and questionnaires for this appendix are supplied separately. They are intended for guidance only and may be customized to suit a particular type of supplier/service provider, as there may be other factors that require consideration when assessing individual suppliers. It is important to recognize that the assessment process should allow for the application of critical thinking before and during the assessment, and that it does not enforce a philosophy of blindly following a checklist.

10.1.1 Changes from GAMP 5 First Edition

While the approach to performing supplier assessments is essentially unchanged, the following considerations have been evolving since this Guide was first published and are addressed as appropriate:

- The use of cloud (XaaS) platforms to provide service provision from infrastructure to applications
- The increased use of iterative approaches to software development (Agile) and the implementation of tools that support the approach and maintain information (records) on the development process
- Increased regulatory focus on data integrity adding the need to assess data integrity risks and mitigations
- External factors (e.g., global COVID-19 pandemic) driving increased use of virtual online auditing (also known as remote or desktop audits) rather than physical on-site auditing

10.2 Reasons for Supplier Assessment

Regulated companies require a high level of confidence that computerized systems will meet their technical, commercial, and regulatory requirements. They also wish to leverage the knowledge, experience, and documentation of the supplier.

Regulated companies should assess the quality and reliability of their computerized system suppliers and service providers. Regulated companies require documented evidence that computerized systems will consistently perform as intended, and assurance of the structural and functional integrity of the software.

Note: The increased use of Agile approaches and supporting software development tools as well as the appropriate generation of information and records rather than paper documentation, should be taken into consideration when assessing suppliers. References to documentation/documented evidence are often interpreted within a very narrow context of traditional approved documents/specifications rather than seeing these also as records/information/artifacts within software tools.

The computerized system supplier should build quality and integrity into a software product during development, as it cannot be added effectively (e.g., through testing and rework) later by the regulated company. The supplier is also better positioned to produce most of the required information during the development process. Suppliers should, therefore, be assessed to determine the adequacy of their development and support processes, and the supplier-assessment approach described in this appendix provides a scalable approach to carrying out such an assessment.

For many systems there is likely to be more than one supplier and each of the suppliers within the supply chain should be considered for assessment. In some cases, a single supplier may provide several components or perform a number of activities, in other cases multiple suppliers may be involved. It should be noted that in the case of audits of cloud service providers, particularly SaaS providers, the audit may be limited to the primary service provider. If there is no contract in place with other supplementary service providers in the supply chain beyond that, audits for all providers will not be possible and therefore such a provider may be considered unacceptable.

Consideration should be given to the scope of products and services to be assessed. It is considered inefficient to focus the process solely on one product required by a project when the regulated company is likely to use a wider range of products or services from the supplier. By adopting a broader approach, multiple assessments can be avoided – this is often a particular issue for larger, globally regulated companies where supplier-assessment coordination across the company can be difficult.

Supplier assessments also are an opportunity to develop relationships with suppliers, to clarify expectations and intentions, and to identify misunderstandings and risks. The assessment process enables the regulated company to establish a picture of the supplier's operation, which is vital input when planning specification and verification activities. The assessment report should, therefore, provide a balanced view of what was found, including positive observations, along with a list of concerns.

10.3 Types of Assessment

On-site auditing may be expensive and time consuming in terms of preparation, travel, availability of personnel and facilities during the audit, and time needed to write and review reports. The use of a risk-based supplier assessment approach that focuses on key suppliers can help to reduce these costs and is the basis of this appendix.

There are three main options for performing a supplier assessment:

1. Basic assessment based on available information (For components that are considered as commodities, e.g., common desktop applications, a documented decision not to perform any assessment may be appropriate)
2. Postal audit using a questionnaire (can also be performed via email)
3. On-site audit (or virtual online audit if appropriate) by relevant specialist, auditor, or audit team

Typically, a basic assessment is sufficient for lower impact systems, while higher impact systems may require formal audits. Postal audits may be appropriate for suppliers of standard and configurable products and services. Some cloud (XaaS) service providers may not accept postal audits, in which case it is necessary to confirm that the service provider has acquired and maintained an appropriate certificate of accreditation of their IT and Quality management processes.

Note: Virtual online audits should be considered as a subset of on-site audits. Virtual online audits should be performed to the same level and detail as on-site audits, following the same assessment process except where differences are indicated in this appendix.

Note: Other assessment processes may be considered such as that adopted by SOC2+ [42].

10.4 Assessment Process

It should be noted that the assessment process should be completed before any contract is finalized and the service commenced.

The overall assessment process is shown in Figure 10.1. The main steps are (with further details given in the subsequent sections):

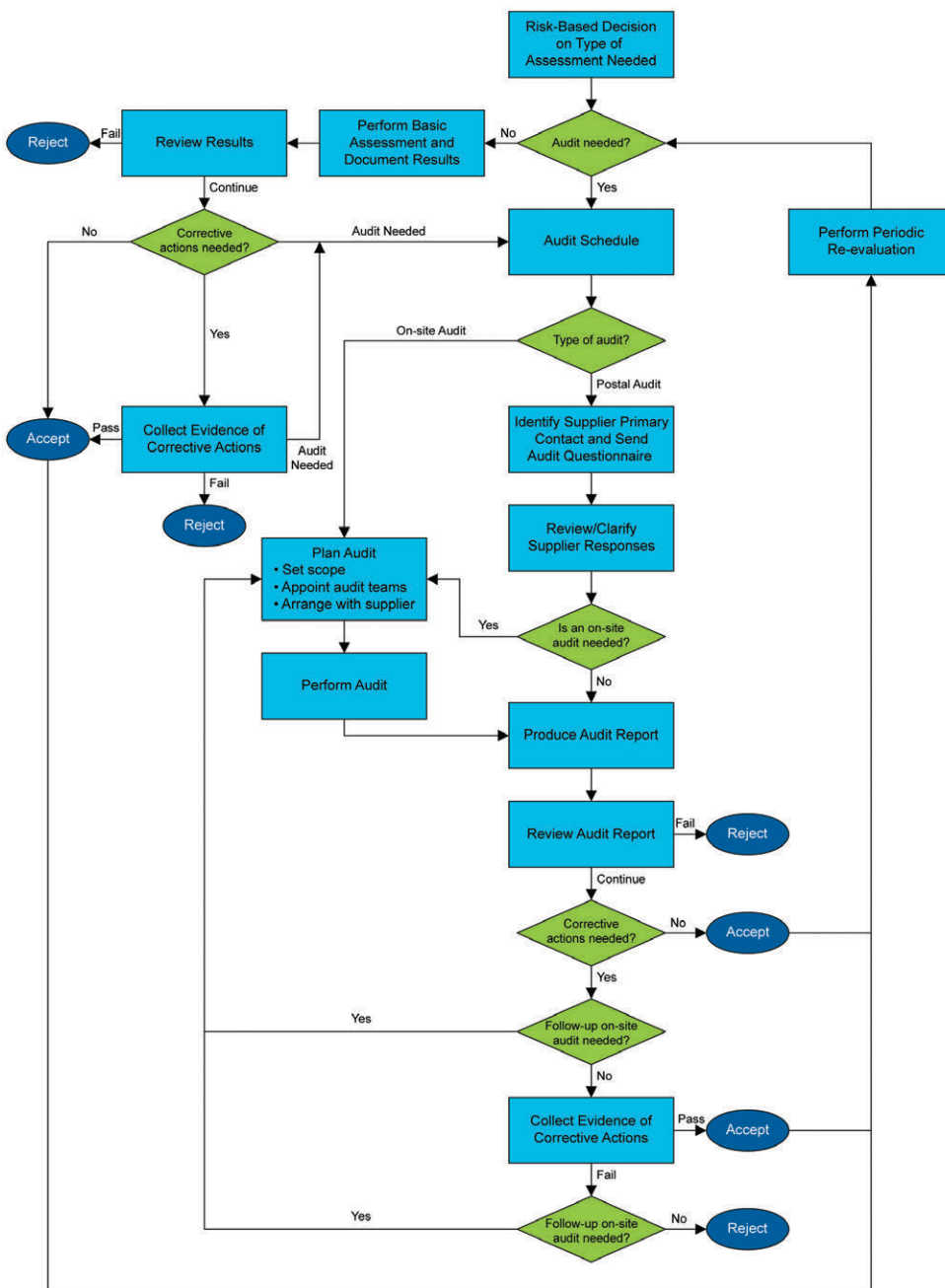
- A risk-based decision on the most appropriate assessment route
- Perform a basic assessment if this is deemed sufficient. Otherwise, perform either a postal/email audit or an on-site/virtual online audit, as determined following the initial risk assessment. An on-site/virtual online audit also may be required based on the findings of the postal email audit.
- Document assessment or produce audit report
- Determine and monitor corrective actions and document completion, which could involve a follow-up on-site/virtual online audit
- Accept or reject the supplier

Once suppliers have been accepted, they may be subject to periodic re-evaluation by the regulated company at a frequency specified in their SOPs. Periodic re-evaluation can be performed by a basic assessment, postal/email audit, or an on-site/virtual online audit. Suppliers should be made aware of this possibility in advance of the initial audit. The re-evaluation should take into account any service performance issues encountered by the regulated company.

Regulated companies normally maintain a supplier audit schedule that indicates which suppliers have been audited, when the audit took place, the reason for the audit (e.g., new supplier, follow-up audit, surveillance audit, see Section 10.4.4.2). Audit schedules also help regulated companies to plan joint audits of suppliers.

Useful guidance on planning, performing, and documenting audits of quality systems may be found in ISO 19011 [43].

Figure 10.1: The Assessment Process



10.4.1 Need for Audit

A decision should be taken on the need for an audit based on the results of the initial risk assessment and system impact (see Section 11.5.3.1 of Appendix M3), considering novelty and complexity, and the categorization of components (see Appendix M4).

10.4.2 Basic Assessment

A basic assessment may be based on:

- A review of public domain information, including information regarding certification
- Market reputation
- Knowledge and experience of prior performance
- Discussions with other regulated companies

The results of the assessment should be documented, either in a separate assessment report or as part of another document. Identified issues should be addressed satisfactorily and documented. The assessment may determine that an audit is necessary.

For components that are considered as commodities, e.g., common desktop applications, a documented decision not to perform any assessment may be appropriate.

10.4.3 Postal/Email Audits

There is a significant cost to both regulated companies and suppliers associated with conducting audits at the supplier's premises. This cost is associated both with the regulated company representatives traveling to the supplier and with the number of days of effort from both parties required to support an audit. A postal/email audit provides a mechanism for reducing these costs. Any problems found during the review of the postal/email audit may be resolved via more information from the supplier or escalated into an on-site/virtual online audit, which either replaces the postal/email audit or focuses on specific areas of concern and provides supplementary information that can be appended to the postal/email audit.

A postal/email audit can provide a good understanding of the supplier's systems. It also may provide an indication of how a supplier approaches the management of quality, which may be confirmed by a site visit.

The value of a postal/email audit is enhanced when all documentation requested is provided by the supplier. The postal/email audit then comes closer to being a desktop version of an on-site/virtual online audit. Only limited value may be obtained from the postal/email audit when no supporting documentation is provided in response to the audit challenges.

10.4.3.1 Applicability of Postal/Email Audits

A postal/email audit may be used as:

- A part of the tendering process in order to determine if a supplier merits further consideration. It can assist in the production of a shortlist of potential suppliers who may, or may not, be subject to a detailed postal/email audit or an on-site/virtual online audit prior to award of contract
- A preliminary audit to provide information to the audit team in order to focus efforts in critical areas during an on-site/virtual online audit, thus potentially reducing the length of the audit at the supplier's premises or time taken online with the supplier's representatives

- A broad audit of the supplier's business processes to determine whether or not the system or service can be considered to be a mature, trustworthy product that does not require an on-site/virtual online audit. The assessment would typically review company information, number of customer installations, product history, product release information, the supplier quality manual and key procedures, and system life cycle and support activities supported by documented evidence.
- A follow-up audit for suppliers that have successfully passed an on-site/virtual online audit with outstanding corrective actions
- A means of periodically reassessing a supplier who provides an ongoing service or who is an ongoing supplier of products
- A means of auditing other premises of the supplier where the same QMS is implemented
- A means to address a broad range of topics and to determine any areas of weakness in the supplier's business processes that may indicate that an on-site/virtual online audit of the supplier is required

10.4.3.2 Postal/Email Audit Primary Contact

To ensure continuity in communication between the supplier and the regulated company a primary contact should be established for both parties as part of the postal/email audit process. It is also recommended that the person providing the information required in the postal/email audit on behalf of the supplier is independent of the product or service being audited, e.g., from the supplier QA function.

10.4.3.3 Postal/Email Audit Questionnaire

The content of the postal/email audit questionnaire will depend on its purpose. The postal/email audit questionnaire for producing a shortlist of suppliers will tend to be high level. If there is a need to assess a specific product or service, however, the questions should be more detailed and specific. An example postal/email audit questionnaire is supplied separately. This questionnaire can be used as a framework for the production of other questionnaires such as: supplier selection short list postal audits (e.g., focus on organization and QMS plus product literature), and product and area specific postal audits (e.g., focus on QMS and software life cycle activities).

Postal/email audit questionnaires are sent to the supplier for completion and may include the following:

- Company overview including any product-specific locations
- Organization, roles and responsibilities, staff training and experience
- Key products and/or service history and development plans
- QMS implementation at company level and for product-related processes
- Product/project management
- Software development life cycle processes and deliverables
- Software development life cycle support processes
- Service delivery processes
- User training
- Product support/maintenance

- IT infrastructure management and control
- Security
- Use of subcontractors, including both external organizations and individuals

The assessment of software products will normally be version specific and so the questionnaire should be drawn up accordingly.

The returned questionnaire is examined by the regulated company and clarification sought for areas of discrepancy. A summary of the audit findings and conclusions and the status awarded to the supplier are written up in a short report, which is sent to the supplier for verification. Once agreement is reached, the report is issued to the regulated company for review and approval.

If during the review of the supplier's response to the questionnaire problems or concerns are found, the postal/email audit process may be terminated and an on-site/virtual online audit scheduled, conducted, and documented. The final audit report should mention the reason(s) for changing to an on-site/virtual online audit.

10.4.3.4 Postal/Email Audit Evidence

The value of any postal/email audit will be limited or even irrelevant if there is no evidence supplied to support the completed questionnaire. The postal/email audit questionnaire should, therefore, request supporting evidence wherever possible, including real examples of the work performed.

10.4.4 On-Site/Virtual Online Audit - Preparation and Organization

10.4.4.1 Planning/Scheduling

Planning for the on-site/virtual online audit involves defining the scope, deciding on the audit team, and arranging the audit with the supplier.

10.4.4.2 Define Scope of the Audit

The audit scope is determined by the purpose of the audit (e.g., detailed, follow-up, or surveillance), and the supplier's main activities (e.g., software product development, equipment manufacture, software integration, and support services).

- Detailed audits usually cover all aspects relating to the product or service under consideration. They can, however, also be used to assess the supplier's capability to produce a quality product when custom services are being sought.
- Follow-up audits usually concentrate on specific areas of concern, as identified during previous audits, or the progress made on agreed corrective actions.
- Surveillance audits, when used by the regulated company, normally focus on areas of weakness found during previous audits, on new products and services, and can provide a vehicle for monitoring ongoing compliance.

The nature of the supplier's services will determine which areas the on-site/virtual online audit needs to cover, e.g., product development, custom software development, or support services.

The scope should be defined to meet the overall audit objective.

Example:

A regulated company has decided that a follow-up audit of a software product supplier is needed. This was due to concerns raised during the on-site audit regarding the lack of documented testing carried out by the supplier. The follow-up audit would, therefore, concentrate on the following:

- The test strategies adopted at each stage of development
- Evidence of both structural and functional testing
- Evidence that each key function of the product has been tested thus providing traceability
- Evidence of stress testing, and testing of abnormal conditions
- The use of testing tools
- The documentation standards employed, including the generation of agreed-to life cycle specifications that have traceability to controlling or preceding specifications; test results and raw test data and their traceability back to the test specifications/definitions; review of test results; actions taken in event of test failure
- Involvement of the supplier QA function in the testing process
- The independence and qualifications of testers and reviewers
- Verification that traceability from requirements through to testing is available and adequately documented

In order to cover the above topics effectively, the auditor would need to be refreshed about the supplier's organization, the software life cycle used, the quality processes followed for testing, and the appropriate controlling specifications, recognizing that these areas will have been covered already during the on-site audit.

The audit scope, therefore, drives the development of a suitable audit agenda.

10.4.4.3 Select Audit Team

The scope of the on-site/virtual online audit to be performed determines the size and makeup of the audit team. On-site audits of complex systems may require a team of at least two people, e.g., a lead auditor plus a system user and/or a technical specialist familiar with the technology/service under consideration. Less complex systems may require only a single auditor. A similar approach is adopted for virtual online audits.

A lead auditor should be appointed who has overall responsibility for the execution of the audit. The lead auditor should be an experienced auditor, with formal auditing qualifications and experience in the development of computerized systems, as applicable, and their use in a regulated environment. At least one member of the audit team should have an understanding of the technology being supplied and of the proposed application.

The audit team may also include less experienced auditors, a technical specialist, a user representative, a quality unit representative, or a purchasing representative.

The supplier should be informed in advance whenever a third-party auditor is proposed to conduct, or take part in, the on-site/virtual online audit so that any objections or concerns regarding conflict of interest or confidentiality may be raised and discussed.

10.4.4.4 Supplier Notification

Audit details should be confirmed in writing with the supplier. The reason, objective, scope, location (whether on-site or virtual online), timing, and audit team details, including the use of third-party auditors, should be included. A provisional agenda should also be provided, so that the supplier can prepare accordingly or suggest improvements. In the case of a virtual online audit, arrangements for online videoconferencing should be stressed, together with requirements for online collaboration facilities to review documentation/records remotely. The required availability of supplier staff (including technical staff) should be made clear. The need for a confidentiality agreement should be addressed in advance of the audit.

10.4.5 On-Site/Virtual Online Audit - Performing the Audit

An on-site/virtual online audit has three parts:

1. Opening meeting
2. Review and inspection
3. Closing meeting

These are described in this section of this appendix.

10.4.5.1 Opening Meeting

This meeting allows for formal introductions and permits the lead auditor to summarize the purpose and scope of the audit. The agenda can be confirmed or rearranged depending on the availability of supplier staff. In the case of on-site audits, other issues, such as the provision of a quiet room, lunch arrangements, and access to documentary records are usually clarified. For virtual online audits, other issues can include availability of staff to account for time zone differences, as well as access to documentation/records via collaborative tools.

The auditors should attempt to accommodate the supplier's suggestions or preferences, providing the audit objectives are not compromised. The supplier may wish to provide a presentation to the auditors to familiarize them with the company, and with relevant products and services as identified in the scope. This is acceptable if a time limit is agreed and observed.

10.4.5.2 Review and Inspection

Irrespective of whether the audit is on-site or virtual online, this is the main part of the audit, where the audit team examines the supplier's practices and records in accordance with the agreed scope and agenda. While each auditor will have an individual style, the purpose of the audit is to establish, through questioning and inspection, the adequacy of the supplier's operations, and to identify any areas of concern and to bring them to the attention of the regulated company's management. The auditor should adopt a "show me" (evidence based) approach when explanations are provided, making sure to interview designers and operatives as well as the management.

The auditor should consider the use of a checklist based on the agreed scope and agenda to ensure that key areas are covered and to provide a roadmap for the audit process. Example checklists and questionnaires for this Guide are supplied separately.

Any checklist used does not constrain the auditor from following up a topic in greater detail, based on professional judgment and experience. In practice, auditors will cover all checklist topic areas to a certain minimum level but will choose some areas for a more detailed inspection.

Issues of potential concern found during the audit should be raised with the supplier when they are found and examined further until the auditor is satisfied that enough relevant information has been gathered. Audit findings should have a reference to verifiable objective evidence (e.g., traceable to specific documents/records, visual observations).

10.4.5.3 Closing Meeting

The closing meeting allows the lead auditor to list observations noted during the audit, covering both positive issues and areas of concern. It also gives the supplier representatives an opportunity to respond to these findings. This response should be documented in the audit report.

The lead auditor should explain the next steps following the audit. A typical process outline would be:

- a. A draft audit report is produced by the lead auditor and may, as a courtesy, be submitted to the supplier for comment before formal submission to the regulated company's management. This can help to ensure that the report correctly describes the areas visited and the findings highlighted, and that the supplier has not been misrepresented.
- b. The lead auditor should consider any comments received from the supplier and should obtain clarification where needed. The lead auditor may consider, but is not bound by, this input and may incorporate agreed comments into the final report. The audit report should not contain any significant issues not discussed during the audit or at the closing meeting.
- c. A formal audit report is produced and issued to the supplier and the regulated company management by the lead auditor. The supplier should be told when it could be expected.
- d. The report is reviewed by the regulated company management and required supplier corrective actions determined.
- e. The regulated company's representative contacts the supplier to agree on a plan for implementing any corrective actions. This plan may include further audits.

10.4.6 Audit Report for Postal/Email and On-Site/Virtual Online Audits

The audit report for both postal/email and on-site/virtual online audits is a quality record that:

- Provides a formal record of the audit and its findings
- Acts as a major input when determining corrective action
- Provides objective evidence to support the selection or continued use of a supplier

The report should present an accurate, objective record of the findings, and auditors should gather copies of key documentation as support material. References made in the audit report to documentation examined during the audit should be unambiguous (e.g., by title, reference, date, version, copy number, and author). The supplier and regulated company should treat the report as confidential. Audit reports should be retained by the regulated company as part of the documentation set for the system.

The audit report will normally contain:

- Introduction (including type of audit performed)
- Scope of the audit
- Organization of the audit, including agenda, criteria, representatives
- Detailed findings: the checklist format may be used as the basis for presenting the information gathered in each area inspected and any findings

- Record of the closing meeting
- Conclusions

A suitable date for receipt of the corrective action plan from the supplier in response to the audit findings should be documented and agreed with the supplier.

Some regulated companies also require that the audit report include specific recommendations by the audit team.

10.4.7 Supplier Acceptance and Rejection

Based on the outcome of the audit, the regulated company may decide:

- To use the supplier unconditionally
- To use the supplier for certain products or certain versions only
- To use the supplier subject to specific corrective actions being addressed
- To provide additional regulated company supervision of the supplier and/or conduct additional testing to overcome shortfalls in the supplier's processes and/or deliverables
- To agree with the supplier on the application of a documented QMS for the purposes of the contract
- To prohibit the use of the supplier

10.4.8 Corrective Actions and Follow-up Audits

If the supplier is requested to carry out corrective actions as a result of a quality audit, then the regulated company should follow up and obtain documentary evidence of successful completion. Evidence could include copies of new procedures, testing records, design review, and code review documents. A letter of confirmation from the supplier is not normally sufficient.

If a follow-up on-site/virtual online audit is required then it should be planned, carried out, and documented. It should, however, be noted that the lead auditor should not be constrained to just the area of the corrective actions; follow-up audits often raise further corrective actions.

The outcome of the review of the audit report should be formally recorded and documented by the regulated company.

10.4.9 Re-Evaluation

Once suppliers have been accepted, they should be subject to periodic re-evaluation by the regulated company at a frequency, and following the process, specified in their SOPs. The periodic re-evaluation process should determine whether a reaudit is required based on risk and, if so, whether this will be a postal/email audit or an on-site/virtual online audit.

The decision on whether to reaudit the supplier can be influenced by:

- The criticality of the product/service provided
- Change of supplier ownership (acquisitions/mergers)
- Changes in the supplier management structure at technical/operational (business focus) level

- changes to the QMS (e.g., new business processes; changes to the certification standard; changes in the scope of certification)
- Change of license model (e.g., transition from closed source to open-source or freeware)
- Change of delivery model (e.g., transition from on-premise to cloud)

The frequency of re-evaluation will depend on factors including the results of previous assessments and experience based on use. Typically, re-evaluation will focus on specific areas rather than the whole of the supplier's QMS, thus taking several re-evaluations to assess the whole QMS in detail.

If a formal supplier surveillance program is required (e.g., for ongoing service providers), this should form part of the re-evaluation.

10.5 Joint Audits

Maintaining individual regulated company audit programs has several drawbacks:

- It is resource intensive, leading to duplicated activities by regulated companies and suppliers
- It places a heavy burden on supplier time and effort
- Multiple auditing standards may develop, which could confuse suppliers

Joint audits involve representatives from more than one regulated company performing an audit together, and a number of such audits have already been carried out. Several benefits have been identified:

- Reduced time and effort for both regulated companies and suppliers
- Increased cooperation between regulated companies
- Progress toward common auditing standards

There are a number of potential problems to overcome, however, when organizing joint audits. These include:

- Confidentiality and liability
- The makeup and size of the audit team
- Common and consistent auditor training
- Follow-up on corrective actions

Building on the experience of previous joint audits, further cooperation is likely between regulated companies.

10.6 Shared Audit Reports

Some regulated companies have established a practice of sharing their audit reports after conducting a supplier audit. If an audit report is to be shared the following topics should be documented:

- That the scope of the audit is valid for the recipient of the audit report

- That the auditor(s) qualifications meet the requirements of the recipient of the audit report
- That the audit process, including the use of any checklists, is acceptable to the recipient of the audit report

There may be liability and confidentiality issues where audit reports are shared. If reports are shared, then the agreement of all parties involved should be obtained and documented, including that of the supplier. Individual agreements may be made between companies, or a third party may provide this service.

If a shared audit is used, the regulated company should ensure that it covers the relevant aspects of the particular application scope.

10.7 Corporate Audits

Regulated companies often have a number of departments both at local and global level that conduct supplier audits including:

- Business Quality Units
- Regulatory Compliance
- IT QA
- Engineering/Project Management/Validation groups
- Purchasing

There can be a number of the above departments each serving different business lines and in different parts of the business (e.g., Research and Development (R&D), Manufacturing).

The regulated company should try to maintain a centralized audit repository so that audit efforts can be leveraged within the company and to avoid a supplier being audited by different parts of the same regulated company. The harmonization of audit requirements and reporting would help in the sharing of these audit reports.

10.8 Supplier Preparation for an Audit

This appendix is intended to provide suppliers with an indication of which products or services would require a supplier audit and the type of audit that would be appropriate. It benefits both the regulated company and the supplier if the supplier is prepared for the audit.

The supplier can prepare for a postal/email audit by:

- Providing answers to all questions in a clear and concise manner
- Providing the quality and technical documentation requested

The supplier can prepare for an on-site/virtual online audit by:

- Making the requested personnel available (or a suitable designee)
- Providing answers to questions in a clear and concise manner
- Making quality and technical documentation easily available on-site or remotely via appropriate collaboration tools to reduce time and to prevent disruption to the audit flow

10.9 Supplier Certification

There is an increasing dependence on suppliers to provide quality-assured solutions and services. Suppliers who operate to very high standards of quality may be certified against the regulated company's quality standards. In such cases, direct involvement in the design, implementation, and testing of the system by the regulated company may be limited, as greater assurance and trust in the supplier's quality approach is obtained during the audit process.

In such situations, the audit process should be very rigorous and detailed, and periodic re-evaluation of the supplier is essential.

10.10 International Standards and Certification

The certification of a supplier against a nationally or internationally recognized accredited quality standard, such as ISO 9001 [39] may be taken into consideration when planning the scope of the audit program. For example, ISO 9001 requires organizations to maintain documented information to the extent necessary to support the operation of processes and retain documented information to the extent necessary to have confidence that the processes are being carried out as planned. It is, however, very important to understand the scope and current applicability of certification.

The assessment of IT infrastructure and platform services can be assessed through the evaluation of a range of certifications and attestations made available by the IT/IS service provider. These include SOC 2+ reports (see July/August 2019 *Pharmaceutical Engineering* article "Application of the SOC 2+ Process to Assessment of GxP Suppliers of IT Services" [42]), ISO 9001 [39], ISO 27001 [44], ISO 27002 [45], ISO 27017 [46], ISO/IEC 20000-1 [47], and other certifications. NIST Special Publication 500-322 [48] provides information on each service type. Evaluation of these materials helps ensure quality and compliance with quality obligations and IT security.

11 Appendix M3 – Science-Based Quality Risk Management

11.1 Introduction

This appendix provides further detail on the QRM process introduced in Chapter 5.

QRM is a systematic process for the assessment, control, communication, and review of risks to patient safety, product quality, and data integrity, based on a framework consistent with ICH Q9 [14]. It is used:

- To identify risks and to remove or reduce them to an acceptable level
- As part of a scalable approach that enables regulated companies to select the appropriate life cycle activities for a specific system

Organizations should already have established risk assessment methods and tools (see Section 11.5). While this Guide describes one suggested approach to risk management, it does not intend or imply that existing processes and techniques should be discarded. They should continue to be used as appropriate as part of an overall QRM process.

The methods described in this appendix should be applied in conjunction with the use of critical-thinking skills (see Appendix M12) to ensure that all risks to product quality, patient safety, and data integrity have been fully and properly assessed, and as needed, mitigated.

11.1.1 Changes from GAMP 5 First Edition

While the approach for risk management is essentially unchanged, several considerations have been evolving since this Guide was first published, and are addressed here. Principle among these is the use of cloud platforms and applications, which move some of the risk-management activities outside the regulated company.

11.2 Scope

This appendix is applicable to all types of computerized systems used in regulated activities, including those supporting clinical trials, toxicological studies, Active Pharmaceutical Ingredients (API) production, formulated product production, warehousing, distribution, medical devices, and pharmacovigilance.

Separate and existing risk assessment processes may be relevant to some systems, e.g., in relation to analytical methods, chemical processes, HSE, and PAT. These should be taken into consideration and leveraged. Similarly, this approach may be used to assess and address risk in other business areas.

11.3 Benefits

Application of QRM enables effort to be focused on critical aspects of a computerized system, in a controlled and justified manner, leading to specific benefits, such as:

- Identification and management of risks to patient safety, product quality, and data integrity
- Scaling of life cycle activities and associated records according to system impact and risks
- Justification for use of supplier documentation

- Better understanding of potential risks and proposed controls
- Highlighting areas where more detailed information is needed
- Improving business process understanding
- Supporting regulatory expectations

11.4 Roles and Responsibilities

QRM is part of the overall responsibility of the business process owner, which may be delegated to project team members. Key roles are shown in the Table 11.1. Note that the roles below will all be more effective if the persons in that role are skilled in critical thinking,

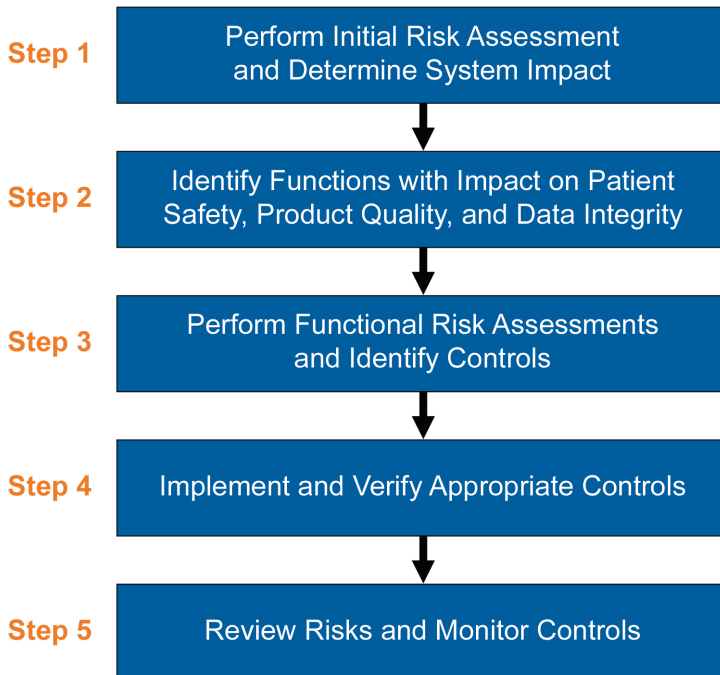
Table 11.1: Risk-Management Roles and Responsibilities

Role	Responsibilities
Process Owner/System Owner	<ul style="list-style-type: none"> • Establish team and provide resources (may be delegated to nominated project manager) • Participate in risk assessments as required • Approve documentation
Team consisting of Subject Matter Experts (SMEs) and key users*	<ul style="list-style-type: none"> • Identify, analyze, and evaluate risks to patient safety, product quality, and data integrity • Develop controls
Quality Unit	<ul style="list-style-type: none"> • Identify, analyze, and evaluate those risks associated with regulatory compliance and maintaining company quality standards and policies • Participate in risk assessments as required • Approve documentation where appropriate and necessary
Supplier	<ul style="list-style-type: none"> • Provide information on their product <ul style="list-style-type: none"> - How it was developed - How it works - How it might fail • Provide advice on controls • Participate in risk assessments as required
<p>*SMEs may include as necessary Process Owner, System Owner, Quality Unit, Business or IT Application Support, IT or Engineering Operations Support, Infrastructure specialists, supplier, or any other appropriate specialist.</p>	

11.5 Guidance

Section 5.3 describes a five-step process, as shown in Figure 11.1.

Figure 11.1: Quality Risk-Management Process



The Main Body describes each of the steps. This appendix provides further guidance on the following topics:

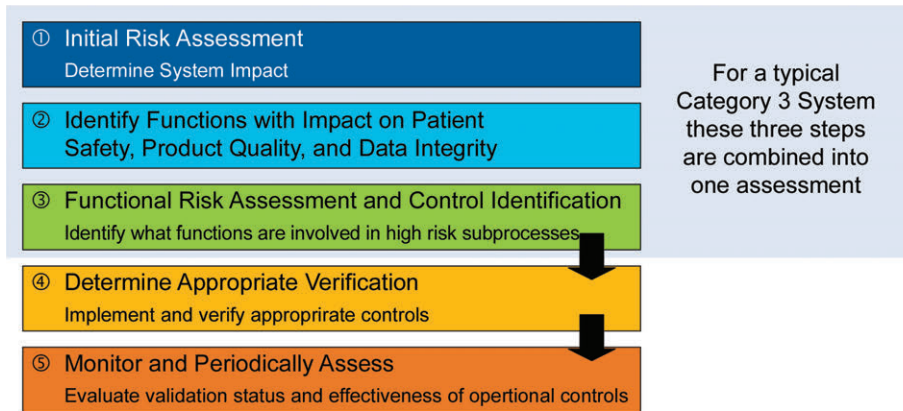
- Scalability of the process
- Applying risk management based on the business process
- Risk management throughout the system life cycle
- Risk assessment method
- The selection and use of controls
- Residual risk
- Using risk assessments to scale system life cycle activities
- Risk communication and documentation
- Examples of applying the process to different types of systems

11.5.1 Scalability of the Process

The five-step risk management process may be scaled according to risk, complexity, and novelty of individual system, with each step of the process building upon the previous output.

As an example, Figure 11.2 shows how the process is applied to a typical Category 3 product.

Figure 11.2: Process Applied to a Typical Category 3 Product



Other examples showing the scalable approach of applying the process are given in Section 11.5.7.

The process may also be used during operation, e.g., during change control. In this case Steps 2 to 5 typically should be used and information from the original Step 1 should still be available and used as appropriate.

11.5.2 Applying Risk Management Based on the Business Process

To effectively apply a QRM program to computerized systems, it is important to have a thorough understanding of the business process supported by the computerized systems, including the potential impact on patient safety, product quality, and data integrity. Aspects to consider include:

- What are the hazards?

To recognize the hazards to a computerized system requires judgment and understanding of what could go wrong with the system based on relevant knowledge and experience of the process and its automation. Consideration should include both system failures and user failures.

- What is the harm?

Potential harm should be identified based on hazards. Examples of potential harm include:

- Production of adulterated product caused by the failure of a computerized system
- Failure of an instrument at a clinical site that leads to inaccurate clinical study conclusions
- Failure of a computerized system used to assess a toxicology study that leads to incomplete understanding of a drug's toxicological profile
- Failure leads to supply interruption and a potential drug shortage

- What is the impact?

In order to understand the impact on patient safety, product quality, and data integrity, it is necessary to estimate the possible consequence of a hazard.

- What is the probability of a failure?

Understanding the probability of a failure occurring to a computerized system assists with the selection of appropriate controls to manage the identified risks. For some types of failure such as software failure,

however, it may be very difficult to assign such a value, thus precluding the use of probability in quantitative risk assessments. To some extent, probability of failure aligns with the GAMP Categories 1-5; as complexity is introduced, failure likelihood increases.

- What is the detectability of a failure?

Understanding the detectability of a failure also assists with the selection of appropriate controls to manage the identified risks. Failures may be detected automatically by the system or by manual methods. Detection is useful only if it occurs before the consequences of the failure cause harm to patient safety, product quality, or data integrity. Similar to the above, the categories are paralleled; as the complexity of failure detection processes increase, the likelihood of undetected failure increases. This applies to both manual and (generally more reliable) automated processes.

- How will the risk be managed?

Risk can be eliminated or reduced by design, or reduced to an acceptable level, by applying controls that reduce the probability of occurrence or increase detectability. Controls may be automated, manual, or a combination of both.

The above considerations are context sensitive. For example, risks associated with a solid oral dosage manufacturing area are very different to those in a sterile facility, even when the same computerized systems are used.

Similarly, the risks associated with an adverse event reporting system are very different to those in a training records database. The former can have a direct effect of patient safety, whereas the latter system is very unlikely to affect patient safety.

It is worth noting that zero risk is usually an unattainable goal, and there are diminishing returns as it is approached. Instead, companies need to think in terms of acceptable risk, sometimes known as risk tolerance. Risk tolerance is context dependent, taking into account a variety of patient safety, data integrity, business, and regulatory considerations, and it is not necessarily the same for similar processes in one company, or even for identical processes in different companies. For example, risk tolerance related to product release is lower for a life-saving oncology drug than it is for an allergy cream, even though the processes may be identical. There should be greater care and rigor exercised for the oncology drug.

It is not unusual for risks to have multiple potential impacts. It is highly advisable to recognize what the worst-case scenario may be. However, when deciding what mitigation steps to apply, that may not always be the best focus. If the likelihood of a worst-case outcome is vanishingly small it may make more sense to focus on other potential negative outcomes. This illustrates why a truly effective risk-management process is critical.

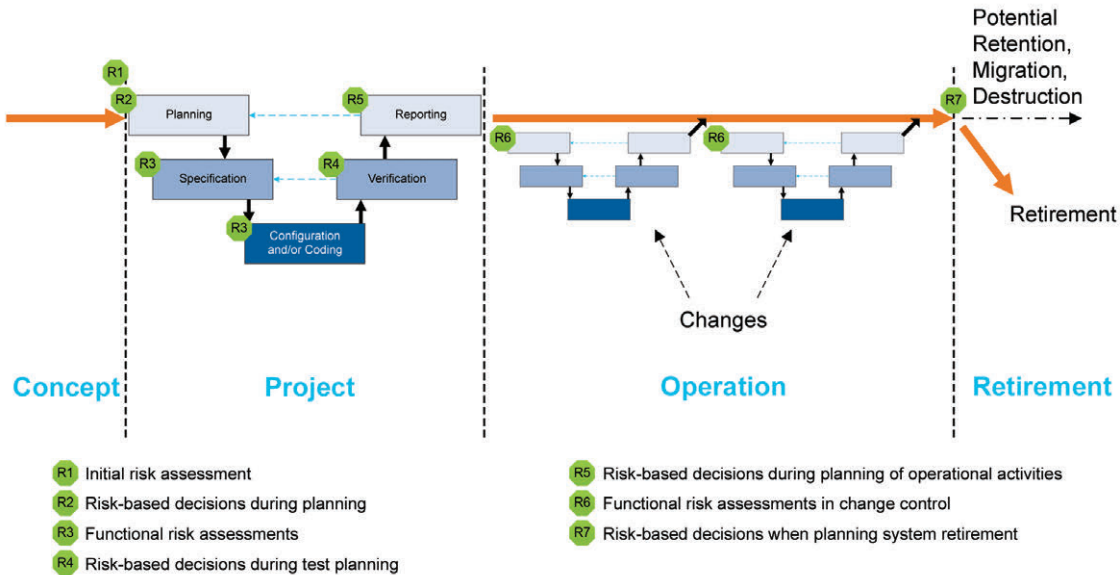
11.5.3 Risk Management Throughout the System Life Cycle

Appropriate risk-management processes should be followed throughout the life cycle to manage identified risks and to determine the rigor and extent of the activities required at each phase of the life cycle.

While risk-based decision-making should be used throughout the life cycle, different approaches may be appropriate to different situations, ranging from formal risk assessments to decisions considering pertinent risk factors. For example, formal risk assessments are usually performed at several stages when developing new software. A formal risk assessment would normally not be required, however, when determining the need for the rigor of supplier assessment. This risk-based decision is typically made and documented by the project team also considering novelty and complexity, the categorization of components, and any intention to leverage supplier documentation.

Figure 11.3 shows the typical use of risk-based decision-making throughout the life cycle.

Figure 11.3: Typical Use of Risk-Based Decision-Making



11.5.3.1 Initial Risk Assessment

An initial risk assessment should be performed at (or before) the beginning of the project phase. This is Step 1 of the process described in this section. The earlier this can be done the better, as this is the step in which a system is defined as in or out of scope for GxP. Systems that are in GxP scope may have user requirements that affect the design or selection of an application, e.g., definition of data audit trails.

This assessment therefore should precede, or at worst be in parallel with, development of the RS.

The assessment should be based on an understanding of business processes and business risk assessments, regulatory requirements, and known functional areas. Any relevant previous assessments may provide useful input, and these should not be repeated unnecessarily.

Risks introduced by computerization of the business process (e.g., electronic record integrity) should be included in the assessment.

This risk assessment focuses on important risks related to the business process, rather than detailed functions and technical aspects. One of those business risks is whether or not the system falls within the scope of GxP regulations. The process owner and the quality unit, typically, are involved at this stage in addition to the input of appropriate SMEs.

Important prerequisites for this assessment are:

- A clear understanding of the business process
- A defined boundary around the business process
- The role of the computerized system in supporting the business process including interfaces with other computer systems and other business processes. In the latter case it is important to recognize the risk and criticality of the secondary business process
- Sufficiently defined requirements (development of requirements may be iterative and influenced by the risk assessment)

Benefits of the initial risk assessment include:

- Early identification of key areas that require focus in subsequent stages, including CQAs and CPPs where appropriate
- Information for requirements development, system specification, and system descriptions
- Information to assist with developing the strategy for achieving compliance and fitness for intended use

GxP Determination

The initial risk assessment should include a decision on whether the system is GxP regulated (i.e., a GxP assessment). If so, the specific regulations should be listed and to which parts of the system they are applicable. If a system is only partially in scope of GxP, understanding this may simplify validation substantially.

For similar systems, and to avoid unnecessary work, it may be appropriate to base the GxP assessment on the results of a previous assessment, provided the regulated company has an appropriate established procedure.

It is worth noting that while the methodology described in this appendix is geared to GxP compliance, and hence focuses on risk to patients, it can effectively be applied to business risk as well.

System Impact

The initial risk assessment should determine the overall impact that the computerized system may have on patient safety, product quality, and data integrity due to its role within the business processes. This should take into account both the complexity of the process, and the complexity, novelty, and use of the system. Categorization assists in assessing system complexity and novelty (see Appendix M4).

In general, high-impact systems typically include those that support processes that:

- Generate, manipulate, or control data supporting regulatory safety and efficacy submissions
- Control, manage, or maintain critical parameters or data used at any stage, including pre-clinical, clinical, development, and manufacture
- Control, manage, or store or provide data for product release
- Control, manage, or store data required in case of product recall
- Control, manage, or store data for adverse event or complaint recording or reporting, or generally support pharmacovigilance

Process knowledge assists with determining system impact (see Section 11.7.2 for an example).

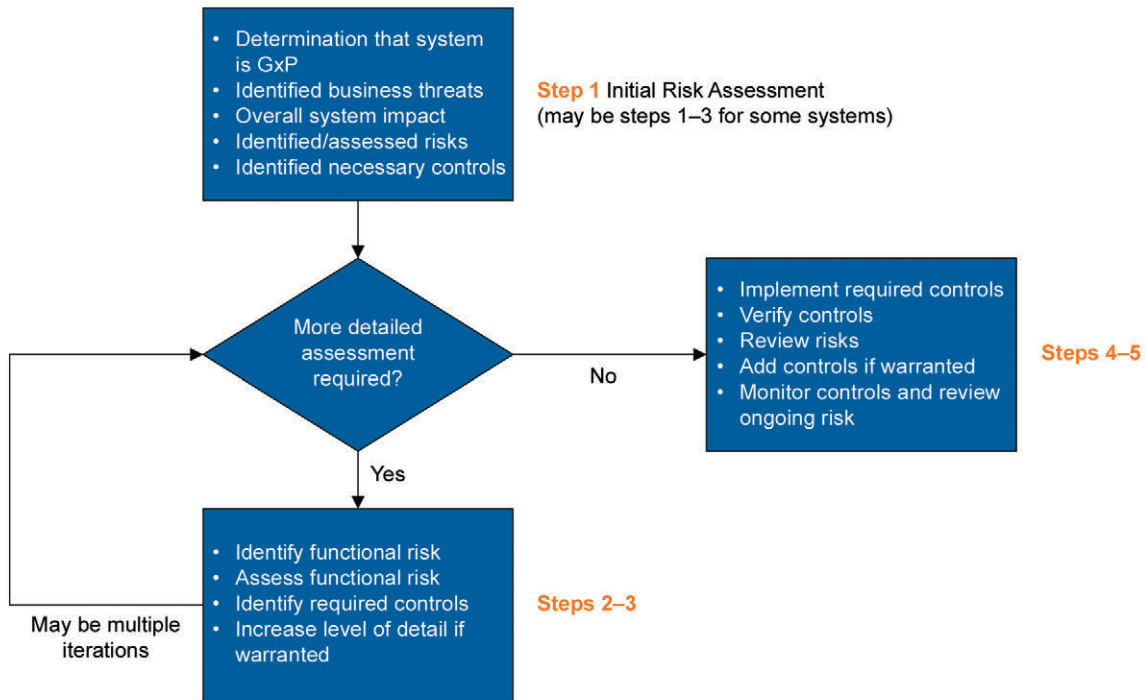
Systems that are of lower overall impact can be documented and tested less rigorously (see Section 11.5.7).

Need for Further Assessments

The amount of information available when performing the initial risk assessment depends on both the business process and on the GAMP category (see Appendix M4). For Category 3 products the amount of information available at the time of the initial risk assessment may be sufficient for all relevant risks to be identified, assessed, and controlled without the need for further risk assessments⁵. This would not be the case for a custom application (Category 5), where more detailed assessments would be required as the system is developed, as shown on Figure 11.4.

⁵ Bear in mind, however, that if interfaces exist to other systems, a deeper assessment is advisable even for a Category 3 system.

Figure 11.4: Deciding on the Need for Further Assessment



The need for further risk assessments, therefore, varies widely. See Section 11.7.1 for examples of typical approaches for different categories of systems.

11.5.3.2 Risk-Based Decisions during Planning

Risk management is an integral element of good project management practice and the approach for achieving compliance should be integrated within the overall approach.

The outcome of the initial risk assessment described in Section 11.5.3.1 should contribute to the planning process. Key risk-based decisions taken during planning include:

- Need for, and rigor of, supplier assessment
- Using the results of supplier assessments to assist in the planning to achieve compliance and fitness for intended use, including determining the involvement of the supplier
- Determining activities, deliverables, and responsibilities for achieving compliance and fitness for intended use, including extent of specification and verification
- Need for further risk assessments, when they are required in the life cycle, and the method to be used. A risk assessment method is provided in Section 11.5.4. When deciding on the level of further assessment required, information already gathered should be considered (e.g., results of supplier assessment, degree of standardization).

These decisions should be documented.

Employing risk-based decisions for achieving and maintaining compliance allows improved efficiencies at two levels:

- **Scalability:** At the system level, systems that are of lower overall impact can be documented and tested less rigorously; see Section 11.5.7 for further details
- **Focusing:** At the functional level, greater rigor can be applied to the testing and control of functions that are of the highest risk, with less rigor applied to low-risk functionality

11.5.3.3 Functional Risk Assessment

Where these are required, functional risk assessments should be used to identify and manage risks to patient safety, product quality, and data integrity that arise from failure of the function under consideration. It should be noted that there is significant value to extending this approach to the consideration of business risk. It is hugely important to keep dangerous products off the market, but if risk-based controls can prevent financial or reputational harm to the company, it is also a worthy application of the process. This is covered by Steps 2 and 3 of the process.

Functions with impact on patient safety, product quality, and data integrity are identified by referring to the RS⁶ and the output of the initial risk assessment.

A method for performing functional risk assessments is provided in Section 11.5.4. The assessments should be performed by SMEs.

Computerization may introduce particular risks (e.g., electronic record integrity, system availability, security, infrastructure) not otherwise associated with the manual business processes. The design of computerized systems may provide controls for identified risks, but may introduce other risks that require controlling. This should be included in the assessment. EU Annex 11 [32] states:

*“Where a computerised system replaces a manual operation, there should be no resultant decrease in product quality, process control or quality assurance. **There should be no increase in the overall risk of the process.**”*
[Emphasis added]

Note that this sometimes includes the removal of incidental quality controls by removing a human from the loop who would notice a problem. SME review should catch this if it occurs.

More information on the use of risk assessments for particular system types and for infrastructure is given in the relevant *ISPE GAMP Good Practice Guides* [15].

11.5.3.4 Risk-Based Decisions during Test Planning

Testing is often performed at several levels depending on the risk, complexity, and novelty of the system.

Significant savings may be realized if the need for additional controls for the business process or the computerized system is recognized early in the development process. Measures identified to manage risk should be implemented and verified. Verification of controls, typically, is covered during testing of the system and should cover any additional controls required to address deficiencies found during testing.

The results of functional risk assessments should influence the extent and rigor of verification. Testing should be focused on the high-risk functionality, minimizing effort on low-risk areas (see Appendix D5).

If controls have been added after the functional risk assessment, it may be appropriate to reassess the conclusion of that assessment, since the new controls may allow the adoption of simpler test cases.

⁶ For Category 5 systems, this identification extends to the functional specification. This may occur via a traceability matrix or tool.

Other information also may affect test planning, such as the results of supplier assessments. These decisions should be documented.

11.5.3.5 Risk-Based Decisions during Planning of Operational Activities

Operational activities should be selected and scaled according to the nature, risk, and complexity of the system in question. Opportunities for risk-based decisions when planning operation include:

- System availability
- Frequency and level of backup and recovery
- Disaster planning
- System security
- Change control (see Section 11.5.3.6)
- Periodic reviews

Any critical business processes should be identified and the risks to each assessed. Plans should be established and exercised to ensure the timely and effective resumption of these critical business processes in case of failure. The output from this risk-management approach can provide a valuable base from which to build DR and BCPs.

11.5.3.6 Functional Risk Assessments in Change Control

Change management should provide a dependable mechanism for prompt implementation of technically sound improvements following the approach to specification, design, and verification described in this Guide. The rigor of the approach, including extent of documentation and verification, should be based on the risk and complexity of the change.

If an implemented change either raises or lowers the risk profile of a function (or the system as a whole) it should be captured and documented. The verification strategy may be affected by such a change.

11.5.3.7 Risk-Based Decisions when Planning System Retirement

Risk-based decisions are required when planning system retirement, for example:

- Approach to data and record retention, destruction, or migration
- Approach to verification

11.5.4 Risk Assessment Method

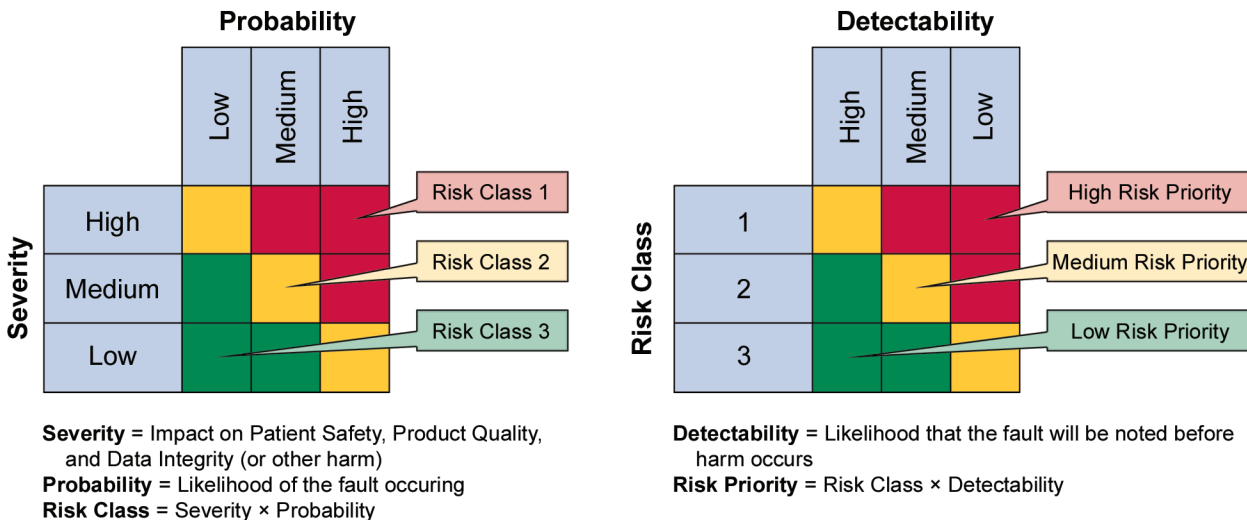
Risk management aims to establish controls such that the combination of severity, probability of occurrence, and detectability of failures is reduced to an acceptable level. Severity refers to the possible consequence of a hazard.

The method presented in this section provides a simplified functional risk assessment tool. It is not mandatory – other detailed risk assessment methods may be used. It is used, if necessary and appropriate, during Step 3 of the 5-step process.

Each of the hazards identified for a function is assessed in two stages, as shown in Figure 11.5:

1. Severity of impact on patient safety, product quality, and data integrity is plotted against the likelihood that a fault will occur, giving a Risk Class.
2. Risk Class is then plotted against the likelihood that the fault will be detected before harm occurs giving a Risk Priority.

Figure 11.5: Risk Assessment Method



The Risk Priority obtained helps to focus attention on areas where the regulated company is most exposed to hazards. These should be considered in relation to the risk tolerance, which varies from company to company based on a variety of safety, business, and regulatory drivers.

Successful application of this method depends on the ability to agree on the meaning of High, Medium, and Low for each segment of the assessment. These should be considered specifically in the context of the system in each project. An example form for documenting the functional risk assessment is provided separately.

11.5.4.1 Scaling the Method

To use resources most effectively, the decision of where to apply the most rigorous functional risk assessment should be focused on functions with highest impact. Other aspects, such as probability of occurrence and detectability should be investigated when performing the functional risk assessment. An example approach to scaling functional risk assessments based on impact is provided in Section 11.7.3.

Function impact is context sensitive. For example, failure of an instrument in an in-process Quality Control (QC) laboratory for chemical intermediates is far less likely to affect patient safety than the same instrument in a QC laboratory that releases drug product to market. This is because there are many additional controls between the intermediate and the patient in the former case, where there may be none in the latter.

11.5.5 The Selection and Use of Controls

Controls are measures that are put in place to reduce risk to an acceptable level. They may be part of a computerized system function, parallel manual procedures, or they may be downstream, intended to trap fault conditions after they have occurred, e.g., QC release testing.

Controls typically are aimed at:

- Eliminating risk through process or system redesign
- Reducing risk by reducing the probability of a failure occurring
- Reducing risk by increasing the in-process detectability of a failure
- Reducing risk by establishing downstream checks or error traps (e.g., fail-safe or controlled fail state)

In some cases, it may not be possible to reduce risk through downstream controls (e.g., for an adverse event reporting system for which there is no downstream), so controls in such cases generally are integral to the system or process and are aimed at preventing the failure from occurring or making it more detectable if it does. In other cases, the identified risk may be sufficiently low or easily detectable such that specific controls are not required.

Controls for a given process may be automated within the system, such as alarms, restrictions to data fields, required data fields, dialog box prompts for verification. Alternatively, they may be entirely independent external processes, such as subsequent chemical or physical analyses, or operator checks. Examples of controls that could be used to reduce risk are shown in Table 11.2.

Table 11.2: Examples of Controls to Reduce Risk

Control Strategy
Introduction of automated checks of data quality in downstream computerized systems
Introduction of procedures to the business process to counter possible failures, such as QC testing of products
Introduction of automated controls within the computerized system being assessed, e.g.: <ul style="list-style-type: none"> • Data verification checks within the system design to reduce the likelihood of data entry errors (such as acceptable input ranges) • User prompts to verify inputs to increase the detectability of a user error
Use is made of proven methods, tools, and components; fault-tolerance may be built into the computerized system (e.g., using replicated parts, system mirroring); the operating environment may be controlled
Increased rigor of verification testing to demonstrate that the computerized system performs as expected and can handle error conditions
Enhanced training of users

If the selected controls are still not adequate to bring risk to an acceptable level, wider risk control strategies should be considered, such as those shown in Table 11.3.

Table 11.3: Wider Risk Control Approaches

Modify Project Strategies
<ul style="list-style-type: none">• Project structure and makeup: The experience and qualifications of staff; the type of project organization; the level of education and training provided• Level of documentation and review: Alter the amount of documentation that is approved and controlled; introduce or remove formal review points to reflect identified risk
Modify the Business Process
<ul style="list-style-type: none">• How the computerized system is used in the business process: If the computerized system introduces or increases risk, consider alternative approaches to how the system is used.• Redesign of the business process: Change the business process to lessen or eliminate key points of risk.
Risk Avoidance
The risks are so high that the new way of working should not be implemented.

11.5.6 Residual Risk

Residual risks after implementing control measures should be considered. For example, reviewing control measures after testing or after implementing supporting procedural controls to determine whether selected control strategies for the system should be adjusted.

If the residual risk is above the threshold of acceptable risk, then appropriate further controls should be implemented and verified, and the impact on previously implemented risk control measures should also be considered. If this is required, analysis of residual risk should be repeated. This sequence continues until residual risk is acceptable.

11.5.7 Scaling Life Cycle Activities

Activities aimed at ensuring GxP compliance and fitness for intended use throughout the life of the system should be scaled according to:

- System impact on patient safety, product quality, and data integrity (risk assessment)
- System complexity and novelty (architecture and categorization of system components)
- Outcome of supplier assessment (supplier capability)

Specific activities that may be scaled include:

- Levels of specification
- Need for and extent of design reviews
- Need for and extent of code reviews
- Extent and rigor of verification activities

Examples showing the levels of specification and verification for different categories of system, and how the five-step process is applied, are given in Section 11.7.

The strategy for supplier assessment also can be scaled based on system impact, GAMP category, and the maturity of the product. Likewise, suppliers may contribute substantially to the risk assessment process as SMEs.

11.5.8 Risk Communication and Documentation

As defined in ICH Q9 [14], Risk Communication is the sharing of information about risk and risk management between the decision makers and others. The output of the risk-management process, including the assessments of impact and risk and the evaluated effectiveness of monitored controls, should be shared by the decision makers with other involved parties, such as the quality unit (where necessary and appropriate), the business process owner, and as appropriate the supplier.

This communication should take place throughout the risk-management process and does not necessarily take the form of a report. Although it is not necessary to communicate the acceptance of every risk, special emphasis should be given to communication to the appropriate individuals when a risk or impact has changed, so that any necessary adjustments can be made. Where necessary, the process should enable escalation of risks to senior management in a timely fashion. Regulated companies should be able to explain risk-based decisions if questioned by a regulator.

This information can also be used to improve the efficiency of the change management process. Every change to be applied can use the risk assessment information to identify the areas of the system or process impacted by the change and the risks involved in doing so. To facilitate this, risk assessments should be documented such that the results can be easily accessed during the life cycle. This is often achieved using a risk register. A risk register is a listing of recognized risks wherein risks can be prioritized and measures taken to mitigate them are recorded. This is often managed in a database or spreadsheet.

The risk-based approach will be effective only if the risk control strategies that are put in place are monitored during the life of the computerized system to ensure they remain in place and are effective. Hence, as part of the periodic review, the risk register should be reviewed to ensure that all the control strategies remain appropriate.

11.5.9 Risk Management for Outsourced Activities

The use of cloud-computing services means that direct control of risk-management processes must be delegated to a supplier in some cases. This does not absolve the regulated company of accountability for the actions of a supplier on the regulated company's behalf. Supplier assessment and management processes become a critical part of the regulated company's QRM approach. If a supplier's practices cannot be reconciled with the regulated company's QRM needs, this should be determined during the assessment and they must be prepared to not engage the supplier if the objectionable issues cannot be satisfactorily resolved.

Continuous monitoring of engaged suppliers of IT services to ensure that quality gaps do not arise should be an integral, contractually obligated aspect of the supplier relationship.

See Appendix M11 and *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management*, Chapter 4 [20], for further information on this topic.

11.6 Other Risk Assessment Methods and Tools

The following are commonly used methods and tools for risk assessment. They could be used to supplement or in some cases, replace the GAMP tool described in the previous section.

- Hazard and Operability Analysis (HAZOP)
- Computer Hazards and Operability Analysis (CHAZOP)
- Failure Mode and Effects Analysis (FMEA)

- Failure Mode, Effects, and Criticality Analysis (FMECA)
- Fault Tree Analysis (FTA)
- Hazard Analysis and Critical Control Points (HACCP)
- Basic Risk-Management Facilitation Methods
- Preliminary Hazard Analysis (PHA)
- Risk Ranking and Filtering

For further details see ICH Q9 Annex I: Risk Management Methods and Tools [14].

11.7 Examples

This section includes examples of the application of risk management. They are indicative and not intended to be definitive. Other approaches can be equally applicable.

11.7.1 Example 1 – Approaches for Different Categories of Systems

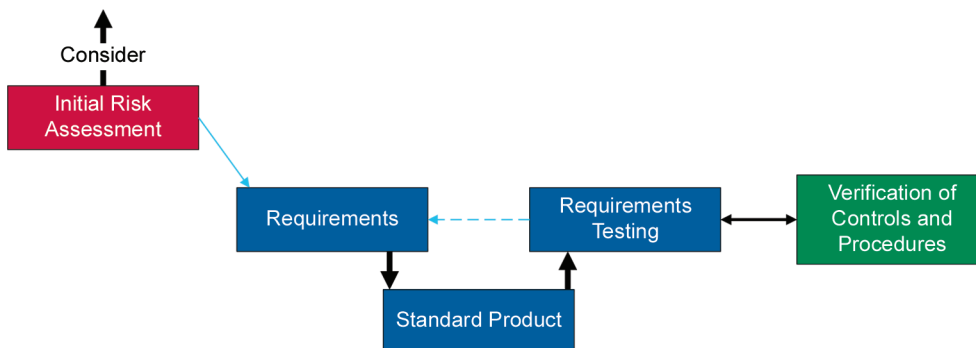
The examples provided in this section show the risk-management process applied to three categories of system.

11.7.1.1 Example Category 3 Standard Product

For a typical Category 3 product it may be possible to cover all relevant risks in a single assessment as shown in Figure 11.6. For a specific system it may be decided that further assessments are required, and these should be planned as appropriate.

Figure 11.6: Risk-Based Approach for Standard Product (Category 3)

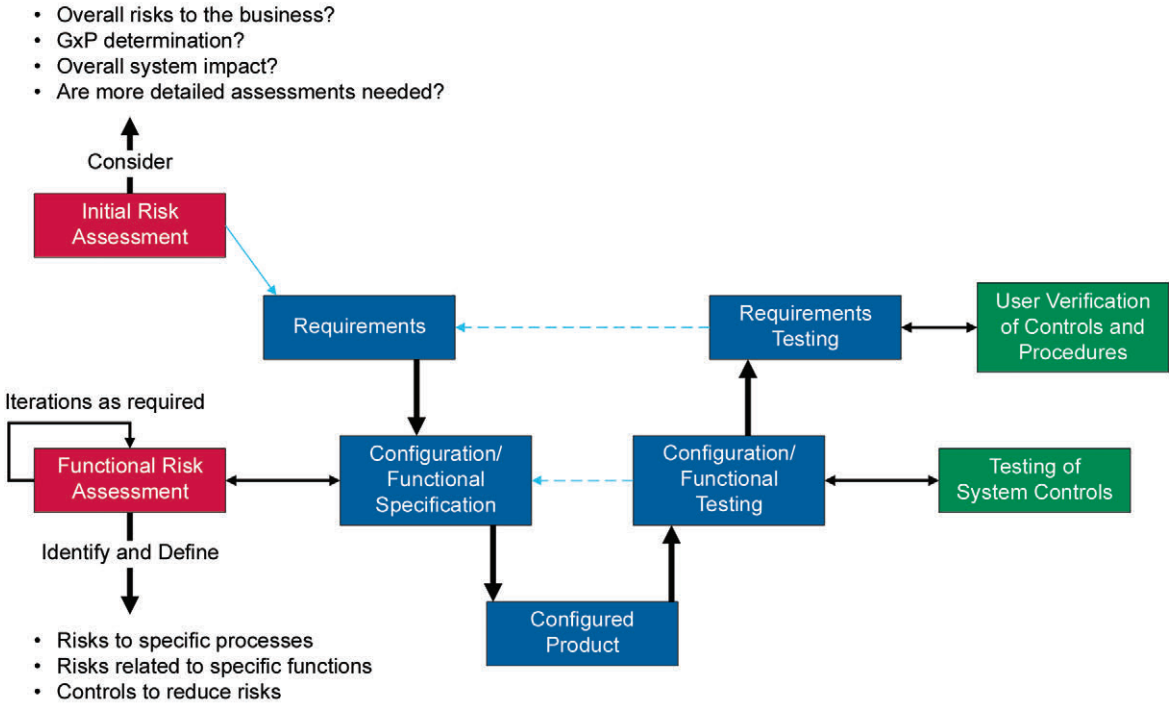
- Overall risks to the business
- GxP determination
- Overall system impact
- Risks related to specific requirements
- Specific controls to reduce risk



11.7.1.2 Example Category 4 Configured Product

For a typical Category 4 product it may be necessary to carry out an initial risk assessment to determine whether the system is GxP regulated and to understand the overall system impact, followed by one or more detailed risk assessments as the system specification is developed. However, for some systems it may be possible to cover all risks in the initial assessment; see Figure 11.7.

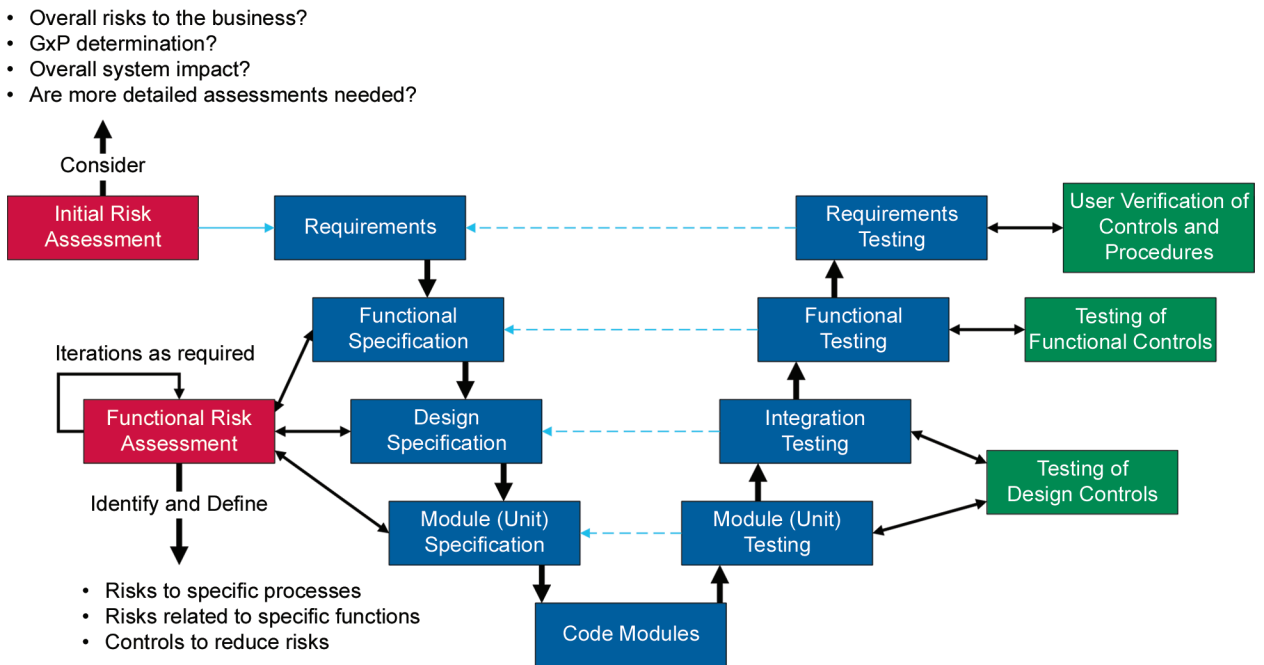
Figure 11.7: Risk-Based Approach for Configured Product (Category 4)



11.7.1.3 Example Category 5 Custom Application

For a typical Category 5 custom application it is necessary to carry out an initial risk assessment to determine whether the system is GxP regulated and to understand the overall system impact, followed by one or more detailed risk assessments as the system specification and design are developed; see Figure 11.8.

Figure 11.8: Risk-Based Approach for Custom Application (Category 5)

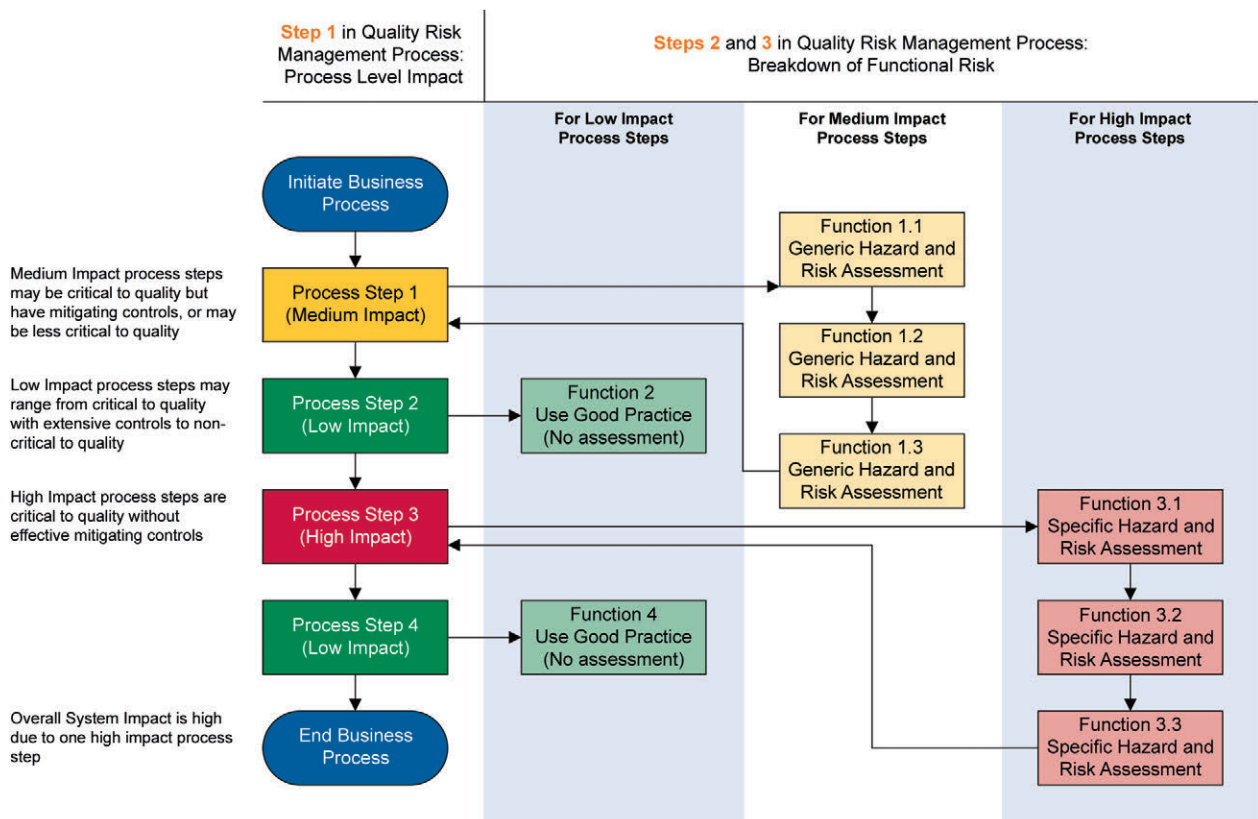


11.7.2 Example 2 - Determining System and Functional Impact

This example presents a method of determining system impact and provides information that can be used later as part of functional risk assessments.

Figure 11.9 shows how process knowledge helps determine system impact, and how the understanding of the importance of the process steps assists with the determination of functional risk in Step 2 of the five-step process. System impact is chosen to be the impact for the highest assessed process step. System impact can be used to scale compliance activities.

Figure 11.9: Analyzing the Business Process for Steps 1, 2, and 3 in the Five-Step Process



11.7.3 Example 3 - Functional Risk Assessment Based on Impact

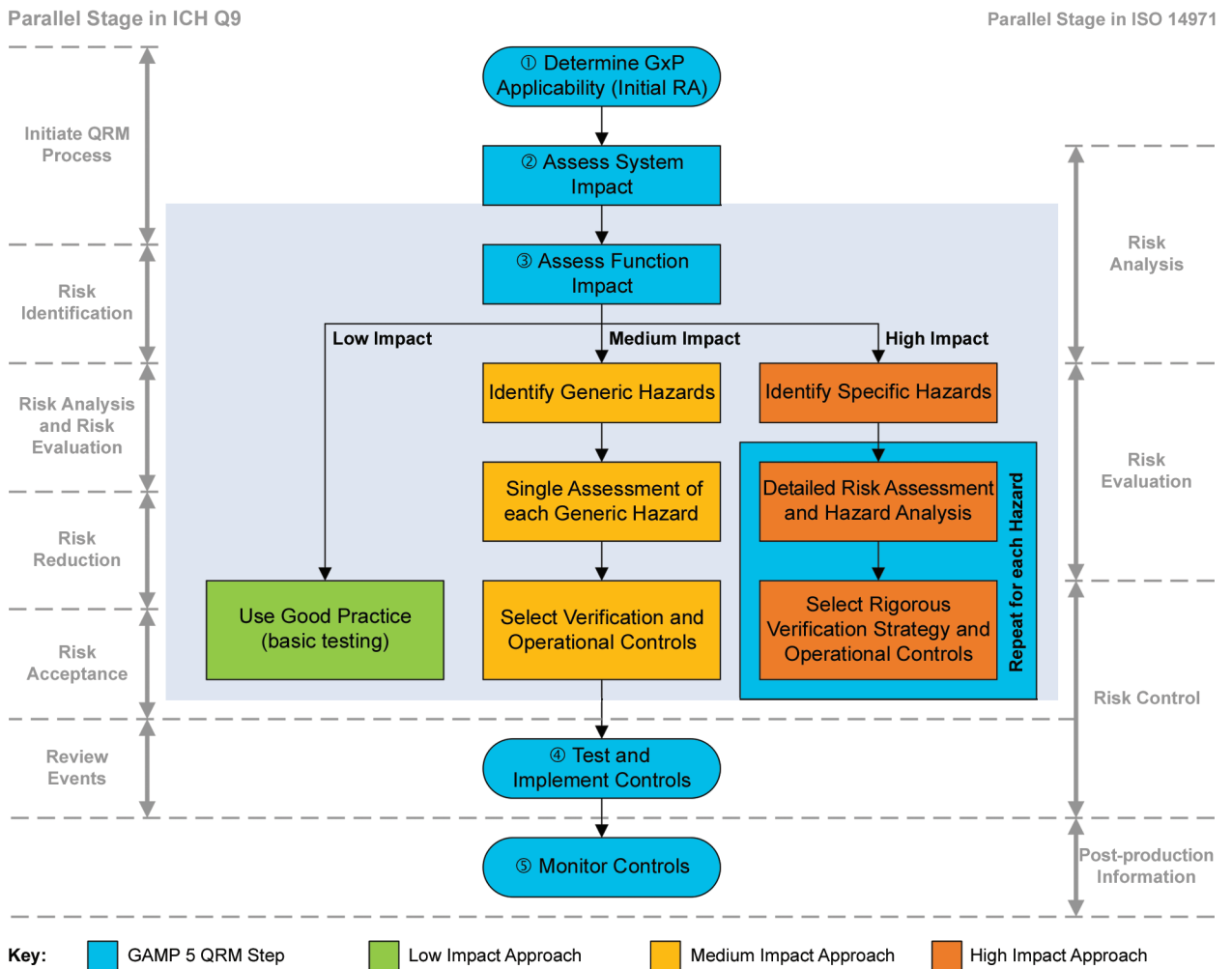
Figure 11.10 shows the five-step process with Step 3 expanded to provide an approach to the functional risk assessment based on impact. This example classes functions as one of High, Medium, or Low impact. The level of rigor applied to the assessment will be dependent upon the impact.

For High-impact functions it may be necessary to carry out a detailed assessment of hazards based upon the probability of occurrence and detectability.

For Low impact, it is reasonable to forego formal risk assessment, applying good practice to provide adequate control. For Medium impact, hazard scenarios should be considered but hazards can be grouped generally, whereas for High-impact functions more detailed and specific hazards should be considered.

In this appendix, Section 11.7.2 provides an example of how the impact of individual functions can be established. Section 11.7.4 provides examples of Medium and High-impact functions. The risk assessment method described in Section 11.5.4 may be used to carry out the assessment, such as for the High-impact functions shown in Figure 11.10.

Figure 11.10: Risk Assessment Based on Impact



11.7.4 Example 4 - Example of Medium and High-Impact Functions

Table 11.4 shows five examples of system functions, and compares the generic assessments appropriate for Medium-impact functions to the greater level of detail appropriate for High-impact functions. In both cases, consequences related to corresponding risk scenarios should be assessed for Risk Priority.

Table 11.4: Example Risk Assessments for Medium and High-Impact Functions

System	Function	Risk Scenarios for Medium-Impact* Functions		Risk Scenarios for High-Impact* Functions	
		Generic Hazard	Consequence	Specific Hazard	Consequence
Packaging line	Thermal seal	Control failure	Package or product damage	Control failure – high temperature	Package damage
					Product damage
				Control failure – low temperature	Package not sealed
Liquid filling line	Filling	Power problem	Inaccurate vial fill	Voltage spike	Damage to electronics
				Brief voltage drop due to initiation of co-located equipment	No impact
				Prolonged voltage drop (e.g., brownouts)	No impact as long as Uninterruptable Power Supply (UPS) maintains backup; inaccurate vial fill if UPS battery runs out
				Power loss < 30 min	No impact (UPS assumes load)
				Power loss > 30 min	If controlled shutdown not initiated, line crashes
IT change-control database	Change status of request	Move change status to “Approved” fails	Change status stays “Submitted”	Move change status to “Approved” fails	Change not executed
					No documented approval for executed change

Table 11.4: Example Risk Assessments for Medium and High-Impact Functions (continued)

System	Function	Risk Scenarios for Medium-Impact* Functions		Risk Scenarios for High-Impact* Functions	
		Generic Hazard	Consequence	Specific Hazard	Consequence
Toxicology database	Audit trail	Audit trail fails	Inadequate change documentation	Audit trail fails	Data changes inadequately attributed
					Old versions of data lost
Antivirus software	Automated virus definition update	Updates not downloaded	Exposure to potential virus attack	Updates not downloaded	Virus causes temporary loss of system
					Viruses cause loss of data
HPLC control system	Solvent pump control	Control failure	Incorrect assay	Control failure – high flow	Incorrect assay result due to loss of peak resolution or misidentification of peaks
				Control failure – low flow	Incorrect assay result due to incorrect component peaks assigned to reference standard or expected component peak windows

*Note that there is no implication that these functions should always be defined as high or medium impact; such an assignment must be made within the context of the business process. They are simply used as examples to illustrate the concept of generic versus specific hazard analysis and risk assessment.

For further guidance on applying this QRM approach to designing test strategies see Appendix D5.

12 Appendix M4 – Categories of Software and Hardware

12.1 Introduction

This appendix describes how software and hardware components of a system may be analyzed and categorized in terms of increasing complexity, novelty, and inherent likelihood of residual defects, as a very high-level preliminary risk assessment. These software and hardware categories may then feed into the risk assessment and supplier-assessment approaches.

It should be noted that Categories 3 to 5 are effectively a continuum with no absolute boundaries, and that most systems will contain components of multiple categories. For example, core functionality within a computerized system may be Category 3, with Category 4 workflow configuration and bespoke interfaces to other systems being Category 5. The software categories can assist in understanding the system; however, the life cycle activities should be scaled based on risk, complexity, and novelty, and supported by critical thinking.

12.1.1 Changes from GAMP 5 First Edition

This appendix has been revised to emphasize that:

- Computerized systems are generally made up of a combination of components from different categories; the categories should be viewed as a continuum
- The software category is just one factor in a risk-based approach; the life cycle activities should be scaled based on the overall GxP impact, complexity, and novelty of the system (derived from the criticality of the business process supported by the system)
- Software categories still bring benefit in deciding the rigor of supplier assessment and also when judging the probability of a failure or defect occurring in a system

12.2 Using the GAMP Categories

When coupled with critical thinking, risk assessment, and supplier assessment, categorization can be part of an effective quality risk-management approach. Categorization is not intended to provide a checklist approach to validation. Life cycle activities need to be added, removed, and scaled based on the nature of the components and identified risks.

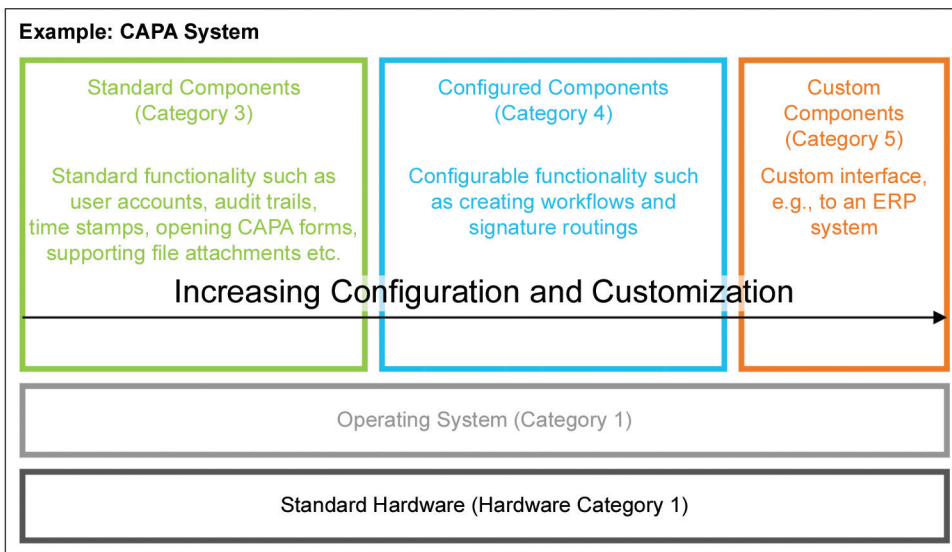
There are two main ways to use the categories:

- **Whole-system assessment:** On a whole-system level, the category of the main component may be used to help define the approach to supplier assessment. Combining categorization with an assessment of system GxP impact can help to decide the extent of supplier assessment required.
- **Functional risk assessment:** In a functional risk assessment, categorization can help increase objectivity in the assessment process. The increased risk derives from a combination of greater complexity and less user experience. An understanding of the software and hardware category can contribute to assessing risk as follows:
 - There is generally an increasing risk of failure or defects with the progression from standard software and hardware to custom software and hardware – the “Probability” in Appendix M3.

- Where the Detectability of a fault is reliant on a system function (e.g., such as an alarm or error message), the category can also influence the assessment of Detectability. A bespoke alarm may be more likely to fail compared to standard functionality and may require testing in its own right to ensure correct functionality.
- The Severity is, however, independent of the category, and is instead derived from the business process that the system supports.

Most systems have components of varying complexity, such as an operating system, standard components, and configured or custom components, and therefore the category may be different for different functions under assessment. Figure 12.1 shows an example CAPA system with a combination of categories.

Figure 12.1: Example of a Multi-Category System



It should be noted that some companies may refer to a system as “Category X” as shorthand for “This system is mainly based on a central Category X component, which will drive much of the life cycle and validation strategy decisions, even though there are some other components of different categories involved, which will need some other arrangements.”

Referring to systems as a single category can be further misleading given that even within a category there can be dramatic differences in GxP impact and complexity and that risk is a continuum. For example:

- A balance with a configurable serial output may be simplistically labeled as a “Category 4” system based on the presence of the configuration component. A Chromatography Data System (CDS) is also often classed as a “Category 4” system. However, there is an enormous difference in complexity between the balance and a CDS, and therefore the life cycle approaches should be scaled to reflect this.
- Within complex systems, there are still significant differences. A CDS may be similar in complexity and configurability to a Learning Management System (LMS) but if the CDS is used for product quality data and decisions, then it has significantly more GxP impact than the LMS and again, the life cycle approaches will not be the same.

Regulated companies are responsible for the fitness for purpose of the computerized system used in support of a business process in a GxP environment. The life cycle approach should address the layers of software involved and their respective categories. It should reflect the assessment of the supplier and any audit observations, GxP risk, size, and complexity. It should define strategies for the mitigation of any weaknesses identified in the supplier’s development process or during functional risk assessment.

12.3 Categories of Software

12.3.1 Category 1 – Infrastructure Software, Tools, and IT Services

Infrastructure elements link together to form an integrated environment for running and supporting applications and services.

Software in this category includes:

- **Established or publicly available⁷ layered software:** Applications are developed to run under the control of this kind of software. This includes operating systems, database managers, programming languages, middleware, ladder logic interpreters, statistical programming tools, and spreadsheet packages (but not business applications developed using these packages: See Appendix S3).
- **Infrastructure software tools:** This includes such tools as network monitoring software, batch job scheduling tools, security software, antivirus, and configuration management tools. Risk assessments should, however, be carried out on tools with potential high impact, such as for password management or security management, to determine whether additional controls are appropriate.
- Software, systems, and tools supporting computerized system life cycle activities and IT and infrastructure processes (as opposed to supporting business and pharmaceutical and medical device life cycle processes)

Layered software is not subject to specific functional verification although their features are functionally tested and challenged indirectly during testing of the application as part of the environment. The identity and version numbers of layered software and operating system should be recorded and verified during installation.

Infrastructure software tools are generally highly reliable, and significantly removed from any aspect of patient risk. All infrastructure software should be controlled and managed. See the *ISPE GAMP® Good Practice Guide: IT Infrastructure Control and Compliance (Second Edition)* [49] and *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management*, Chapter 4 [20] for further guidance.

12.3.2 Category 2 – Not Used

12.3.3 Category 3 – Standard System Components

This category includes off-the-shelf components used for business purposes. It includes both those that cannot be configured to conform to business processes and those that offer limited configurations using factory-provided values or ranges (also called parameterization, as may be found in process control systems and simple laboratory devices). In both cases, configuration to run in the user's environment is possible and likely (e.g., for printer setup).

A simplified life cycle approach may be applied to systems that predominantly consist of Category 3 components and have limited or moderate GxP impact. The need for, and extent of supplier assessment should be based on risk and any intended leveraging of supplier specifications and verification activities. User requirements are necessary and should focus on key aspects of intended use in the regulated environment.

All changes to software should be controlled, including supplier-provided patches. SOPs should be established for system use and management, and training plans implemented.

Configuration or parameterization choices should be managed, recorded, and verified. Where software and hardware are interdependent, calibration may be a critical quality activity as well.

⁷ Publicly available software includes Free or Open-Source software. Considerations for the use of Open-Source Software are contained in Appendix D4.

12.3.4 Category 4 – Configured Components

Configurable software components enable configuration of user-specific business processes into one or more workflows, specific to methods, or products, or processes etc. This typically involves configuring predefined software modules, and correspondingly there is an increase in the importance of capturing and managing the configuration choices.

Often the risks associated with the components are dependent upon how well the system is configured to meet the needs of user-business processes. There may be some increased risk associated with new software and recent major upgrades and the complexity of the business process being configured. Supplier testing may have used a default configuration whereas the applied configuration is typically specific to the individual company's intended use. This could result in less opportunity to leverage supplier verification and more need for regulated company activities to verify the functionality of their applied configuration.

The life cycle approach should address the layers of software involved and their respective categories. The approach should reflect the outcome of the supplier assessment, GxP risk, size, and complexity. It should define strategies for the mitigation of any supplier quality or product design weaknesses identified during the assessment.

Since each application is configured specific to the user process, support of such systems must be carefully managed. For example, when new versions of software products are introduced, serious problems can arise from the dependency of custom code on features of the software product that may have changed.

Custom software components such as macros developed with internal scripting language, written or modified to satisfy specific user-business requirements, should be treated as Category 5.

12.3.5 Category 5 – Custom Applications and Components

These applications, subsystems, or components are developed to meet the specific requirements of the regulated company. The risk inherent with custom software is high because there is no user experience or system reliability information available, and this may be reflected in the functional risk assessment ratings for the probability of failures or defects. In addition, calculation or logic errors can be difficult to detect and may persist for months or years before detection. There is a very wide range of size and complexity of custom applications, from small end-user applications based on spreadsheets to large, complex, custom process control systems.

12.3.6 Typical Examples and Approaches

Table 12.1 provides examples of systems that contain components of each category and typical approaches to follow for each software category. The actual approach (e.g., extent of supplier assessment and depth of life cycle activities) should be scaled based on how much of the system functionality is based on the different system components, e.g., predominantly Category 3 or Category 4 etc., and the GxP impact of the system. The life cycle deliverables will depend on where users are on the continuum between Standard, through Configurable, to Custom. The rigor of controls around specific functional areas and the extent of testing in those areas should be scaled based on the functional risk assessment as described in Appendix M3.

Table 12.1: Software Categories, Examples, and Typical Life Cycle Approaches

Category	Description	Typical Examples	Typical Approaches and Considerations
<p>1. Infrastructure Software and Tools</p>	<ul style="list-style-type: none"> Layered software (i.e., upon which applications are built) Software used to manage the operating environment and infrastructure Software, systems and tools supporting computerized system life cycle activities 	<ul style="list-style-type: none"> Operating systems Database engines Middleware Programming languages Network and performance monitoring tools Scheduling tools Software, systems, and tools supporting IT processes Requirements management, test management, test automation tools, etc. 	<ul style="list-style-type: none"> Record version number, verify correct installation by following approved installation procedures Assess and record the tool's adequacy for use See the <i>ISPE GAMP Good Practice Guide: IT Infrastructure Control and Compliance (Second Edition)</i> [49] See also ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management, Chapter 4 [20]
<p>3. Standard System Components</p>	<p>Run-time parameters may be entered and stored, but the software cannot be configured to suit the business process</p>	<ul style="list-style-type: none"> Firmware-based applications COTS software Some instruments (See the <i>ISPE GAMP® Good Practice Guide: A Risk-Based Approach to GxP Compliant Laboratory Computerized Systems</i> [50] for further guidance) <p>Note: Most computerized systems contain some components in this category; even a fully customized development will leverage standard software modules and libraries.</p>	<ul style="list-style-type: none"> Requirements definition for key functionality and intended use Life cycle approach scaled based on system complexity Risk-based approach to supplier assessment (scaled on component categories, GxP impact, and intent to leverage supplier activities; focus is on the supplier development life cycle as applied to the component or system under consideration, and the robustness of the supplier's product and change management processes ongoing) Demonstrate supplier has adequate QMS Record version number, verify correct installation Risk-based testing and leveraging of supplier testing to demonstrate application works as designed in a test environment (for simple systems regular calibration may substitute for testing) Procedures in place for managing data Procedures in place for maintaining compliance and fitness for intended use

Table 12.1: Software Categories, Examples, and Typical Life Cycle Approaches (continued)

Category	Description	Typical Examples	Typical Approaches and Considerations
<p>4. Configured Components</p>	<ul style="list-style-type: none"> • Software, often very complex, that can be configured by the user to meet the specific needs of the user’s business process. (e.g., via workflows, process flows) • Software code is not altered 	<ul style="list-style-type: none"> • LIMS • Data acquisition systems • SCADA • ERP • Clinical trial monitoring • DCS • ADR reporting • CDS • EDMS • Building Management Systems • CRM • Spreadsheets • Simple HMI <p>Note: Specific examples of the systems containing these components may also contain substantial custom elements</p>	<p>Per Category 3 components plus:</p> <ul style="list-style-type: none"> • Business process map and data flow diagram • Risk-based testing to demonstrate the applied configuration delivers an application meeting the business needs and workflows
<p>5. Custom Applications and Components</p>	<p>Software custom designed and coded to suit the business process</p>	<p>Varies, but includes:</p> <ul style="list-style-type: none"> • Bespoke interfaces between systems • Internally and externally developed IT applications • Internally and externally developed process control applications • Custom ladder logic • Custom firmware • Spreadsheets (macro) <p>Note: Even a fully customized development will leverage standard software modules and libraries.</p>	<p>Same as for configurable, plus:</p> <ul style="list-style-type: none"> • Supplier-assessment focus on the supplier’s QMS for new component development • Design and source-code review • Coding standard • Full life cycle information (design specifications, unit, module, integration and functional testing, etc., where relevant)
<p>ADR: Adverse Drug Reaction CDS: Chromatography Data System COTS: Commercial off the Shelf CRM: Customer Relationship Management DCS: Distributed Control System EDMS: Electronic Document Management System</p>		<p>ERP: Enterprise Resource Planning HMI: Human Machine Interfaces LIMS: Laboratory Information Management System QMS: Quality Management System SCADA: Supervisory Control and Data Acquisition</p>	

12.4 Categories of Hardware

These hardware categories are provided for information only, and do not explicitly require additional documentation.

12.4.1 *Hardware Category 1 – Standard Hardware Components*

The majority of the hardware used by regulated companies fall into this category.

Standard hardware components should be documented including manufacturer or supplier details, and version numbers. Correct installation and connection of components should be verified. The model, version number and, where available, serial number, of preassembled hardware should be recorded. Preassembled hardware does not have to be disassembled. In such cases the hardware details can be taken from the hardware's data sheet or other specification material. Configuration management and change control apply.

12.4.2 *Hardware Category 2 – Custom Built Hardware Components*

These requirements are in addition to those of standard hardware components. Custom items of hardware should have a Design Specification (DS) and be subjected to acceptance testing. The approach to supplier assessment should be risk-based and documented. In most cases a supplier audit should be performed for custom hardware development. Assembled systems using custom hardware from different sources require verification confirming compatibility of interconnected hardware components. Any hardware configuration should be defined in the design documentation and verified. Configuration management and change control apply.

13 Appendix M5 – Design Review and Traceability

13.1 Introduction

This appendix covers the design review process and requirements traceability for computerized systems.

Defects should be identified and corrected at the earliest opportunity in the life cycle. Design reviews and traceability assist with ensuring that computerized systems are fit for intended use and with keeping the overall cost of projects down through the early identification of defects and resolution of problems.

Design review and traceability can help ensure that:

- Requirements have been addressed
- The functionality is appropriate, consistent, and meets predefined standards
- Existing functionality is not negatively affected
- The system is appropriately tested with no unintended gaps between requirements and tests

For non-linear software development life cycles, for example Agile, it is appropriate for design reviews and traceability to be achieved in an iterative manner, for example, within each sprint cycle for Scrum.

13.1.1 Changes from GAMP 5 First Edition

This revised appendix encourages automated and tool-based reviews rather than manual traceability approaches where possible, as this drives efficiency and accuracy. Adoption of Agile approaches is also considered.

13.2 Scope

This appendix is intended to cover GxP regulated systems, but the methods are appropriate to all system types. Design review and traceability also may cover operational, commercial, safety, and environmental considerations.

13.3 Design Review

Design reviews evaluate deliverables against standards and requirements, identify issues, and propose required corrective actions. They are planned and systematic reviews of specifications, design, and development performed at appropriate points throughout the life cycle of the system. They are an important part of the verification process.

Design review should be performed by appropriate SMEs. The individuals performing the review should be identified. Tools can also be used to perform aspects of design review; for example, code analysis and traceability tools can provide confidence and an efficient approach.

The rigor of the design review process and the extent of documentation/artifacts should be based on GxP risk, complexity, and novelty.

Aspects that should be considered when planning design reviews include:

- The scope and objectives of the review
- What method or process will be followed
- Who will be involved, and the specific role/responsibilities
- What the outputs will be

For standard products (GAMP Category 3), a design review by the regulated company typically is not required.

For systems based on configured product, a significant part of the design review activities should have been performed by the supplier during development of the product. This should be verified during supplier assessment. User design review activities should focus on the configuration and implementation activities.

For custom applications, design reviews typically are conducted at each level of detail of specification and design, or within the sprint process for Agile software development life cycles.

13.4 Traceability

13.4.1 Introduction

Traceability establishes the relationship between two or more products of the development process.

Traceability ensures that:

- Requirements are met and can be traced to the appropriate configuration or design elements
- Requirements are verified, and can be traced to the test or verification activity that shows the requirement has been met

Accurate traceability can also provide benefit by:

- Enabling more effective risk-management and design review processes
- Judging potential impact of a proposed change
- Facilitating risk assessment for a proposed change
- Identifying scope of regression testing for changes
- Enabling fast and accurate responses during an inspection or audit

This section describes methods of achieving traceability. The approach applied should be selected based on the level of GxP risk, criticality of the business process, complexity, and novelty.

13.4.2 Principles

A means of linking relevant specification records to testing should be established and maintained. It should also be possible to trace from testing back to the relevant specification record. Traceability provides a method to ensure that all applicable elements of specification, including requirements, have been verified. It also enables faster responses to questions about verification of functions, both to meet internal business needs and during regulatory inspection. Accurate traceability depends upon on the completeness and accuracy of the specification records.

The rigor of traceability activities and the extent of documentation should be based on GxP risk, complexity, and novelty; for example, a standard product may require traceability only between requirements and testing.

For more complex systems, the relationship between requirements, specifications, and verification may not be simple, for example:

- Multiple requirements can be covered by a single DS and verified by a single test
- Multiple DS may be linked to a single requirement
- Multiple tests can be required to address one requirement or one DS

The process (documentation and/or tools) used to achieve traceability should be documented during the planning stage and should be an integrated part of the complete life cycle.

13.4.3 Methods of Achieving Traceability

Traceability may be achieved in a number of ways, including, automated software tools, spreadsheets, or embedding references directly within documents, or even a separate Requirements Trace Matrix document (RTM). Traceability may be generated as a separate deliverable or as part of an existing deliverable, such as the requirements document.

The use of automated tools is recommended as manual methods, for example, spreadsheets or RTM documents, are difficult to maintain and keep accurate.

Traceability for simpler systems can be achieved through common or consistent numbering of requirements, designs, and testing documentation, rather than a separate matrix, although tool-based traceability is recommended; see Figure 13.1.

Figure 13.1: Example of Embedded Traceability

Requirements Document for System XYZ Version 1.0	Design Document for System XYZ Version 1.0	Test Documentation for System XYZ Version 1.0
Requirement Number SysReq1.0 Temperature recording 1.1 Range 1.2 Frequency 1.3 Alarms	Design Number SysDes1.0 Temperature recording 1.1 Range 1.2 Frequency 1.3 Alarms	Test Number SysTst1.0 Temperature recording 1.1 Range 1.2 Frequency 1.3 Alarms

As shown in Figure 13.1, the numbering for temperature recording is the same in the requirements, design, and test documentation, thereby enabling traceability without creating a separate traceability matrix. This method is considered appropriate for a smaller system in low-risk situations.

For standard products, traceability between user requirements and verification may be sufficient.

For configured products, the Design column in Figure 13.1 may be replaced with a link to configuration items, providing traceability between user requirements, configuration, and verification.

For custom applications, traceability should be provided from requirements through each level of specification to the appropriate verification. Figure 13.2 provides a simple example RTM for a custom application.

Figure 13.2: Custom Application Example of a Manually Produced and Maintained RTM

Requirements	Specification (Design)	Testing
U1.1.1	S2.4.1	T1.1
U1.1.2	S2.4.5	T1.2
U1.2.1	S3.1	T2.3.1
U1.2.2	S3.2	T8.1
U1.2.3	S3.3	T8.2
Etc.		

Each reference within the matrix, e.g., U1.1.2, S3.1, T8.2, could be a reference to a section or sub-section within the relevant document, or to a totally separate document.

In practice there often is not a simple one-to-one relationship from the requirements through the different design documents. One function may fulfill different requirements, or one requirement may require different design elements.

For Agile software developments, the traceability of requirements through the life cycle is often inherent in the method and the tools used, although this should be verified as part of tool selection. For example, in Scrum, there is traceability from the requirements (epics/user stories) in the product backlog into the sprint backlog and then through the sprint and artifacts necessary to build and test the software through to the initial release and sprint retrospective.

13.4.4 Additional Requirements Traceability Considerations

Requirements traceability may be enhanced by adding more information or ideally linking to the information within tools, such as:

- A brief description of each requirement, which may assist verification
- Inclusion of change-control numbers or linkage to the change tool to enable tracking the system history and change impact. A reference to other processes that impact the system, such as deviations, may be beneficial.
- An indication of the criticality of the requirements to assist levels of testing applied to any given requirement. High-criticality requirements may have more detailed testing applied and may, therefore, reference multiple tests, whereas low criticality requirements may have a reference to a single test.
- Identification of where a requirement has been satisfied by a procedure rather than software, along with the reference to the procedure and version number
- Reference/linkage to testing may be expanded to indicate:
 - At what level the testing occurs, (e.g., unit, integration, or acceptance for linear development) or during a sprint for Agile
 - When (e.g., development, test, or operational)
 - Where the testing occurs (e.g., global or local)

In this case, the level of effort in testing should relate to the criticality of the requirement and the level of acceptable risk.

- Traceability to a maintenance or calibration record for the instrument required for a test and requirement

Note, however, that any such additions to a manually developed and maintained traceability matrix will make it more difficult to navigate and maintain.

For large projects other tools, such as a Document Management System (DMS) with the capability to maintain the links between documents (both in the DMS and reference to documents generated and stored outside) may be used.

13.4.5 Practical Issues to Consider

The level of detail required for traceability can be a difficult balance to strike. The following considerations seek to help in balancing usefulness, complexity, and maintainability.

- For linear projects, the strategy for traceability should be established during planning and requirements development. For Agile projects, the strategy for traceability should be established at the start of the project and reviewed during execution on a continuous-improvement basis.
- Traceability could simply comprise references to supplier documentation, if this is adequate.
- The supplier should have their own traceability for the software under their control. This should be verified during supplier assessments where this is appropriate.
- Requirements need not trace to technical controls in all circumstances. Requirements can trace to procedural controls, in which case a cross-reference to identified SOPs is appropriate.
- The use of automated tools is strongly recommended as these provide a more accurate, sustainable approach than manual traceability methods.
- For simple systems, an RTM is not recommended as sufficient traceability can be incorporated within document cross-references.
- For global systems, early and careful planning for traceability is required since the control and tracking of local and global requirements should be resolved.

14 Appendix M6 – Supplier Quality Planning

14.1 Introduction

Quality planning defines how a supplier fulfills the quality requirements for a new software release, project, or service introduction. Quality planning defines the approach, tools, control mechanisms, roles and responsibilities, and deliverables. Quality planning ensures the appropriate application of the supplier's QMS based on risk.

Supplier organizations work across multiple industries and are often not governed by GxP regulations. Quality agreements and/or other contractual documents define the quality responsibilities and measures to be attained by the supplier to enable regulated companies to leverage their effort and activities.

In the context of this appendix, suppliers may be external or internal to the regulated company (e.g., internal IT departments).

The output of quality planning may be defined in IT tools, a document quality plan, project plan, Service Level Agreement (SLA), or other artifact such as a release plan.

Quality planning for medical device software should consider the requirements of IEC 62304, Medical device software – Software life cycle processes [18].

This appendix is not intended to give detailed guidance on project management or project planning, which is outside its scope. The ISPE Project Management Community of Practice [51] is a forum for professionals creating a body of knowledge on project management.

14.1.1 Changes from GAMP 5 First Edition

The following changes have been made:

- Address the considerations of quality planning rather than contents of a quality plan
- Update to recognize the use of standard approaches and IT tools
- Removed project planning as this is covered by the ISPE Project Management Community of Practice [51]

14.2 Scope

Quality planning may be applied to new software product releases, introduction of new services, or client projects requiring new system development and/or configuration.

Quality planning should take account of quality agreements, SLAs, and other contractual documents defining quality requirements between the supplier and regulated company.

14.3 Considerations of Quality Planning

14.3.1 Quality Agreements

Quality agreements are established between regulated companies and suppliers to define the responsibilities and measures to be taken to ensure computerized systems and services are fit for intended use. These quality measures enable the regulated company to leverage the supplier effort and avoid duplication of activities.

Quality planning should ensure that the responsibilities and measures defined in quality agreements are addressed by the supplier’s QMS and individual quality plans where required. Quality responsibilities and measures may also be defined in contractual documents, SLAs, technical agreements, and other documents.

Suppliers may contract with multiple regulated companies and therefore quality agreements should define quality measures at an objectives level rather than imposing specific processes on the supplier. This will enable suppliers to better address the quality measures within their QMS.

14.3.2 Quality Planning Considerations

Note: Some of the quality planning considerations discussed in this section may not be relevant to smaller projects.

For customer projects, the quality plan should address any quality requirements defined in contractual documents and/or quality agreements. Similarly, suppliers and regulated companies may collaborate to define all quality requirements within a combined validation plan.

The quality approach defines the overall process that will be adopted for the project. The general approach should be defined in the supplier’s QMS, but the approach should be scalable in accordance with the solution and project risks. Such risks should include process/functional criticality, solution complexity, and novelty.

For internal software product releases, the approach defined in the supplier’s QMS should generally be followed for all new releases irrespective of whether the release is a new feature release, bug-fix release, or patch release. As such, the supplier’s SOPs may be used in place of a specific quality plan.

The scope of features to be included in a release and their priority may be defined within IT tools supporting the system development life cycle or may be defined in a release plan.

Quality planning should address the relationship between the supplier and customer QCs. In such cases, a documented quality plan aligned to the customer’s validation plan may be beneficial.

The areas to address in quality planning are described in Table 14.1.

Table 14.1: Quality Planning Considerations

Area	Planning Consideration
System/Service/Project Approach	The life cycle approach as appropriate to the project, products, and services being provided
Risk Management	Risk-management approach and plans Managing customer input to risk assessments for customer projects (understanding risks to patient safety, product quality, and data integrity)
Organization (including use of third parties)	Roles and responsibilities of supplier, customer, and third parties as appropriate Third-party governance controls (e.g., supplier assessment and oversight, roles and responsibilities, governing procedures, training considerations, contractual agreements)
Specification	Will customer requirements be integrated into the core software product or managed as custom developments? Define the process for capturing, analyzing, and recording customer requirements

Table 14.1: Quality Planning Considerations (continued)

Area	Planning Consideration
Design and Configuration	<p>How will design records be articulated (e.g., IT Tools, documents, etc.)</p> <p>For projects, customer review and approval expectations</p> <p>Configuration management approach</p>
Traceability	<p>How to ensure requirements traceability throughout the life cycle (e.g., IT tools, documents, traceability matrices)</p>
Build	<p>Source-code management and integration tools</p> <p>Software coding standards, code review and automated code verification tools</p>
Test	<p>Risk-based test planning.</p> <p>Testing roles and resources (including independence of testers from developers)</p> <p>Use of test-management tools and automated testing</p> <p>Approach to development testing, functional testing, and requirements testing</p> <p>Approach to test-defect management</p> <p>Regression analysis and regression testing requirements</p> <p>For projects, customer role in accepting the solution/service</p>
Release Management	<p>Acceptance criteria (based on quality planning requirements)</p> <p>Release authorization requirements</p>
Installation	<p>Procedures for software deployment/installation (test and production environments)</p> <p>Approach to verification of deployment/installation (installation verification test and/or automated deployment process)</p>
Data Migration	<p>Data migration approach</p> <p>Data verification approach to ensure data integrity</p>
Acceptance	<p>Supplier and customer responsibilities</p> <p>Risk-based testing strategy</p> <p>Leveraging supplier testing effort</p>
IT Tools	<p>Define IT tools supporting the project, product, service</p> <p>Consider customer access to tools for customer projects</p> <p>Assessment and control of IT tools</p> <p>Controls for ensuring data/record integrity</p>
Records/Document Management	<p>Ownership of records and documents</p> <p>Records and document management processes (including supplier/customer roles)</p>

Table 14.1: Quality Planning Considerations (continued)

Area	Planning Consideration
Deliverables	Define deliverables for product/project/service: <ul style="list-style-type: none"> • Life cycle records and information (including responsibilities) • Software products • Configuration • Customized software source code (as appropriate) • Services • Training • User and administration manuals/guides • Reports
Project Change Management	Project change process and governance responsibilities Transition from supplier processes to customer processes during handover
Handover to Support Organization	How the solution will be supported post-handover Knowledge transfer
Reporting	Progress, issues, and risk reporting Quality metrics reporting (e.g., deviations)
Project Audits	Are periodic projects reviews to be conducted, if so at what stages and by whom
Training	User and system support/administration requirements and approach Training materials (including ownership) Roles and responsibilities (customer and supplier)

14.4 Projects Involving Multiple Organizations

Multiple organizations may be involved in a project to deliver different technology solutions (e.g., IT infrastructure, applications, middleware) or may be involved in different aspects of the project (e.g., system development/integration, testing, data migration, business readiness).

Quality planning should ensure that all parties understand:

- Roles and responsibilities
- Dependencies between related activities
- Timescales for each party's activities
- Requirements to jointly participate in activities (e.g., system interface testing documents, data, equipment, test execution)
- Responsibilities for deliverables (creation, review, approval, and sharing)

15 Appendix M7 – Validation Reporting

15.1 Introduction

This appendix describes the computerized system validation reporting process. It covers the activities involved, the roles and responsibilities, and identifies where the process fits within a typical computerized system life cycle.

Accurate and informative planning and reporting are key elements of effective and successful governance, and the validation report is often the first document related to a system that is examined during a regulatory inspection.

This appendix should be read in conjunction with Appendix M1, which discusses computerized system validation planning.

15.1.1 Changes from GAMP 5 First Edition

Changes have been kept to a minimum to avoid disruption to companies that have been successfully following GAMP guidance on this topic since First Edition publication. The changes are:

- Align with updated Appendix M1 Validation Planning
- Take into account the validation of SaaS solutions
- Take into account the validation of systems developed in an incremental or iterative manner (Agile)

15.2 Scope

This appendix gives guidance on the reporting of the outcome of validation activities relating to the implementation of either a specific computerized system, or a group of related systems.

15.3 Roles and Responsibilities

Specific roles and responsibilities will vary depending upon the scope and scale of the project. These roles should be defined in the appropriate section of the corresponding plan and will cover who is responsible for the creation, the review, and the approval of the computerized system validation report(s). Any changes in roles and responsibilities that were made during the project should be noted in the report.

The quality unit is responsible for ensuring that the generated deliverable(s) comply with requirements specified in the corresponding plan, are produced in line with company policies and procedures, and meet the appropriate regulatory requirements.

15.4 Reporting Process

Where required by a computerized system validation plan, a validation report should be produced, focusing on aspects related to patient safety, product quality, and data integrity, highlighting those areas and how these aspects have been delivered and controlled. It should summarize the activities performed, any deviations from the validation plan, any outstanding and corrective actions, and a statement of fitness for intended use of the system.

The level of detail in the report should reflect the risk, complexity, and novelty of the system. For simple or low-risk systems a separate document may not be needed; applicable aspects may be covered by another document.

The structure of the report should mirror the structure of the corresponding plan, although there are some components of the plan (e.g., organizational structure) that typically will not have a corresponding section in the report unless a significant change has been made.

The report should be approved, as a minimum, by the process owner and the quality unit. It also may be appropriate for other approvers of the corresponding plan to approve the report, such as the system owner.

It is common to produce one final report. There may, however, be other reports that either feed into this document or are created after it and which supplement it.

For example, for larger systems it may be advantageous to issue sub-reports to cover phase completion (such as specific testing or verification phases) and there may be an interim report to release the system. In general, each plan should be addressed in a corresponding report and the scope of the report should correspond to the scope defined in the related plan. For computerized systems using an incremental development life cycle, for example Agile Scrum, the report would be expected prior to the initial release of the system. The approach to reporting and acceptance of subsequent releases through sprints should be defined in the validation plan.

15.4.1 Contents of the Computerized System Validation Report

This section lists topics that may be included in the computerized system validation report; not all sections or subsections may be relevant. The guidance provided is intended to be neither prescriptive nor exhaustive.

15.4.1.1 Introduction and Scope

The introduction should reflect the corresponding plan, and highlight any differences that have arisen since the plan was issued. It should contain the following information:

- Purpose and scope of the report
- Who created the report, and under what authority
- Summary of approach adopted
- Cross-reference to controlling plans, policies, or procedures

15.4.1.2 Scope Changes

It may be necessary to modify the original approach; the report should highlight and justify such scope changes.

In large complex projects the recording of such events may be centralized as part of a formal tracking system, e.g., a Risks, Actions, Issues, Decisions (RAID) log. In such cases this section of the report may reference such sources of information.

15.4.1.3 Supplier Assessment

Supplier assessment activities should be summarized, or a reference made to other sources of information, such as a Supplier Assessment or Audit Report.

Responsibility for system and related activities may be delegated to suppliers and service providers, but in all cases regulatory accountability lies with the regulated company. Regulated companies may, however, leverage the knowledge, experience, activities, and artifacts of the supplier or service provider through defined risk-based assessment, management, and governance processes.

Information available in other documents should not be repeated.

Contents of supplier assessments or audit reports should not be included.

15.4.1.4 Summary of Activities

The summary should refer to existing documentation, e.g., verification or test reports, and information should not be duplicated.

This section may include subsections relevant to each phase.

15.4.1.5 Summary of Deliverables

The report should verify that all of the deliverables noted in the corresponding plan are complete, adequate, and where appropriate approved. This includes system development documentation, information maintained in tools or supporting systems, and SOPs required for operational support.

15.4.1.6 Summary of Deviations and Corrective Actions

The report should describe any activities and results that did not conform to the expectations specified in the plan, and explain the impact, including corrective actions. Outstanding corrective actions should be highlighted and appropriate next steps identified or referenced.

15.4.1.7 Statement of Fitness for Intended Use

There should be a clear statement on the status of the system and whether it is fit for intended use, bearing in mind any outstanding deviations or corrective actions.

15.4.1.8 Training and Knowledge Management

The report should verify that personnel involved with new processes, equipment, or systems have been trained and that this training is documented. Knowledge management arrangements should be verified.

15.4.1.9 Maintaining Compliance and Fitness for Intended Use

The report should outline how the compliant status of the system will be maintained. This may be efficiently achieved by referring to relevant policies and procedures or other QMS elements. See the Operational Appendices for further details.

15.4.1.10 Glossary

Definitions of any terms that may be unfamiliar to the readership of the document should be included.

15.4.1.11 Appendices

There may be a requirement for appendices, depending on the purpose, size, and complexity of the report, and the corporate styles and policies adopted for reporting. These may include references to project-specific documentation and references to other relevant documentation such as policies and procedures, guidelines, and standards.

16 Appendix M8 – Project Change and Configuration Management

16.1 Introduction

This appendix covers change and configuration management of computerized systems during the project phases prior to acceptance and handover to operational use.

Any controlled item that undergoes review, approval, or test should be governed by appropriate configuration management, and every controlled item should be subject to appropriate change management.

Change management should be applied to each controlled item upon its first formal approval to avoid unintentional or unauthorized change. Different controlled items may require different levels of formality and rigor. The project change management approach should be documented. The project manager and the user should agree on the level of user involvement.

Project change management processes typically are simpler than those for operational GxP systems due to fewer people involved, faster communication, and lower risk to patient safety, product quality, and data integrity.

The point of transfer from project to operational change management should be clearly defined before handover to operational use.

See Appendix O6 for further details on operational change and configuration management.

16.1.1 Changes from GAMP 5 First Edition

- Added ISPE GAMP Good Practice Guide: *Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20] references to introduce Agile as a toolset to manage requirement changes, artifacts/deliverables, and for DevOps and Continuous Integration/Deployment
- Added concept of traceability of requirement changes
- Added risk assessment and discrepancy handling deliverables to change management
- Added role to include quality unit to provide guidance and approve regulatory and compliance user and functional requirement changes

16.2 Scope

This appendix applies to changes to controlled items such as documentation, application software, operating software, firmware, hardware, and system, master, and configuration data within the scope of the specified computerized system during its project phase.

This appendix is not aimed at changes to project scope, which typically are initiated and managed by change procedures forming part of project management processes, and which may have significant financial implications. This appendix is aimed at changes to controlled items that may be triggered by such project scope changes among other reasons.

ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management, Section 3.4 [20] introduces the benefits of using Agile to create and manage changes to requirements and maintain traceability throughout the project phase to release for operations.

16.3 Guidelines

16.3.1 Configuration Management

All components of a computerized system, and changes to them, should be controlled. The exact hardware and software configuration of the system should be documented throughout the life of the system. The level of formality is greater for an operational system than for a system early in its development, but the principles that apply are the same.

Configuration management should begin as early as possible during development. The more formality is introduced during development, the easier it is to document the baseline configuration for operational configuration management.

Configuration Management consists of:

- Configuration Identification (WHAT to keep under control)
- Configuration Control (HOW to perform the control)
- Configuration Status Accounting (HOW to document the control)
- Configuration Evaluation (HOW to verify that control)

Configuration management activities, responsibilities, procedures, and schedules should be clearly defined. For a large or complex project or product, a separate Configuration Management Plan should be produced.

The use of automated configuration management tools can bring significant advantages and should be considered. The selection, verification, and use of such tools should be documented and based on risk, complexity, and novelty.

See Appendix O6 for further details on configuration management.

16.3.2 Change Management

Project changes should be controlled and documented. As the project advances the formality of the change management process generally increases. This progresses from informal project team meetings and discussions, through formally recorded project meetings, to formal change management requests. The increase of rigor and formality depends on the impact on both the preceding and subsequent deliverables in the documentation set, as these are developed and linked to each other. Some controlled items may require different levels of formality and rigor. Projects should define their approach to project change management during project planning.

All deliverables should be identified so that the controlled items subject to change management may be defined. These may include:

- Planning documents
- Vendor or supplier contracts and assessments
- Requirements specifications
- Design specifications
- Quality review documents
- Risk assessments

- Test specifications including acceptance criteria
- Testing results and discrepancy handling
- Reports
- Hardware (e.g., Programmable Logic Controllers (PLCs), Personal Computers (PCs), minicomputers, servers, communication interfaces, printers)
- Developed software code (e.g., PLC code, source code, executables, data files)
- Third-party software (e.g., operating systems, firmware, library files, configurable products, drivers, compilers, virtualization software). This includes software delivered with the system and customer-supplied software items.
- Configuration files (for configurable products, alarm, and process setpoints, etc.)
- Manuals (e.g., user manuals, system manuals)

16.3.2.1 Changes during Development and Prototyping

Formal control should not be introduced too early during development in order to minimize non-productive work during what are naturally iterative or evolutionary processes. Documents should be held in a draft status during development without formal change control. Version control should track the current working draft (e.g., Agile backlog) and ensure that documents are not unintentionally modified simultaneously by different project team members.

At the end of the development phase document review and approval should act as the formal verification that the document content is complete, accurate, and fit for intended use.

Changes made during approved prototyping work are exempt and should be subject to these controls only when they become documented design proposals.

16.3.2.2 Changes to Code

Changes to code should be managed effectively to avoid unintended or unauthorized changes. The best solution is the use of an automated code management tool, reference *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management*, Section 3.5 (Tools Instead of Documents) and Section 3.6 (DevOps, Continuous Integration/Deployment, and Product Teams) [20] for use of Agile tools to manage requirement changes. These tools use a check-in and check-out process to protect code so that two developers cannot be simultaneously working on the same file, which could lead to errors and wasted effort. They also make it less likely that a developer will edit an old file, leading to loss of intervening developments or the possible reintroduction of corrected defects (see Appendix D4 for further details on management, development, and review of software).

16.3.2.3 Key Change Management Steps

Raising a Change

Any member of the project should be able to raise a change in accordance with the project change management procedure. Each change should be uniquely identified and indexed.

Change Review and Authorization

Each project should have a designated project manager responsible for ensuring that all changes to the system are implemented in a controlled manner. The project manager may delegate this responsibility.

It is advisable that the project team engage the quality unit to ensure adequate oversight of regulatory and compliance requirements.

Each change raised should be reviewed. Based on a risk assessment there should be a decision to accept or reject the change. If accepted, the activities required to specify, carry out, and verify the change should be defined, including:

- The scope of the change and which controlled items are affected, including documentation
- The impact of the proposed change and the need for further risk assessments to determine what verification is required
- The risks associated with making the change, and any back-out plans if the change fails at implementation

The review, risk assessment, the decision to proceed or the decision to reject with reasons, and activities required, should be recorded and retained as part of the project documentation. Traceability should ensure transparency of each requirement change and the resulting impact of subsequent testing or verification, as applicable.

The authority and responsibility for accepting and rejecting changes should be clearly defined.

Controlled items that have been approved by the regulated company (e.g., RS, contractual documents, tested and accepted software) should be changed only after prior approval of the change.

The quality unit should approve changes to regulatory and compliance user and functional requirements.

Each controlled item should be tracked to completion. A change plan may be required for complex changes incorporating many controlled items.

Change Completion and Approval

When the change has been implemented, documentation revised, and appropriate verification performed, the change should be approved by the project manager or nominated representative and closed.

Example forms to assist with the process of managing a change are supplied separately.

17 Appendix M9 – Documentation and Information Management

17.1 Introduction

This appendix covers the management of relevant information that is created during the development, implementation, and validation of applications and which is required to be able to demonstrate that the application is in a state of control.

17.1.1 Changes from GAMP 5 First Edition

As tools and approaches to information management have evolved the emphasis on all evidence residing in signed hard-copy documents has also evolved. Major new considerations are:

- Much information will never exist on paper, or even in the form of a document
- Some information will be created and managed throughout its lifetime in a tool
- Documentation needs to be searchable, that is, there must be some sort of robust search engine able to locate information when it is needed
- There is no need to create documents simply for the sake of having a document in case of regulatory inspection. If it is not useful for managing the application in a state of control and is not needed.

17.2 Scope

This appendix is applicable to all system life cycle documentation. The guideline principles apply to documentation in all record formats.

It describes an approach applicable to complex, business-critical projects in a GxP environment. It is not intended to be prescriptive. Simpler projects may adopt less formal methods. Suppliers may choose to adopt other approaches.

17.3 Guidelines

A procedure should be established for management of documentation covering:

- Production
- Review
- Approval (where appropriate)
- Issue
- Change
- Withdrawal
- Storage

Where documentation is produced by a supplier, consideration should be given at the planning stage of projects to agree on information management standards and ensure that regulated company and supplier expectations are aligned. The regulated company may assess the supplier approach to information management as part of the supplier assessment processes.

Documentation should be assessed for suitability, accuracy, and completeness. There should be flexibility regarding acceptable format, structure, and documentation practices.

17.3.1 Documentation Production

Documentation standards should be agreed, including the use and management of tools where information may reside, and the searchability of the information.

Where actual written documents are needed, there should be standards covering document layout, style, and reference numbering. Documents should be under version control and in draft form prior to formal issue. Draft and approved versions should be clearly distinguished, e.g., by their version identifier.

Where information resides in a database or other tool, data integrity controls should be in place to ensure that information within the tool is correct and trustworthy. Only people with a justified need to enter or change the content should have access rights. Audit trails may be appropriate so that a record exists of prior versions of the data. Depending on the nature of the data, formal change control may also be necessary.

It is critical that information can be located when needed. This may require metadata to be created for the purpose of indexing; if this is not automated, it should be done manually as part of the creation or modification of the information.

17.3.2 Documentation Review

The review of documentation should be part of the business process of system build, with the level of formality of the review based on risk. Review is typically most effective when carried out by an SME. For example:

- For test records, the most effective review will be by an SME in the business process, as they will be in the best position to evaluate whether a test result is actually acceptable
- For release of an application at the end of a validation process, the appropriate SME is the quality unit⁸, who are best placed to evaluate the overall GxP compliance

If the review results in the need for remedial actions, whatever measures were required should be assigned to a responsible party, addressed, and completed. If a formal approval for a particular record or document is required, this remediation should be completed before submission for approval.

Review reports, if used, should be logged and indexed. Individual actions noted on the review reports should be logged and progressed through to closure.

17.3.3 Documentation Approval

Formalization of the review and approval can take a variety of forms, ranging from a pen and ink signature to an electronic approval in an EDMS to a separate record where an SME attests to having reviewed and accepted the information.

⁸ This may be different for GLP systems, as interpretation of GLP rules may result in greater separation of Quality Unit activities than are expected for GMP, GCP, or medical devices.

The reason for each approval should be defined, e.g., technical approval, system release, etc. Approvals should be dated. Unnecessary approvals should be avoided since this can cause significant delays; it is extremely unusual that more than two approvals are justifiable.

Subsequent changes to approved documentation should be carried out under the applicable change-control procedure.

17.3.4 Documentation Issue

As soon as documentation is approved, or finalized if no approval is required, all indexing actions needed to ensure that the information is searchable and can be located when needed should be completed.

Indexing is critical because documentation that cannot be located has no value, and worse, may lead to compromised decision-making in its absence. Effective knowledge management needs a robust search capability, and the information discussed in this appendix needs to be readily locatable when needed. Methods exist for documents (e.g., by management in an EDMS) but the challenge may be greater for records that reside in a tool (e.g., records from Agile software development).⁹

Superseded versions of the documentation should be removed from use and clearly marked as such. These superseded versions may need to be archived, for example, if an investigation of a deviation or failure needs to reference the version that was applicable at the time. For information that resides in a tool or database an audit trail meets this need.

The procedure for managing controlled copies, if required, should be documented. Where a system of controlled copies is in operation, uncontrolled copies should be clearly identifiable.

17.3.5 Documentation Changes

Modifications to documentation should be controlled. Finalized documentation, whether in the form of an actual document or in a tool or database, should be managed under change control. Examples of tools and databases that might contain relevant documentation might include:

- Trial Master File (TMF) Systems contain important documents like the study protocol, statistical analysis plan, etc., that are frequently updated and reapproved during a study
- Validation support tools that may include living data such as traceability matrices, in addition to standard validation documents
- Systems supporting moving targets such as DR, including plans that must be updated periodically and test results

The rigor of the change-control process should be based on risk. In some tools documentation may be reset to the next draft version, updates made, and the documentation reviewed, approved, and issued in accordance with defined procedures. For low-risk information there may not be a formal approval process.

Indexing should be updated as part of this process; documentation history may be incorporated within the documentation itself or held as a separate document, and databases or tools should have an audit trail.

Modifications to approved documentation should be reviewed and approved by the same functions or organizations that performed the original review and approval, unless specifically designated otherwise.

⁹ See *ISPE Good Practice Guide: Knowledge Management in the Pharmaceutical Industry*, Section 12.5 [10] for further discussion of content management and enabling search tools.

17.3.6 Documentation Withdrawal

There should be a defined procedure for withdrawal of approved documentation, which would normally be handled through change control.

The documentation index and documentation history should be updated to indicate the documentation is being withdrawn. Any controlled copy holders should be notified of the withdrawal and those copies removed from circulation. If required to be retained, the withdrawn documentation should be archived in a manner secure from use and its status clearly identified.

17.3.7 Documentation Records and Storage

Documentation and associated information should be stored safely and securely (whether by paper-based or electronic means), and according to defined procedures. They should have standard data integrity protections against accidental and malicious damage, and should be retrievable throughout the defined retention period. Safeguards should prevent the unintended use of unapproved, superseded, and withdrawn documents.

18 Appendix M10 – System Retirement

18.1 Introduction

This appendix provides guidance on planning the orderly retirement of computerized systems. This appendix assumes that the regulated company has already made a decision to retire a system and has identified the data to be archived and/or migrated to another system(s). Throughout the data and system life cycle it is paramount that data integrity is at the forefront of the solution design; this also carries through to retirement.

This appendix offers a system-centric overview of the retirement process, with more detailed information on system retirement contained in the *ISPE GAMP Good Practice Guide: A Risk-Based Approach to Operation of GxP Computerized Systems*, Chapter 18 [34]. A data-centric view of the system retirement process, focused on the details of managing the data produced in the system via archiving, migration, and/or disposal, is contained in the *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design*, Chapters 3 and 7 [36].

18.1.1 Changes from GAMP 5 First Edition

- Inclusion of retirement process owner responsibilities
- Inclusion of data owner responsibilities
- Accounting for cloud service providers/XaaS
- Inclusion of cloud service providers/XaaS exit strategies

18.2 Scope

This appendix covers retirement planning for all GxP regulated computerized systems. The guidance focuses on the controlling computer system and associated data rather than associated controlled equipment.

This appendix gives comprehensive guidance on all aspects of system retirement planning. Not all aspects will be relevant to all systems. The extent of planning and other activities will depend on the GxP assessed risk, the business criticality, size, and complexity of the system, and other factors that may impact the future use of the data.¹⁰

18.3 System Retirement Planning

18.3.1 General Guidelines

System retirement is a process consisting of these main activities:

- Retirement – System is removed from active operations; that is, “normal operational” users are deactivated, and interfaces disabled. No data is added to the system from this point forward. “Special access” is retained for data reporting, results analysis, and support.
- Decommissioning – The controlled shutdown of a retired system. A system may be stored if required to be reactivated at a later date, e.g., for retrieval of regulatory data or results.

¹⁰ System retirement and data retention within the context of mergers, acquisitions, and divestitures should consider future use under the new entity; see *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design*, Section 3.7 [36] for more details.

- Disposal – Data, documentation, software, or hardware can be permanently destroyed. Each may reach this stage at a different time. Data and documentation may not be disposed of until they have reached the end of the record-retention period as specified in the record-retention policy.

The system retirement process should be documented in a system retirement plan, which typically should receive input from functions such as the business process owner, quality unit, system owner, and IT.

Inputs to the planning process may include:

- A risk assessment identifying and evaluating the business, technical, and regulatory risks associated with retiring a system, as shown in Figure 11.3. If the risks of continuing to operate a system are considered greater than the risks associated with retiring a system, then the retirement should proceed.
- Data and record retention and destruction requirements for historic data and records
- Identification of the current software and hardware configuration as well as interfaced systems, equipment, or instruments
- Identification of any internal and/or external systems that rely on data or records from the system
- Identification of technical and contractual constraints of cloud-based providers

The extent and rigor of planning should be based on the system impact and risks associated with ensuring data integrity (e.g., mitigating the potential for data loss).

The system retirement plan typically should be approved by the process owner, system owner, data owner, quality unit where necessary, and others as required such as the legal department.

18.3.2 Contents of the System Retirement Plan

The system retirement plan should describe the approach to be undertaken, including:

- Introduction
- Roles and responsibilities
- Overview and implications
- Business process description
- Retirement approach
- Data and record migration, archiving, and destruction
- Verification approach
- Ending system maintenance and support
- Change management
- Schedule
- Retirement execution
- System documentation

This appendix provides further detail on each of these topics; not all sections may be relevant. The guidance provided is intended to be neither prescriptive nor exhaustive.

18.3.2.1 Introduction

The introduction should include:

- Who produced the document, under which authority, and for what purpose
- Relationship with, and reference to, relevant policies, procedures, standards, and guidelines
- Relationship with, and reference to, other documents

18.3.2.2 Roles and Responsibilities

The roles and responsibilities associated with the retirement process should be documented in the plan, and should cover the business process owner, quality unit, system owner, data stewards, data owner, the retirement team and its members, and any other contributing parties as appropriate. There must be a responsible owner of the retirement process.

18.3.2.3 Overview and Implications

Consideration should be given to the effect of system retirement on aspects such as:

- Strategy – Document the impact on the overall technology strategy to include archive access controls, including further viewing of the data (and, in the case of dynamic data, reprocessing if required), and initiate any updates to documentation or other necessary actions. For example, in order to access and use archived data, the strategy may require the retention of software, inclusive of licenses, and hardware with appropriate operating system to support the use of the software. *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36] contains extensive guidance on maintaining data readability (Section 3.2), managing inactive data in an archive (Section 7.2) and the challenges of maintaining legacy software (Appendix O5).
- Process – Describe the impact on the support of the business process going forward
- Technology – The scope and boundaries of the system to be retired should be determined and documented as well as the rationale and justification for the retirement. Identify other systems, instruments, or equipment that interface with the retiring system. Data or sources of information may be in place between various systems. If the system is the location of primary records, identify the new location of the primary record (e.g., in the archive or alternative system). Include consideration for data warehouses and data lakes. Identify infrastructure components (networking, etc.) that will need to be decoupled from the system.
- Personnel – Describe the impact on the user base

18.3.2.4 Business Process Description

The pre-retirement business process should be documented and understood from the perspectives of the process, user base, and data/records. This helps to ensure that all business process impacts are identified and all angles of support or automation of the process are translated into the post-retirement scenario. The to-be scenario also should be documented and understood, especially regarding changes to business processes and/or user base, and the location of, and effect on, data/records.

18.3.2.5 Retirement Approach

A decision on whether the system will be replaced should be documented. If it is to be replaced, retirement planning should be referenced and synchronized with implementation planning for the replacement system. The approach to interface decoupling, infrastructure disconnection, ending, or transition of technical support, and any assumptions, exclusions, limitations, or dependencies, should be documented. A risk assessment should be performed to identify and mitigate potential risks introduced by resulting changes initiated by the retirement process.

System retirement is a formal life cycle phase and should be treated as such by identifying the required inputs, outputs, standards, activities, and deliverables.

18.3.2.6 Data and Record Migration, Archiving, and Destruction

The plan should identify which data should be migrated, archived, or destroyed, and the associated approval process.

The approach to data migration and archiving should be determined based on the anticipated frequency of access, the need for re-processability, and the record risk level and media robustness. Controls should be established to ensure that records and data remain secure, complete, and accurate, and that signature/record linking is preserved where applicable.

The approach should take into consideration:

- If data is to be retained, it should be backed up and stored, per data retention schedules and company procedures.
- Before the data is moved or archived from the system, the appropriate data retrieval procedures and technology should be available and tested.
- Cloud service provider/XaaS license and contractual agreements (e.g., Master Service Agreements (MSA), quality agreements) should include language that enables the migration of data to other systems, or holds data in archive in a controlled manner for a predetermined period.
- Archived data media (e.g., solid state drives) should be stored and maintained, per the manufacturer recommendations and under the required environmental conditions.
- If data and records are to be migrated to a replacement system, the migration should be planned, conducted, and verified in such a manner as to ensure data integrity. The migration procedures should be tested or confirmed before the data is completely transferred out of the system.
- For any data migration or data conversion requirements, the methods to be used for migration/conversion and verification of the data records should be defined. This may include piloting work to be done before the actual migration takes place.
- If data is to be moved to a replacement system or a cloud service provider/XaaS, the test strategy for verifying the migration should be defined. If an automated migration or conversion tool is to be used, the approach to ensure its fitness for intended use should be documented.

See Appendix D7 for further details on data migration.

For further information on the compliant handling of electronic records during data migration, archival, and retrieval, refer to *ISPE GAMP Guide: Records and Data Integrity*, Section 4.5 [35] and *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design*, Chapters 3 and 7 [36].

18.3.2.7 Verification Approach

Verification documentation needed as part of the system retirement process should be identified.

18.3.2.8 System Maintenance and Support Discontinuance

The required actions associated with the modification or ending of internal and external support agreements, operations, backup and restore, DR and business continuity plans, technical support, security and user administration, SLAs, cloud subscriptions, and configuration management programs should be planned and documented.

The retired system should also be removed from any inventory lists.

18.3.2.9 Change Management

Formal change management procedures should be followed for the retirement of a computerized system to ensure the retirement process is controlled and managed using established procedures for assessing change impacts.

Changes resulting from the system retirement should also be addressed, such as changes in support roles (technical support, superusers, etc.) and the associated training.

The approach to communicating the impact of the system retirement on affected stakeholders should be documented.

Appendices M8 and O6 contain more details on change management.

18.3.2.10 Schedule

The individual retirement tasks should be documented, along with who is responsible, the associated due dates, and any task dependencies. Critical milestones and checkpoints also should be included in the schedule.

Scheduling archive purge periods to align with record-retention requirements should be assessed to determine the potential risks associated with the purge and the appropriate timing of archive disposal. The schedule should be documented, and applicable procedures updated to reflect the schedule.

Separate project schedules may be more efficient and effective than including this information directly in the plan.

See Appendix O13 for more information on archiving and retrieval.

18.3.2.11 Retirement Execution

The timing of the retirement execution should be carefully considered. For example, this may include cutover to a replacement system (which could be phased in parallel, or a clean cutover.)

Business continuity plans should be in place in case any problems arise during the retirement or migration work. Additionally, a back-out plan is suggested, and should include detailed steps or references to configuration and reinstallation procedures to make the retired system operational again, if deemed necessary. Once the system is retired, the business continuity plans should be updated accordingly.

Any relevant documentation should also be defined.

18.3.2.12 System Records, Software, and Life Cycle Documentation

System records, software including source code for custom applications, and life cycle documentation (e.g., validation documentation, change history, system-related SOPs, etc.) should be defined. Decisions regarding whether to retain specific documents, software, and records should be based on their potential future usefulness and an assessment of the risk associated with destroying them.

Information to be retained should have a designated data owner and be placed in a defined secure location using a taxonomy that is helpful in maintaining traceability to archived records.

Affected inventories, procedures, or other documentation should be updated in a timely manner.

18.4 System Retirement Reporting

After the system retirement plan is executed, a summary (e.g., report, checklist, etc.) should be created to describe the execution and the results. If testing or verification activities were executed, the results of these tests should be summarized, and any deviations discussed along with their resolution. This summary also may include an index or registry of all documentation related to the retired system and where it is archived.

19 Appendix M11 – IT Infrastructure

19.1 Introduction

EU Annex 11 [32] states: “*The application should be validated; IT infrastructure should be qualified.*” However, due to the dynamic nature of infrastructure, it is best to think of achieving and maintaining a qualified state as an exercise of managing to compliance. This state of control is the cornerstone of qualification as expected in EU Annex 11.

A controlled IT infrastructure is a prerequisite for ensuring that GxP applications are managed in a state of control. Infrastructure plays a significant role in ensuring applications are on a stable platform with reliable required communications. The IT infrastructure supports application performance and availability as well as the integrity, availability, security, and confidentiality of data. Many of the processes required to ensure this rely on some aspects of infrastructure management, such as cybersecurity, load balancing, backup and restore, disaster recovery, etc.

19.1.1 Changes from GAMP 5 First Edition

This appendix was originally identified as Appendix S5. It dealt only with managing quality in an outsourced environment. While this is certainly still an important aspect, infrastructure management is a process that inevitably involves both internal and external processes. This significantly revised appendix applies current risk-based thinking on good practice for managing infrastructure that resides within a regulated company’s own facilities as well as ensuring those of external suppliers, for example, for cloud-based suppliers of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [20].

As a result, Appendix S5 has been withdrawn and the revised material renamed as Appendix M11.

19.1.2 Further Reading

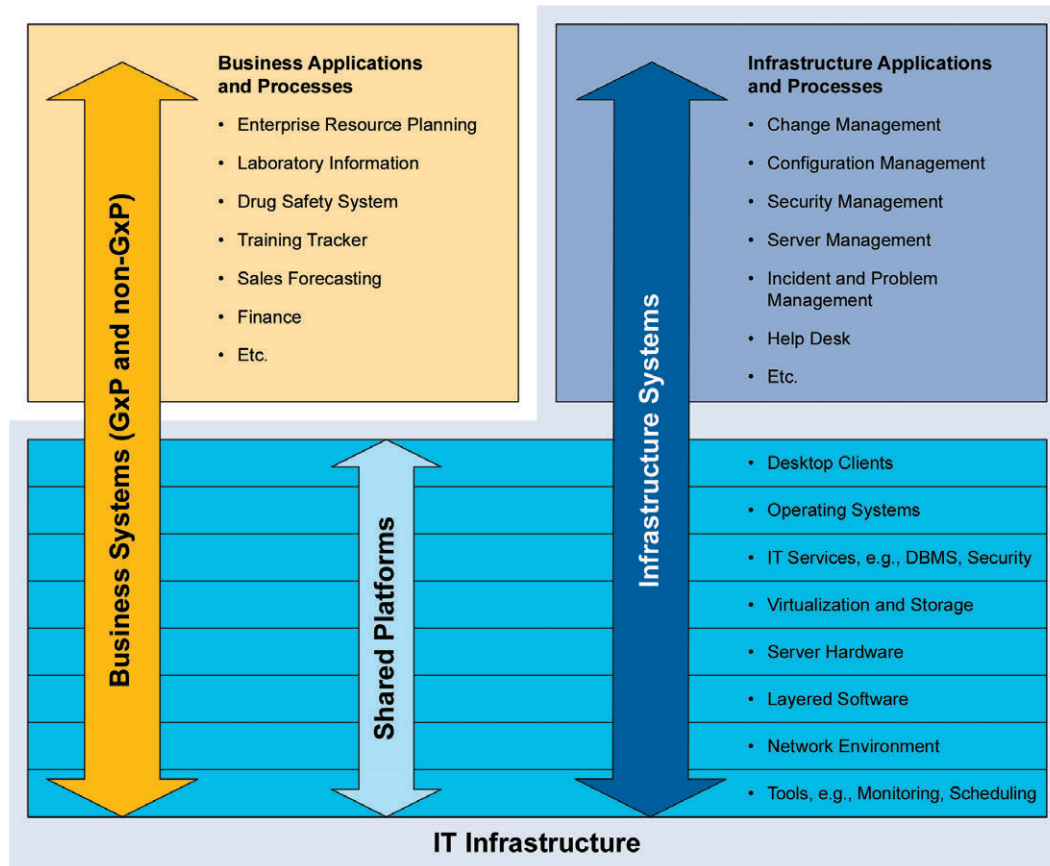
Managing IT infrastructure is discussed in more depth in the *ISPE GAMP Good Practice Guide: IT Infrastructure Control and Compliance (Second Edition)* [49], and additional background for managing outsourced services can be found in Appendix M6 in this Guide. Infrastructure management processes have become more automated, reducing the incidence of human error by substituting verified automated processes for manual ones. This topic is still evolving, as IT infrastructure management embraces increased automation and de-emphasizes reliance on paper records only.

19.2 Principles and Assumptions

19.2.1 Scope of Infrastructure Qualification

Figure 19.1 illustrates the scope of infrastructure considerations and how some of the business processes within IT need to be considered as part of a qualified infrastructure.

Figure 19.1: Relationship of IT Infrastructure and Processes with Business Systems and Processes
Adapted from ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management [20].



19.2.2 Infrastructure Qualification versus Application Validation

A regulated company approaches implementing an application to meet their business needs by developing requirements, configuring the application as necessary, performing risk-based verification as needed, establishing appropriate operational controls, and managing it to a controlled state (these combined efforts comprise validation). This is far simpler for most, but not all infrastructure elements. The company has no input into the design and manufacture of components such as servers, network switches, etc., which are effectively off-the-shelf purchases. After the initial component build has been verified to perform acceptably, new components are simply configured to the standard, and testing of each build may be extremely limited (e.g., verifying that a network node can reach another node) or absent (e.g., by using a review-by-exception approach – see Section 19.2.4). Some components are initially verified but may also require additional configuration and subsequent verification of that configuration.

Infrastructure, which is continually evolving to meet business needs, must be managed to a controlled state. Both confirmation of components' fitness for purpose and management to a controlled state are likely to involve automated processes.

The greater dynamism of the infrastructure, combined with the widespread use of standardized components that use the “one qualification, many implementations” model is the primary difference highlighted as noted above in EU Annex 11 [32].

Note also that infrastructure software is GAMP Category 1, and as such is qualified, not validated.

19.2.3 Component-Based Nature of Infrastructure

Infrastructure is generally comprised of many instances of certain standard components. For example, companies have many network switches, servers, and storage devices. These components are generally deployed in standard basic configurations. If these standard configurations have been demonstrated to perform acceptably, it is a reasonable assumption that additional components deployed in those configurations also will. This indicates that IT infrastructure presents lower risk than applications.

19.2.4 Infrastructure Automation

Infrastructure as Code (IaC) is a means of provisioning and deploying infrastructure using DevOps processes. IaC enables organizations to automate the provisioning of infrastructure, reducing the risk of human errors. Infrastructure code is subject to configuration management ensuring that all code changes are traceable. Infrastructure code development is subject to risk-based software development practices that ensure code is developed in accordance with a life cycle approach including verification prior to deployment. Configuration management and deployment tools allow for review and authorization prior to deployment of new/updated configurations, use of templates to ensure maintenance of consistent configurations across the infrastructure landscape, and monitoring of the configuration baseline to detect unauthorized changes. A further benefit to this being that a “one qualification, many deployments” approach can be taken, particularly with standard virtualized environments.

The configuration of standard infrastructure components is part of the creation of basic building blocks that are the foundational platforms for applications. Such configuration processes are readily automated. For example, an out-of-the-box server can have the operating system, middleware, layered software, and security policies loaded via an automated process. There may be multiple similar automated build scripts, for example, to build servers with different database engines. These automated builds can produce various log files or even perform simple automated tests that demonstrate a successful build, and a review-by-exception process similar in principle to that used for pharmaceutical manufacturing obviates the need for further verification. Critical thinking must be applied when deploying such “standard” environments, however. In some cases, greater control may be warranted; for example, cybersecurity risks may differ from one example to the next, which might merit hardening of the security for an otherwise standard component.

19.2.5 Tools

There are many automated tools that help IT to operate in an effective and controlled manner, for example, a Configuration Management Database (CMDB), or scheduling tools that initiate routine processes like backup, data transfers, or ETL¹¹. There are also service management tools that ensure adherence to IT quality processes and standardization of information supporting these processes, such as IT service management tools that process and catalog customer service requests, incidents, changes, problems, and other services. Such tools do not require validation, although proper installation, use, and a state of control for them is required. (See Appendix D9.)

19.3 Risk Management and Infrastructure

Given that IT departments service the entire enterprise and not just GxP areas, cases may occasionally arise where a risk to the enterprise as a whole requires a change that cannot go through a time-consuming change control process that includes sequential evaluation, approval, testing, and release. For example, if a zero-day security vulnerability¹² is recognized that threatens serious harm to the regulated company if it is exploited, remediation will be done as rapidly as possible. It is usually the case that risk from cybersecurity threats directly correlate to the data integrity considerations of GxP risk. See Section 19.4 for further discussion of QA roles.

¹¹ ETL is a tool that extracts, transforms, and loads data.

¹² A zero-day vulnerability means there are known exploits of the security flaw.

IT risk priorities therefore tend to focus on risks to system availability, performance, and information security. These priorities, however actually do address GxP concerns: applications are available when needed, they work well, and there is data integrity.

As Reid and Wyn [52] observed:

“Threats to the IT Infrastructure environment largely come from cyberattacks, unauthorized access, system and component failure, or inadequate re-source provisioning (storage capacity, processing capacity). These risks are continuous, and it is therefore imperative that the currency of IT infrastructure controls is maintained (e.g., through security patching) and monitoring is in place to provide early detection of any threat. IT infrastructure design incorporates a high degree of resilience that mitigates both single-point and complete failure.”

Another risk may come from authorized but uninformed decisions to alter system-wide parameters that may affect GxP systems or data. This highlights the need for training and SME oversight of the infrastructure staff.

Because of the high degree of resilience noted above, the GxP risk for properly managed infrastructure is relatively low; however, with many organizations adopting cloud-first strategies, the evaluation of “as a Service” vendors can influence risk both at the system and enterprise levels. This can be a starting point for risk management considerations for applications, although it is possible that non-standard implementations where a hostile environment or a single point of failure could raise infrastructure risk.

19.4 The Role of Quality

The quality unit can play an effective role in the development and manufacture of medicinal products and medical devices largely because staff are familiar with the operation of those areas. While they may not be SMEs, they typically understand the processes. This is not the case for IT. Add to this that infrastructure groups manage many non-impact changes and that the risk to patients for IT infrastructure decisions is typically extremely low, and heavy quality unit involvement in infrastructure is unwarranted and potentially extremely inefficient. There may be exceptions where greater involvement could be warranted: for example, SaMD may have greater potential direct-patient impact if there is a failure of underlying infrastructure.

The preamble to the US FDA 21 CFR Parts 210/211 [53] states:

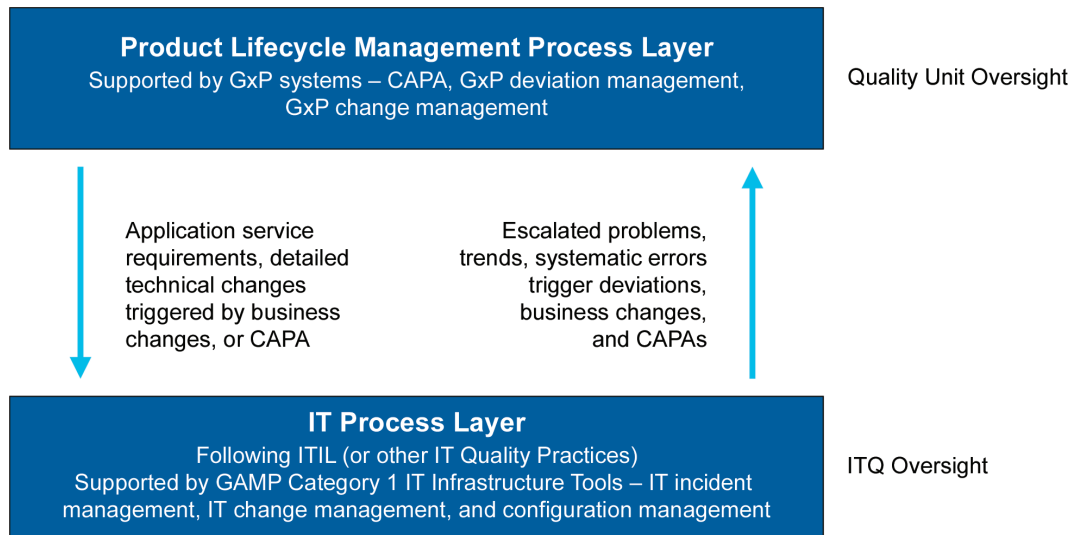
“Functions that are properly those of the engineering department or other specialized units because of their unique training and experience should not be duplicated or usurped by the quality control unit. Where expertise is in other units, the responsibility of the quality control unit is to assure that such expertise has been utilized.”

This is not to say that there should be no quality oversight. There should be an agreement between QA and IT system owners on an IT quality framework. This should define what fitness for purpose means for infrastructure. It is advisable that IT organizations establish an internal IT Quality (ITQ) function. This function should oversee IT processes to ensure that they support a state of control, and they should be familiar enough with GxP expectations to know when the quality unit should be involved in a decision. For example, it may make sense for them to be involved in a change of database engines used in multiple GxP applications, or when a major operating system upgrade is undertaken. The quality unit should not need to worry about less impactful changes like microcode updates or the installation of new virus definitions. If the quality unit is involved, their overall aim should be to ensure that quality processes are followed and that there is adequate evaluation of risk and controls. They should not be involved in the technical solution or decisions because they most likely lack the expertise.

There should be cooperative interaction between ITQ and the quality unit.

Figure 19.2 shows the relationship between the product life cycle quality processes and ITQ processes.

Figure 19.2: Interaction of ITQ and the Quality Unit [20]



There will be many changes and other activities that only merit verification by an SME; allocating additional storage space is a good example where the risk is negligible and process efficiency should be the prime consideration. In general, far more routine IT changes fall into this category as opposed to those meriting formal quality unit review. IT procedures should clearly define circumstances where quality unit involvement is warranted. These should be limited to situations where significant risk to applications or data is possible, for example, for significant functional application upgrades or overhaul of the core database engine.

Responsibility for any quality decision should be procedurally defined.

19.5 IT Processes in Scope for Infrastructure Qualification

Several processes managed by IT are key to ensuring a controlled state of applications. There are excellent resources available for management of these IT business processes, such as ITIL¹³. Some of these processes will require input from the business users of applications. For example, business input is needed to define expected service levels: help desk hours, response times, etc. In some cases, costs may need to be negotiated.

19.5.1 Security Management

Critical to data integrity within the GxP scope are several security-related processes:

- Access control and password management – ensuring that only authorized people have access to applications and data, including a process such as periodic review to ensure that individuals no longer needing access have their privileges revoked
- Management of administrative access – to minimize elevated privileges
- Cybersecurity – including data segregation, firewalls, intrusion detection and prevention, antivirus, and expediting management of vulnerabilities
- Management of encryption key(s)

¹³ For an overview of ITIL see www.ibm.com/cloud/learn/it-infrastructure-library [54].

It should be noted that IT will approach data integrity with a generic enterprise-wide solution unless they are informed that additional protections are needed. It is the responsibility of the business process owner and/or the data owner, possibly with assistance from the quality unit, to assess the supporting infrastructure processes and request modification if necessary.

19.5.2 Backup and Recovery

Processes need to include analysis of the required frequency of backup (both of applications and of the data) to meet business needs; the process for executing backup; and the process for recovering a backup copy, including who is authorized to request a recovery. It is vital for GxP applications that both Recovery Point Objectives (RPO, which dictates the frequency of backups) and Recovery Time Objectives (RTO, which dictates the interval between system failure and recovery), are defined and testable at a frequency commensurate to the system risk and agreed by the business owner. All backup processes should be periodically verified to demonstrate that they are recoverable and meet the current business need. Problems or failures should be reported back to the data owner. It is an obvious, yet often overlooked point, that the importance of periodic testing is to ensure that problems are found proactively, and not when the integrity of GxP data is compromised by a failure.

19.5.3 Archive and Restore

Processes should include how and when data (including metadata) is moved to an archive, how archived data can be accessed and by whom, and how to destroy data no longer needed, including authorization to do so. Data needs to be readable throughout the retention period. This can require attention from both the business area and infrastructure support, including for issues like data migration or support for older versions of a database engine.

19.5.4 Change Management

Change management within the IT infrastructure requires a controlled and defined process; however, it cannot be a direct parallel to change-control processes for GxP applications. Infrastructure changes should be managed at the IT process layer and not at the product life cycle management process layer. For most infrastructure changes, GxP risk is extremely low. There are also a relatively large number of changes within infrastructure, some of which will be done on a routine schedule, and are best managed through other processes (see below). The role of the quality unit versus ITQ as noted in Section 19.4 should be defined for change approval.

There should be differentiation between actual changes and activities such as repair (including like-for-like replacement), maintenance, and system administration, which will be managed under their own processes. Further, the ITIL [5] concept of standard changes can be extended into GxP infrastructure management. A list of pre-agreed changes that IT can make without the need for additional oversight can maintain the infrastructure's compliant state while ensuring operation efficiency. Provisions for emergency changes should also be made. For example, when a database administrator notices that an application needs more table space, allocation of that does not constitute a change.

19.5.5 Configuration Management

Configuration management is both a regulatory expectation and necessary for running an effective IT operation. Configuration management encompasses both the configuration of applications running on infrastructure as well as the hardware and infrastructure software itself. In a modern IT environment, it is impractical to try to achieve this without a CMDB.

19.5.6 Disaster Recovery and Business Continuity Planning

Clearly a shared responsibility with business process owners, the IT infrastructure team needs to ensure that their disaster response is adequately planned, resourced (both facilities and staff), and tested. This may include leveraging cloud resources.

The roles and responsibilities of business and IT groups should be defined, and the business process owners of the GxP application should understand the benefits, constraints, and limitations of the agreed IT service delivery. This is a regulatory expectation, e.g., in EU Annex 11 Section 3.1 [32].

GxP process owners and system owners need to understand that this is another case where enterprise risk priorities may supersede GxP priorities. While it may seem counterintuitive to prioritize restoring manufacturing after finance systems, if the business may fail without the latter, they must be handled first. There should be an enterprise-level prioritization for this, and it should be shared with all involved parties. If there is a potential public health impact, for example, a drug shortage, that should be part of the consideration for prioritization.

DR and BCP processes should be defined, documented, and practiced. See Appendix O10 for further discussion.

19.5.7 Other IT Processes

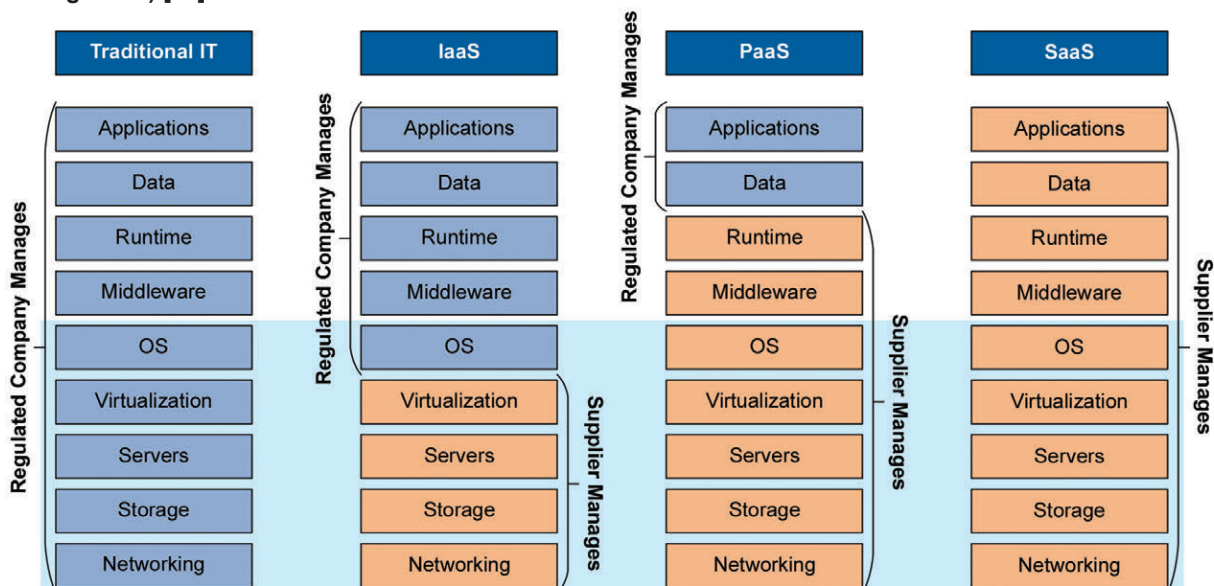
There are other processes that are often managed by IT infrastructure groups that are not limited in scope to infrastructure but do bear some relevance. Examples include:

- IT service desk
- Incident and problem management (including CAPA where relevant)
- Service level management
- Asset management

19.6 Cloud Infrastructure

Any use of cloud resources brings infrastructure qualification considerations into scope. Figure 19.3 shows the various levels of delegation of control to the supplier for IaaS, PaaS, and SaaS deployments. In all cases, however, the infrastructure management and control expectations are constant, and are represented by the blue background. In both PaaS and SaaS, all infrastructure activities are delegated to the supplier, while it is shared in IaaS, and of course resides solely with the regulated firm, for a traditional deployment in their own data center.

Figure 19.3: Customer and Supplier Responsibilities for Cloud Implementations (infrastructure within blue background) [20]



The difficulty with cloud suppliers lies in the fact that, with very few exceptions, they are not GxP regulated. This does not mean that they are universally unsuited to host GxP applications. The core question is whether they have an adequate state of control over their infrastructure. This can be determined via an effective supplier evaluation process (see Appendix M2), a robust set of contractual agreements on service levels, quality, and monitoring. Such agreements should have agreed escalation procedures for incidents and problems.

When outsourcing a GxP system to a third party it is vital to understand where the data is resident and to ensure that the service provider has taken adequate steps to ensure their delivery is robust. A SaaS or PaaS provider, as an example, can only offer as much uptime and performance as their own infrastructure provider can accommodate. The higher the risk of a cloud-delivered GxP application, it is expected that its associated infrastructure is more robust (regional mirroring versus tape backups as an example).

Data integrity should be a factor in any decision to manage GxP data in the cloud. Considerations involving infrastructure include:

- Access management: What are the implications (if any) if supplier staff sees data?
- Encryption: If data (in motion and/or at rest) is encrypted, who manages the key?
- DR: What happens if the cloud supplier has a major incident? RTO and RPO must be agreed.
- Certifications: Which (if any) certifications does the “as a Service” provider hold? e.g., ISO 27001 [44], SOC 1, SOC 2 Type 1 or Type 2 [55], HITRUST® [56], etc.
- Frequency of vulnerability scans and third-party penetration tests
- Local, regional, and global redundancies and segregation
- Deployment model and service model compatibility with the level of GxP risk

Cloud service suppliers are not GxP regulated, and it is the accountability of the regulated organization using such services to ensure that quality processes provide an equivalent level of assurance that patient safety, product quality, and data integrity are protected.

19.7 Continual Improvement

For all IT infrastructure, but especially for cloud implementations, there should always be the question: “What could we be doing better?”

Key processes should be monitored by the service provider and adherence to SLAs agreed by the regulated customer, along with dialog between the parties as to the areas that need improvement, are performing adequately, or are working very well. Including specific Key Process Indicators (KPIs) in the monitoring is recommended.

20 Appendix M12 – Critical Thinking

20.1 Introduction

This appendix discusses the concept of critical thinking to proactively optimize the approach taken to ensure quality and compliance of computerized systems (i.e., better development, testing, operation, and maintenance) within the context of the business processes they support. Further detailed discussion about critical thinking can be found in the *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20] and *ISPE GAMP Guide: Records and Data Integrity* [35].

While GAMP 5 promotes a risk-based approach to ensuring fitness for intended use, some practitioners do not apply sufficient thought to ensure the approach they are taking is customized and proportionate to the needs of different systems. The use of rigid tables, overly prescriptive templates, and tick-in-the box methods impedes critical thinking and could inhibit innovation and the adoption of new technologies. Wasting time and effort on non-value-added activities can lead to insufficient or excessive work with potential budget overspend and delays, and may reduce focus on more valuable and essential quality activities.

20.1.1 Changes from GAMP 5 First Edition

This is a new appendix.

20.2 Scope

Critical thinking should be applied in a holistic manner to the entirety of the business process that the computerized system supports. The role of ancillary equipment and interfaces should be included in the scope of implementation, validation, and operational activities to avoid missing potential risks to patient safety, product quality, and data integrity.

The interplay between system and data life cycles should also be subject to critical thinking and QRM. Multiple systems may be involved in supporting a single data life cycle. For example, data may be created and processed in one system, reported and used to make GxP decisions in an ERP system, and then archived for the retention period in another.

Operational compliance and data integrity is highly dependent upon personal behaviors. Critical thinking will play a key role in addressing human factors through effective behavioral, procedural, and technical controls, recognizing that the strength of quality culture varies across different locations driven in large part by geographic values and local historical context.

Interdependencies with suppliers and service providers, including in-house suppliers, are a further consideration within the scope of critical thinking. The supplier's application of their own QMS to the development and release of their software/service can influence business process risks and whether or not the software/service functionality is fit for the regulated company's intended use.

20.3 Guidance

20.3.1 Facilitating Critical Thinking

For critical thinking to be used successfully, the QMS and validation policy must permit and encourage its application. Examples of this are:

- Where templates are provided, they are used as an aide memoire to ensure the relevant areas for the particular system are covered rather than as a form to be completed.
- An addition or alternative to templates can be provided to cover essential elements that must be captured or defined as part of the activity, leaving the format and structure open to optimization for the particular system or tool used.

Sections 20.3.2 – 20.3.12 provide examples of how critical thinking can be applied to improve efficiency for the activities within the system life cycle.

20.3.2 Planning

A business process maybe supported by multiple computerized systems. Planning may involve adding a new system, updating an existing system, or consolidating multiple systems into a single enterprise system. When planning implementation, a full understanding of the regulated aspects of the process to be supported, including the intended use of data within the process, is fundamental.

Critical thinking should be applied to understand the implementation risks as well as functional risks to patient safety, product quality, and data integrity. Business process flowcharts illustrate the business activities, decision points, and subprocesses, while the data flow diagrams identify the creation, processing, review, use, and archiving of data. The data flow diagram should identify the primary record for regulated data if the data resides in more than one system or location. Together these diagrams help in the understanding of the business process to identify risks to patient safety, product quality, and data integrity associated with the process. Data integrity cannot be achieved without a complete understanding of the data flow. Developing business process maps and data flow diagrams is described in *ISPE GAMP Guide: Records and Data Integrity* [35] and the *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36]. Planning for individual systems is discussed in Section 4.2.1 of this Guide.

Examples of planning using critical thinking include [20]:

- Utilizing business process mapping and data flow diagrams to understand where the computerized system will fit in the process, what regulated data will pass through it, and what part of the regulated data's life cycle the system will support. The process maps and data flows do not need to be contained within a particular document but should be current and accessible and referenced by the system and life cycle information when needed. The use of process mapping and diagramming tools is encouraged.
- Defining consistent nomenclature for use in the process to facilitate data transfer, trending, and analytics.
- Selecting a solution that best fits the business requirements minimizing configuration and customization, recognizing that it may be more pragmatic to adjust the business process to fit a standard application.
- Understanding the interfaces needed to other systems, and what standard interfaces are available versus developing new interfaces (with their additional test and management burden).
- Planning for how the system should respond in an error or failure situation and how it can recover from such a situation, as well as how it should behave during normal operation.

20.3.3 Quality Risk Management

Critical thinking should be applied to ensure that [14]:

- *“The evaluation of the risk to quality should be based on scientific knowledge and ultimately link to the protection of the patient; and*
- *The level of effort, formality and documentation of the quality risk management process should be commensurate with the level of risk.”*

In the context of computerized systems, scientific knowledge is based upon the system specifications and the business process being supported.

Critical thinking should be applied to ensure risk assessments are as effective as possible and reflect the system requirements and the business process. When assigning a rating for severity of harm to a potential failure, it is important to consider the overall risk of the system to patient safety, product quality, and data integrity. A complete failure to meet a requirement may render part of the system non-functional, but if the overall system remains fit for its intended use with no impact on patient safety, product quality, and data integrity then this might be acceptable. There could, however, be good reason for classifying such scenarios as high risk if they adversely impact the organization's other operating imperatives (e.g., productivity targets).

Similar functions relating to a specific area of functionality may be grouped together in a recursive hierarchy consisting of major and subsidiary functions and controls. This hierarchical approach, which may have multiple levels depending on the complexity of the process or system, may help simplify risk assessment.

Critical thinking should determine whether some or all of the more detailed subsidiary functions need to be individually assessed within the hierarchy of functionality and controls. A subsidiary function cannot have a higher risk than that of the overarching function, so there may be little or no benefit to assessing subsidiary functions of a low-risk overarching function.

Assessing the risk at the major and/or subsidiary level as needed ensures the selection of optimal control measures, testing rigor, and documentation commensurate with the assessed risk. Risk mitigation may involve behavioral, procedural, or technical controls to reduce risk to an acceptable level. These controls may be part of a computerized system function, or in parallel manual procedures, and may be upstream or downstream of the system. Controls are typically aimed at:

- Eliminating risk through process or system redesign
- Reducing risk by reducing the probability of a failure occurring
- Reducing risk by increasing the in-process detectability of a failure
- Reducing risk by establishing downstream checks or error traps (e.g., fail-safe or controlled fail state)

The rigor and extent of controls should be based on the level of risk. For example, application timeout and password complexity are both controls to reduce the risk of unauthorized access to a system. Neither the password complexity nor the system timeout functions will have greater impact than preventing unauthorized access, which is above them in the hierarchy. The duration of the timeout and the extent of password complexity may nevertheless be influenced by the assessed risk, with shorter timeouts and greater complexity where the risk is high.

Testing should verify that controls are appropriate and sufficient. Identifying subsidiary functions with lower risk prevents the need to automatically test all subsidiary functions to the same level of detail. Combining this approach with a testing strategy based on increasing test rigor and documentation with increasing risk ultimately results in efficiency gains.

20.3.4 Requirements

The key to the implementation of a computerized system fit for intended use is to thoroughly understand the business process and data requirements to enable the definition of requirements. Not all requirements need to be finalized before proceeding (for further information on Agile approaches see Appendix D8).

Critical thinking should ensure that requirements specifically relating to regulatory compliance are tailored to the system's intended use rather than indiscriminately applying every potentially applicable regulatory reference when some are not appropriate or necessary. SMEs should be consulted on when and where particular regulations are applicable. For example, critical thinking should be applied to identify GxP regulated records that need an audit trail rather than defaulting to all data requiring audit trails regardless of context. Critical thinking can then be applied again to evaluate what is needed in terms of a data audit trail taking into account whether users are expected to create, modify, or delete regulated records during normal operation.

If an audit trail is required, critical thinking should be used to develop requirements (including Agile user stories) for an audit trail that is contextual, searchable, filterable, and reportable. In this way, critical thinking enables the development of more efficient and effective controls. The same considered approach should be taken to assess individual subsidiary requirements of an overarching requirement to better scale the controls and the testing effort.

Requirements Management and Traceability Tools

Requirements management and traceability tools can make development, implementation, and maintenance of systems more efficient and less error prone. Such tools provide a collaborative workspace that enables the definition of requirements and the development of test cases in tandem with requirements coding, and tracks issues and defects. Relational databases within these tools can be leveraged to manage the traceability of the requirements to validation activities throughout the life cycle.

Critical thinking should be applied to determine and document the adequacy of these tools. Where sufficient detail and approvals are contained and available within the tool, there is no benefit to patient safety, product quality, and data integrity for manually creating separate documentation as audit evidence. Some tools allow reports to be generated in common portable formats (e.g., requirements, test outcomes, and traceability matrices) on request. Good IT practice should ensure that the data/records held are controlled, secure, and available. Further advice on the application of tools can be found in Appendix D9.

20.3.5 Supplier Assessment and Selection

Critical thinking should be applied to the assessment and selection of suppliers, whether it be for the provision of software, a system or service to affirm is it fit for the regulated company's intended use. The assessment should cover both software functionality and its development. It is important to understand the origin of the supplier's software modules and libraries, which is why it is becoming increasingly common to request a Software Bill of Materials to confirm the use of OSS. In the case of "as a Service" offerings, supplier assessments should include assurance of the long-term provision and stability of that offering. The potential need for an audit of the supplier (initial and ongoing) should be based on risk.

Knowledge of the processes followed by the supplier can enable the regulated company to reduce their validation effort by leveraging supplier activities, as described in Section 8.3 in the body, and in *ISPE GAMP® RDI Good Practice Guide: Data Integrity – Key Concepts* [57]. Supplier activities should focus on ensuring their product meets their specifications, recognizing there will be a minimum level of detailed information and evidence of effective testing expected by a regulated company. For example, critical thinking would identify that ineffective testing approaches instead of robust and/or automated testing may reduce the opportunity for the regulated company to leverage the supplier's activities. Where the supplier has used a risk-based approach, the regulated company should determine if their assessment of risk is different to that of the supplier and adjust accordingly. The supplier's assessment may have been based on generic risks, for example, if function "x" fails it will negatively impact data integrity; the regulated

company may need to consider the specifics of their use of the system based on their own business processes, for example, considering the criticality of the data or function impacted by the failure on patient safety and product quality. Appendix D5 provides further discussion on how regulated organizations can leverage supplier testing.

Critical thinking should also ensure that supplier selection criteria contain performance measures such as system reliability, service continuity, and reputation for customer responsiveness. For certain types of system architecture, such as SaaS, reliance on the supplier may not be limited to the system life cycle and can include the data life cycle, which should be assessed during service provider selection. Where there are frequent changes, there may be little to no opportunity for the regulated company to reject the change, which reinforces the need for adequate supplier change control. Further discussion of how critical thinking can be applied to reduce user validation of COTS products has been published elsewhere; see *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20].

By effectively leveraging supplier and IT/IS service provider effort, the regulated company maximizes value and avoids duplication of effort by ensuring activities are undertaken by organizations with the prerequisite skills and experience.

Critical thinking should be applied to ensure that:

- The supplier appreciates how their product is used in the life sciences industry and the implications this may have on their development and support practices.
- Regulated companies are accountable for ensuring the system is fit for intended use and should evaluate the supplier's approach against the regulated company's intended use. Access to supporting information around supplier activities should be defined in the purchase contract or SLA.
- Leveraging is enabled through a risk-based approach to IT/IS service provider assurance that ensures service providers adopt the necessary controls and records within their QMS. This demonstrates that services and solutions are fit for intended use and data integrity is maintained. This does not imply that service providers must provide the regulated company with documentation and records maintained as part of the regulated company's QMS. The value of service provider documentation and records is within the service provider's QMS where they support and demonstrate the effectiveness of a service provider's processes and controls. This does not preclude service providers from providing copies of documentation and records where this facilitates an effective and efficient approach to assurance.
- Supplier deliverables specifically produced for regulated companies should be assessed for suitability, accuracy, and completeness. There should be flexibility regarding acceptable format, structure, and documentation practices as there is no value in recreating documentation in a different template.
- Information and test evidence in the form of artifacts within requirements management tools and automated test tools have the same status as formal documentation.
- If the supplier's system is to be interfaced to other systems within the regulated company, for example LDAP¹⁴ authentication or single sign-on, the interfaces need to be defined, implemented, validated, managed, and maintained.
- Where feasible, the supplier should aim to develop customer-requested new functionality into their mainstream product offering where this is beneficial to the wider user base rather than creating special versions for individual customers. This reduces the supplier's product management effort and simplifies future updates for all parties. If the functionality is provided as a configurable option, there is no obligation on other customers to implement it. Critical thinking should ensure there are no unintended consequences such as reduced system performance when such changes are made to the mainstream product.

¹⁴ Lightweight Directory Access Protocol

- Where a regulated company requests the development of a new or amended feature, keeping it within the supplier's development life cycle ensures it is done once and tested once rather than developed and tested repeatedly by regulated companies.
- Reuse of standard modules of code or custom-developed forms, by the supplier or regulated company, should be based on a master template to ensure changes are automatically replicated across all instances. This reduces the possibilities of errors (e.g., introduced by copying and pasting for reuse), condenses the need for testing to just the master template, and eliminates the possibility of one or more instances not being updated.
- Ongoing management and implementation of change is done in a controlled way by:
 - Leveraging supplier activities and knowledge (i.e., in the same way as that for the original development). Care should be taken to recognize instances where the supplier support transitions between project teams and service teams as this may impact the standards being applied and the supplier assessments conducted by the regulated company.
 - Ensuring changes to the product can be communicated to the regulated company so they can assess the risks of taking or not taking the change. Note that in some offerings the change is not optional for the regulated company, e.g., multitenant SaaS.
 - Having a pre-verified method for installing the change. The regulated company should apply critical thinking around the risks of taking the change versus not taking the change, assessing likely risks to their intended use as a result of the change, and leveraging what has been done by the supplier in order to install the change with the minimum of repeated effort.
- Where a supplier is involved within the regulated company's life cycle (e.g., a systems integrator), the supplier and regulated company should jointly agree on the overall rigor of documentation and testing required and who will be responsible for which elements in order to avoid unnecessary duplication.
- Completing arrangements needed to support ongoing regulated company compliance, including ongoing supplier assessments/audits where these are envisaged, and ensuring that information on supplier assessment and management processes and conclusions is available for review.

20.3.6 Testing

Effective testing is crucial when determining that a computerized system is fit for its intended use (see Appendix D5 for guidance on testing). Critical thinking should facilitate the testing of functions and features with appropriate risk priorities that reflect:

- The potential impact on patient safety, product quality, and data integrity (higher-risk functions may require greater test rigor and level of documentation).
- Prior testing of the function or feature either in the supplier's development life cycle or as part of the regulated company's life cycle. Automated tools within the supplier's development life cycle may have enabled and included full regression testing of daily builds.
- The degree of confidence in that prior testing based on supplier assessment.

Using critical thinking, assessment of risk can then be used as input to determine the appropriate test approaches. The aim of testing is to identify and allow the removal of defects and confirm fitness for intended use rather than producing documentation for documentation's sake. Critical thinking can optimize test approaches for the regulated company such as [20]:

- Planning and organizing tests to run as efficiently as possible, for example, by combining tests to minimize repeated test setup activities, and/or by grouping related functionality testing.

- Ensuring sufficient test coverage during the system life cycle, with the rigor of testing commensurate with risk, and avoiding repeating tests that are similar or identical to those carried out by others.
- Ensuring that the test cases demonstrate that functions operate correctly.
- Differentiating between proving steps and non-proving steps to limit the amount of test-execution evidence created and retained. Proving steps demonstrate a requirement has been fulfilled, while non-proving steps are used to set up the proving step. This ensures test evidence is only captured for proving steps that demonstrate a higher-risk GxP or business requirement has been fulfilled.
- Ensuring the testers and reviewers have the skills and expertise to execute the tests, operate the system, and evaluate the results in the context of the intended use and/or the business process. This then obviates the need for overly prescriptive and detailed test instructions and in turn reduces test incidents arising from poorly written test scripts.
- Minimizing pressure on testers, e.g., to meet a deadline for moving to the operational phase, which can bias testing outcomes.
- Enabling a proportionate review of completed tests based on the risk of the function under test.
- Leveraging automated test tools and test-management tools in place of extensive manual effort and referring to the test artifacts within the tools in place of documentation.
- Managing incidents and their corrective actions to ensure any changes are fully verified once completed, including an appropriate amount of regression testing to ensure that the change has not adversely affected other functionality.

The level of detail for test cases and test evidence can vary depending on the test strategy and the risk posed by the function or feature being tested. Critical thinking should be employed to ensure excessive hard-copy test evidence is not requested. In most cases, screen shots and excessive commentary do not add value and are unnecessary.

Test strategies should ensure sufficient system testing to detect defects, and that test-management processes are robust enough to maintain control of the system during the testing activities. The scheduling of testing should take account of new and revised risks, design changes, and other factors that could impact the timing of this activity. Unusual patterns of test failures associated with particular authors and/or individual testers should be investigated to determine whether there are any wider testing implications on the rigor of completed testing. The critical-thinking rationale behind the test strategy must be documented.

20.3.7 Managing Build and Configuration

The traditional “static snapshot” approach of installation qualification and configuration management has proven difficult to apply, and is ineffective when using modern virtual environments and cloud computing. If the build is maintained in an automatically controlled environment with regular checks to verify the system remains within its specified setup condition, then the need to independently confirm the installation is reduced to a review by exception. Most automated build installers provide reporting tools for any configuration and setup anomalies and failures with a dashboard to help operational staff quickly respond and resolve any issues. Such configuration management of the build script ensures that the continuous verification is synchronized to compare the current system against the correct setup conditions. Critical thinking can be applied to identify effective modern processes such as this, since technology has superseded the need for traditional aspects of qualification. Section 4.3 discusses IT service management in more detail.

20.3.8 Operation and Maintenance

20.3.8.1 System Evolution

Computerized systems will be subject to ongoing changes and evolution during their operational life. Critical thinking allows change to be embraced in support of improved operation rather than avoided as long as possible because of the perceived validation and documentation burden associated with change. The regulatory expectation is that application updates that offer improvements to patient safety, product quality, or data integrity should be applied when available and, for instance, that GxP systems do not run on obsolete or unsupported operating systems. See *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20].

Influences such as the increased adoption of Agile software development methods, SaaS application offerings and the associated software-release cadence, and the need to update systems more frequently as part of cybersecurity efforts, mean there will increasingly be more changes during the operational phase of the life cycle. Indeed, the increasing number and frequency of cyberattacks have forced the industry to routinely adopt patches and hotfixes immediately after release to address potential security vulnerabilities.

Automated tools can facilitate efficient change management and regression testing, verifying critical functionality and ensuring that regulated data is not adversely impacted after an application upgrade. Appropriate performance metrics can confirm solutions continue to be fit for intended use while safeguarding patient safety, product quality, and data integrity throughout the changes. Appendix M11 discusses the advantages of moving from a fixed point-in-time qualification approach for infrastructure to a continuous control and monitoring approach, which has the flexibility to accommodate and manage more frequent changes.

When a defect is discovered during operational use and the incident management process triggered, it is important to apply critical thinking during the root-cause analysis to assess whether the original risk-based testing approach is still valid. An excessive response may be to initiate a “test all” approach to find other escaped defects. Instead, test metrics should be used to confirm that the actual escape rate of defects is in line with the assumptions made when determining the type of testing to be carried out for functions and features with different risk priorities. An unacceptable number of escaped defects in high-risk functions requires a re-evaluation of the functional risks and the corresponding test strategies.

20.3.8.2 Ways of Working

The operational compliance of a computerized system is also highly dependent on the working practices of its users with respect to safeguarding patient safety, product quality, and data integrity. These practices should be defined in a framework of easily understood and intuitive practical instructions (e.g., SOPs, video how-to guides, built-in online help) for routine use and data review, system support, incident management, system administration, etc. Changes to system configuration are typically captured in system logs that may be reviewed, but management of system configuration changes should be primarily ensured by effective ongoing security, access control, and change and configuration management processes rather than a retrospective review of logs.

The holistic perspective prompted by critical thinking should ensure the work environment uses effective ways of working. Critical thinking should be applied to ensure supervisory measures are proportionate to risk, such as, where is it appropriate to have a contemporaneous second-person verification versus a later check of completed data activity. Special consideration is needed for hybrid situations where records are compiled from manual and automatic processes to ensure controls are complete and complementary without being overly bureaucratic.

Cultural dynamics may challenge the acceptability of openly reporting problems. Codes of conduct should specifically state behavioral expectations to ensure the reliability and completeness of data, explaining the benefits this brings to patients and the personal consequences for employees who breach controls. It may be beneficial to institute confidential reporting lines for whistleblowers¹⁵ where these do not already exist.

¹⁵ A whistleblower is “one who reveals something covert or who informs against another; especially: an employee who brings wrongdoing by an employer or by other employees to the attention of a government or law enforcement agency.” [58]

20.3.8.3 Training

When introducing a new or updated computerized system, critical thinking can be used to help identify and target appropriate levels of training for different audiences. Examples of training that should be considered for relevant personnel include:

- Human factors for design and use of computerized systems – for the implementation and validation SMEs
- Reporting data integrity issues – for all system users
- Transparent investigation of incidents and issues to understand true root causes beyond the initial reasons so that robust computerized systems are created and maintained – for the users, support personnel, and quality SMEs

The content of training needs to be scaled to the detail and reinforcement of principles needed. Training is not necessarily effective if it just repeats earlier content already delivered to the same audience. Neither is training necessarily effective if it relies on attendees documenting they have read and understood the material as they may not be qualified to make such a judgment. Trainers should consider fresh and innovative approaches to make training interesting and informative so that its effectiveness is maintained and enhances organizational capability.

Operational compliance is discussed further within Chapter 4 and *ISPE GAMP Good Practice Guide: A Risk-Based Approach to Operation of GxP Computerized Systems* [34].

20.3.9 Periodic Review

During periodic review the system is assessed for the cumulative impact of changes, defects, or regulatory updates. The frequency of periodic review should be based on the GxP impact of the system, with high-risk systems reviewed more frequently. Critical thinking should be used to determine whether the review frequency should be increased or decreased based on the outcome of the previous review and/or performance trends. For example, if issues and failures were found during the last periodic review, the review period can be shortened to make sure the CAPAs were completed and have resolved those issues.

Effective system and process performance monitoring and metrics allows decreased periodic review frequency, extent, and formality. Critical thinking should also be used in conjunction with system performance metrics and trending data to identify areas for improvement within the system's use, maintenance, and ongoing operation. This includes considering whether incremental changes to the system's functionality have extended the system beyond its intended use.

Critical thinking can also be applied to ensure that the appropriate rigor is applied to change management to minimize the frequency of periodic reviews and ensure the validated state is maintained, thus eliminating the need for revalidation. Periodic review is discussed in more detail in the *ISPE GAMP Good Practice Guide: A Risk-Based Approach to Operation of GxP Computerized Systems* [34].

20.3.10 Retirement

Retirement is the last of the system life cycle phases and consists of withdrawal, decommissioning, and disposal. Data may be retained for a period in its original system for reading, migrated to a new replacement system, or migrated to another system for archive. Critical thinking and risk management are needed to effectively evaluate what data needs to be retained, for what period of time, and how, such as:

- Identifying what data needs to be archived and how to transfer that data if the archival solution is not the original system. Data originally created in this system that has completed its retention period may be deleted from the system so long as there are no legal holds associated with it.

- Balancing the risk of data migration against the complexity of maintaining a legacy copy of the system.
- Retaining legacy systems as a long-term solution to record readability because systems and software become obsolete over time.
- Understanding the diminishing value of data as it moves through the retention period and how that impacts the controls required. Data recently created in the system may need to stay dynamic for a period. Data that is inactive at this stage (i.e., only likely to be needed in an investigation or audit) may be retained in a static, easily readable, and portable format rather than keep it available on the system in which it was created.

Planning the means to meaningfully view archived data if the original system that created the data is decommissioned since data formats used between the original and archive system may differ.

Data migration is discussed in Appendix D7 and *ISPE GAMP Good Practice Guide: A Risk-Based Approach to Operation of GxP Computerized Systems* [34]. Automated migration tools provided as standard items by system suppliers are often either assessed in detail or are used without considering the different types of problems that might occur within the end-user's system. The regulated company needs to consider the maturity of the supplier, the robustness of the tool, and how well the tool meets their migration needs. Assurance of the tool should be focused on actual use cases to test potential problems and confirm that it migrates correctly.

Automated test tools may be also leveraged to test the migrated data. Critical thinking is needed for planning how to cleanse and verify that the data is ready for migration, how to assess the quality of the migrated data, how much data needs to be checked, and how any errors created during the migration process may be detected.

Detailed guidance on system retirement is available in the *ISPE GAMP Good Practice Guide: A Risk-Based Approach to Operation of GxP Computerized Systems* [34]. Further advice on handling mergers, acquisitions, and divestments that can change data ownership and disrupt the data life cycle can be found in Section 3.7 of the *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36].

20.3.11 Inspection Readiness

Critical thinking should be applied to support ongoing regulated company compliance and inspection readiness. Policies and procedures within the QMS can position where critical thinking should be applied, and plans, specifications, and flexible templates can prompt specific considerations. The practical application of critical thinking should be apparent in the rigor of the system activities and ongoing maintenance of the validated state. For example, rationales should be available to explain how high-risk areas were assessed and identified. These rationales in turn justify the level of effort, formality and documentation. A regulator will view the computerized system with a fresh pair of eyes, and obvious viewpoints and decisions may not be self-evident to someone not involved in the initial activity.

During an inspection, a regulated company may need to provide evidence of the assessment and qualification of their supplier (e.g., to satisfy EU GMP Chapter 7 [59] and EU GMP Annex 11 (3.1) [32]). Where supplier activities and information are leveraged as part of the regulated company's validation for intended use, information on supplier assessment and management processes should be available for review.

Holding duplicate copies of supplier information that rightly belong in the supplier QMS is unnecessary, and brings the risk of inconsistency and increased complexity for no quality benefit.

The regulated company may consider contractual agreements allowing access to critical supplier information under specified extreme circumstances. It is prudent for regulated companies to consider checking such arrangements to make sure they are practical and work before they are needed. For clinical trial specific systems, the EMA Q&A: Good clinical practice (GCP) [60] states that the sponsor or Clinical Research Organization (CRO) must have "*detailed knowledge about the qualification documentation and can navigate in it and explain the activities as if they had performed the activities themselves.*" In such cases there should be a clear understanding of the relationship between the supplier's validation environment and the sponsor's production environment, and established and effective configuration management procedures.

20.3.12 Organizational Capability

The successful application of the preceding guidance on critical thinking is highly dependent on the ability of the organization to build and maintain the necessary supportive mindset and culture within its workforce. Open and constructive discussion between stakeholders is vital to evaluate and challenge the situation, the information, assumptions, and organizational precedents in order to make better decisions based on agreed rationales.

The capability of an organization needs to be developed to become more efficient and effective in its application of critical thinking. This capability build can be measured and consequently steered through a series of levels of increasing maturity, for example, applying the maturity levels discussed in and *ISPE GAMP Guide: Records and Data Integrity* [35] to the organizational critical-thinking capability [20] results in Table 20.1.

Table 20.1: Critical-Thinking Capability Maturity

Maturity Level	Capability
1	No application of critical thinking evident in decision-making either by practitioners or by management
2	Some awareness of critical thinking within the organization but highly variable across individuals and departments
3	Critical thinking is described in policies and procedures but is inconsistently applied
4	Critical thinking principles are fully incorporated and routinely applied in working practices
5	Critical thinking is respected as a core competency with organizational capability continually improved

Much of the benefit offered by critical thinking will come from the experience and knowledge of the SMEs planning, specifying, implementing, testing, managing, and maintaining computerized systems. The SMEs can identify how best to realize the opportunities offered by critical thinking within the regulated company. The key objective is that computerized systems are fit for their intended use and efficiently maintained in a state of control.

21 Appendix D1 – Specifying Requirements

21.1 Introduction

This appendix provides guidance for the production of requirements for a computerized system or system component.

Depending on the nature of the computerized system to be specified, the selected project management and software development approaches (e.g., linear or iterative), and the involved parties, the approach to specifying requirements might be achieved using multiple methods, either all at once or incrementally over time. Regardless of the approach used, it should be defined during planning.

The extent and detail of requirements should be commensurate with risk, complexity, and novelty, and should be sufficient to support subsequent risk analysis, traceability, system development, and verification as required. Existing information such as process maps, business-process risk assessments, and data flow diagrams, should be used when determining the extent and detail of requirements.

For example, for a commercially available and low-risk system it may be appropriate to include the requirements in purchasing documentation, while a complex and custom application may require several levels of requirements specification, for example, a Requirements Specification (RS) and Functional Specification (FS). The requirements should define the intended use in the operating environment including limits of operation. The use of Agile, which includes an incremental approach to define/discover requirements, is discussed in more detail in Appendix D8.

Requirements should accurately reflect the business process and data workflows to establish a computerized system that meets the intended use. This is discussed in detail in the *ISPE GAMP® RDI Good Practice Guide: Data Integrity by Design*, Chapters 4 and 5 [36]. The approach should be top-down and based on product and process understanding including CQA and relevant regulatory requirements. This understanding facilitates the adoption of a Quality by Design (QbD) philosophy. It should be noted that cloud computing impacts hardware and logical requirements (e.g., capacity and availability) significantly.

The level of detail provided in this appendix assumes a complex configurable or custom system (Category 4 or 5) with considerable risks for patient safety, product quality, and/or data integrity. For systems of lower risk and/or complexity, e.g., Category 3 systems, critical thinking needs to be applied to determine the appropriate and simpler approach for gathering and documenting requirements. This appendix covers a wide range of requirement aspects that could be of potential interest for specific systems to support that critical thinking. It is by no means a checklist with mandatory requirements that every system must address.

RS is the responsibility of the regulated company but may be collated by a third party or supplier considering the intended use within the regulated company. The intended use can be defined through business process understanding including the process, the data flow, and knowledge of the regulatory requirements. For the development of regulated software or services, third parties or suppliers should gather and document appropriate requirements based on user and industry input.

21.1.1 Changes from GAMP 5 First Edition

The previous version contained separate appendices for User Requirement Specification (URS) and FS, which have been updated and combined into this single appendix. Revisions also to take into account Agile development methods and the increased use of tools and automation in the capture and definition of requirements. The former Appendix D2 has been removed. Functional design specifications, e.g., for custom systems, can be prepared if needed and are covered in Appendix D3.

21.2 Scope

This appendix provides general guidance on the development of requirements for a wide range of computerized systems. It also provides specific guidance on the typical contents of an RS where this is in the form of a document, and these principles also apply where requirements are in the form of records/information within tool rather than a document. See Section 4.2.6 of the Main Body for typical examples of the level of specification required for standard products, configured products, and custom applications.

21.3 Guidance

21.3.1 General Guidelines

Requirements define, clearly and precisely, what the regulated company requires the system to do and what functions and facilities are to be provided to meet the intended use. It should be driven by the business process needs.

Requirements may be developed independently of a specific solution prior to selection. There may be a limited number of suppliers or a preferred supplier for some systems, in which case requirements may be based on the available solution, but these need to be reviewed and tailored for the specific intended use. This is particularly relevant to many Category 3 systems. Such a decision should be based on risk, complexity, and novelty. In these cases requirements related to patient safety, product quality, and data integrity still should be included along with requirements that describe the intended use and business process needs.

Requirements may not initially be fully defined, e.g., for Category 5 systems or custom functionalities, and requirements will be developed during subsequent phases of the project especially when applying an iterative, Agile management approach. There the backlog items are considered as the RS and a predefined classification of backlog items as, e.g., epics, features, user stories etc., should be used to structure the requirements.

RS may be maintained as a document or as records in an appropriate system. The selected approach should meet the expectations outlined below.

The content of a RS typically includes, but is not limited to, as appropriate:

- Operational requirements
- Functional requirements
- Data requirements
- Technical requirements
- Interface requirements
- Environment requirements
- Performance requirements
- Availability requirements
- Security requirements
- Maintenance requirements
- Regulatory requirements

- Migration of any electronic data
- Constraints to be observed
- Life cycle requirements

These are discussed further in Section 21.3.3.

Development of requirements may be assisted through iterative prototyping (see Section 7.5.1 of the Main Body).

21.3.1.1 *Quality-Critical Requirements*

Requirements should address applicable GxP regulations and should highlight those aspects that are critical to patient safety, product quality, and data integrity. For example, the RS should not include/state general and unverifiable requirements such as “Part 11 compliant” or “GMP-compliant;” it should define what functionality the users need in the system to manage risk to patient safety, product quality, and data integrity.

Identification of quality-critical requirements enables companies to focus on those aspects of systems that are critical to patient safety, product quality, and data integrity during subsequent risk analysis, specification, configuration/design, and verification activities.

21.3.1.2 *Requirements Good Practice*

Increasing levels of detail can be expressed and managed by using a recursive hierarchy within the specification by deriving lower-level requirements from higher-level requirements. More information on this can be found in Section 2.3.3 of the *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20].

Requirements should be:

- Sufficient and appropriate:
 - Specific
 - Measurable
 - Achievable
 - Realistic
 - Testable (Verifiable)

Note: Other principles like INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable) [20] may be used instead of SMART. (See also Appendix D8 Section 28.3.7 and Section 3.4.2 of the *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20].)

- Specific enough for testing and checking:
 - Unambiguous
 - Clear
 - Precise
 - Self-contained

- Consider design constraints (i.e., the externally defined limitations that a system must meet, e.g., hardware and/or software platform, speed, power, test, environmental and operating conditions)
- Define internal and external interfaces
- Able to support full traceability through configuration/design and testing – see Appendix M5
- Providing a basis for formal testing, and be used during supplier selection
- Prioritized with emphasis on identifying the mandatory requirements. For example, a three-level prioritization scheme could be used:
 - Mandatory (high)
 - Beneficial (medium)
 - Nice to have (low)
- Use consistent naming conventions, uniquely identified and version controlled, and a change history maintained
- Linked to the associated business-process step(s) where appropriate
- Enable clear communication and management of the critical requirements throughout the life cycle rather than being seen just as a paper exercise
- Provide the supplier with the definitive statement of mandatory and other desired requirements where appropriate

It may be useful to consider categorizing requirements in some other way, e.g., by quality, safety, or business. For commercially available and low-risk systems prioritization may not be necessary. The use of diagrams and graphics where appropriate is recommended

21.3.2 Ownership

Ownership of business requirements lies with the business process owner of the regulated company. Without user ownership the business operational needs and any associated issues can never be fully understood and captured. Defined requirements form the basis for acceptance of the system by users.

Software or service providers may maintain a separate set of requirements that are more technical in nature and more focused on the required functionality. In Agile approaches these requirements are often owned by the product owner.

SMEs, including those from third parties, may help both the user and technical communities analyze and understand the operational needs and develop and capture appropriate requirements.

21.3.3 Contents of the Requirement Specification

Listed in the following sections are topics that may be included in the RS.

Where there is a limited number of suppliers, or a preferred supplier for a system, requirements may be based on the available solution but they need to be reviewed and tailored for the specific intended use. This is particularly relevant to many Category 3 systems where the requirements may be included in purchasing documentation. Such a decision should be based on risk, complexity, and novelty. This section may provide useful guidance for such systems.

The guidance provided in this section is not intended to be exhaustive. If required, information already available elsewhere should be referenced and not duplicated.

12.3.3.1 Introduction

If the RS is created and maintained in a system such as a requirements management system, the introduction information may be included in a planning document.

The introduction should provide information on:

- Who produced the document, under what authority, and for what purpose
- The contractual status of the document (if applicable), for example:
 - Custom development
 - Outsourcing
- Relationship to other documents (e.g., business process map, Request for Proposal (RFP))
- High-level description including a breakdown of the primary components (e.g., sub-systems, segments)
- Assumptions/restrictions: these should state any design or implementation assumptions or constraints (e.g., use of standard products, operating system, hardware)

A data flow diagram should be referenced, created, or updated as necessary. The definition of a primary record should be added as necessary if data resides in multiple systems.

21.3.3.2 Overview

An overview of the system should be provided, explaining why it is required, the essential system functions, interfaces to other internal or external systems, and what is required of the system. If the RS is created and maintained in a system, the overview information may be included in a planning document.

The following should be considered:

- Background: describes the overall goal of the system in context of the present and desired state
- Impact upon patient safety, product quality, and data integrity
- Scope:
 - What portion of the long-term vision the current system will address
 - System limits and boundaries: what business process or portion of a business process is being automated
 - Key objectives and benefits
 - Applicable GxP requirements
 - Other applicable regulations

21.3.3.3 Operational Requirements

Operational requirements include:

- Functions

- Data
- Technical requirements
- Interfaces
- Nonfunctional attributes
- Environment

Process descriptions or flowcharts may be included as appropriate.

Special consideration should be given to critical GxP requirements. It should be possible to trace GxP requirements to applicable regulations.

All requirements should be verifiable. It should be noted that some requirements may be difficult to define and verify because they are subjective and therefore may be subject to different interpretation. The measurement or acceptance criteria for these requirements should be specifically defined as part of the approved requirement.

Functions

Depending on the risks associated with the system, high-level descriptions/requirements may be broken down to the level of the individual functions, for example, by using a recursive hierarchy. Those functional requirements that would enable a system to perform the business process being automated should be documented. The following should be specified as appropriate:

- Objective of the function or facility, and the details of its use, including interfaces with other parts of the system. Inputs, outputs, algorithms, and impact on other functions, other systems, and/or the operating environment should be highlighted.
- Calculations, including all critical algorithms and data entry/edit checks that support CQAs, for example, primary endpoints in clinical trials. Algorithms should be scientifically derived and referenced to their scientific sources where possible.
- Functions that inherently need to be configurable and any limits to the configuration
- Performance: response, sizing, centralized or distributed processing, and throughput. These should be quantitative and unambiguous.
- System safety including action in case of selected software or hardware failures, self-checking, input-value checking, redundancy, access restrictions, time-outs, and data recovery
- Security including access control
- Audit trails
- Use of electronic signatures
- Output (e.g., reports, files)
- Error conditions, unambiguous error messages, failure actions, logfiles, and diagnostics

Data

Data-handling requirements should be documented based on the impact to patient safety, product quality, and data integrity. The following should be addressed as appropriate:

- Definition of records and data: when needed (for example, in custom software) these should be defined in a hierarchical manner with complex objects built up from simpler objects (e.g., files of records; complex types defined in terms of simple types)
- If data is stored in multiple places, the primary record should be specified
- Definition of critical data, potentially including identification of characteristics, formatting, critical parameters, valid data ranges for all inputs and outputs, limits and accuracy, character sets, etc., depending on their relevance for CQAs of the system
- Required fields
- Data relationships
- Data-validation checks
- Data migration
- Data input and subsequent editing
- Backup and recovery
- Data capacity, retention time, and data archiving
- Readability of data
- Data security and integrity including access (for further information on this see the *ISPE GAMP Guide: Records and Data Integrity* [35])

For further details on the data life cycle see the *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36].

Technical Requirements

System technical requirements should be defined. The following should be addressed as appropriate:

- Changes in system operation (e.g., start-up, shutdown, test, failover)
- Disaster recovery
- Performance and timing requirements
- Action required in case of failure
- Minimum capacity requirements
- Minimum access-speed requirements
- Minimum hardware requirements

- Portability
- Availability
- Configurability

Interfaces

System interfaces should be described defining how the systems or sub-systems interact, what they each provide, and what they require. For GxP regulated systems, the security of the interfaces is important. The design and validation of interfaces is covered in *ISPE GAMP RDI Good Practice Guide: Data Integrity – Key Concepts*, Section 4.4 [57]. For complex or custom interfaces, it may be appropriate to create a dedicated interface specification or to manage them as stand-alone projects. The following should be addressed as appropriate:

- Interface with users: this should be defined in terms of roles, e.g., operator or administrator. Topics to consider include error handling and reporting, and security.
- Interface with equipment such as sensors and plant equipment
- Interface with other systems: this should cover the nature and timing of the interaction, and the methods and rules governing the interaction. If there are middleware constraints, this should be noted.

Topics to consider for any interfaces are listed below:

- Data transmitted and received
- Data type, format, ranges, and meaning of values
- Timing
- Rates of data transfer
- Communications protocol: initiation and order of execution
- Any data sharing, creation, duplication, use, storage, or destruction
- Mechanisms for initiation and interruption
- Communication through parameters, common data areas, or messages
- Direct access to internal data
- Error handling, recovery, and reporting
- Access and security

Nonfunctional Attributes

The way in which the system will meet nonfunctional requirements should be described. The following should be addressed as appropriate:

- Availability (e.g., disaster recovery including recovery time objective/recovery point objective, backup and restore, access from different locations/networks)

- Availability of support
- Maintainability (e.g., expansion and enhancement possibilities, cloud storage elasticity, likely changes in environment, lifetime)

Environment

The environment in which the system is to work should be defined. The following should be addressed as appropriate:

- Layout: the physical layout of the plant or other work place may have an impact on the system, such as long-distance links or space limitations
- Physical conditions (e.g., temperature, humidity, external interference, shielding against radio frequency, electromagnetic and/or UV interference, dirt, dust, sterile, or high-vibration environment)
- Physical security
- Power requirements (e.g., voltage, amperage, filtering, loading, earthing protection, Uninterruptible Power Supply (UPS))
- Special physical or logical requirements, e.g., encryption or physical hardening

21.3.3.4 Constraints

The constraints on the specification and operation of the system should be identified. The following should be addressed as appropriate:

- Compatibility, taking into account:
 - Any existing systems or hardware
 - Any regulated company strategy or policy
- Availability
- Reliability requirements
- Maximum allowable periods for maintenance or other downtime
- Statutory obligations
- Working methods
- User skill levels
- Expansion capability
- Likely enhancements
- Expected lifetime
- Long-term support

21.3.3.5 Life Cycle Requirements

Any specific requirements that may impact the supplier's development life cycle and any subsequent verification activities should be identified following a risk-based approach. If this information is already provided elsewhere it should not be repeated.

The following should be addressed as appropriate:

- Development requirements (e.g., minimum standards to be met by supplier's methodology)
- Procedures for project management and quality assurance
- Mandatory design methods
- Special testing requirements
- Test data
- Load testing
- Required simulations
- Acceptance testing
- How deliverable items are to be identified
- In what form deliverables are to be supplied (e.g., format and media)
- Information the supplier must make available on request (e.g., in case of a regulatory inspection)
- Data to be prepared or converted
- Tools
- Training courses
- Archiving facilities
- Support and maintenance required after acceptance

21.3.3.6 Glossary

Definitions of any terms that may be unfamiliar to the readership of the document should be provided.

21.3.3.7 Acceptance and Change

The approvers should be defined. This may need to be defined separately for Agile iterations and for releases of a system. At a minimum this needs to include the appropriate process owners for system releases. Other signatories should include the system owner and quality unit. See also Section 28.3.5.

Once approved, additions, changes, or deletions to the RS should be handled via change management and should be reapproved. Agile approaches must have appropriate approvals embedded in their processes.

21.3.4 Out of Scope Topics

This section is aimed at systems with multiple levels of specification and verification, and may not be applicable to commercially available, low risk, Category 3 systems.

The information listed below should not be included in the RS:

- System configuration/design details
- Implementation details
- Project deadlines
- Cost
- Project organizational details

System configuration/design details are a part of the solution to how the requirements will be met, which will be defined in subsequent specifications. Implementation details also are dependent on the solution and may not be known at this point.

While a regulated company may have deadlines and budget for a project, the final timeline will be driven by the solution selected, as will cost. When submitting the RS to a supplier in the context of an RFP, the desired timeline and available funds can be included as a requirement, but they are not a part of the definition of what a system is required to do.

21.4 Requirements Capture

For Category 4 and 5 systems, this is often the most difficult and time-consuming aspect of producing an RS. Developing the RS is one of the most important tasks the regulated company will undertake in the project.

A suitably experienced individual should be identified and made responsible for managing the requirements-capture process. For iterative development (for example Agile), the product owner has this responsibility, typically working between the Scrum team and the process/business owner to discover and prioritize requirements, and then agree where they fit in terms of defining a Minimum Viable Product (MVP), which will form the basis of an initial working release of the system.

21.4.1 Requirements-Capture Process

There are a variety of ways that business needs can be captured and refined.

21.4.1.1 Discussions and Interviews

Discussions and interviews should be planned and should include participants such as:

- Process and system owners
- Business process participants and users
- SMEs

Asking the participants general questions is not effective. The following specific aspects should be considered:

- Participants should be asked open-ended questions so that their requirements can be investigated
- The participant's actual involvement in the process that is to be automated should be determined. It should be noted that a different level of involvement will have a different focus.
- Participants should be asked to identify the weaknesses of the existing process. It should be determined whether a new system needs to fulfill further requirements in order to resolve those weaknesses.
- The individual who provided specific items of information should be documented to identify who should be asked for clarification, as required
- How a system should respond to errors should be determined
- Terms familiar to the participants should be used; they should not be expected or required to learn technical terms. Facilitation of the gathering process by an external SME may be helpful. This may be supplemented by applicable system demonstrations.
- A project glossary should be developed, to ensure a common understanding among all members of the project team
- Requirement gathering teams should avoid proposing solutions.
 - Proposing supplier/specific solutions stifles consideration of what is actually needed
- Examining documentation such as issue logs from an existing system being replaced may provide valuable information.

21.4.1.2 Observation

The business function should be observed during this activity:

- The current business process should be understood, noting that "what is done today may not be the best solution for tomorrow"
- Caution should be exercised regarding designing a new system to duplicate the current business process; the current process may not be the best way to meet all requirements
- All aspects of the current business process and its interaction with other aspects of the company's overall business process should be examined
- The parts of the current business process to be automated (project scope) should be considered

Critical thinking should be applied to optimize the business process and the requirements supporting that process.

21.4.1.3 Workflow Analysis

This activity involves the examination of workflows and the development of use cases:

- A use case describes the interactions between a system and hardware/software/equipment and people outside the system

- Use cases are tools that aid in:
 - Gathering requirements
 - Developing Standard Operating Procedures (SOPs)
 - Developing training materials
 - Writing test scripts
 - Designing a system

21.4.1.4 Workshops

Workshops usually involve multifunctional meetings and may involve system SMEs:

- Participants should be focused on the task under discussion
- While ensuring that all affected user groups are represented, the size of the workshop or team groups should be manageable
- All participants should understand their role in the workshop
- Participants should focus on their area of expertise:
 - Speculation about what someone else wants should be avoided; however, awareness that a fresh viewpoint may offer a new perspective on a problem is important
 - Tangential discussions that may disrupt the workshop should be minimized
- Those persons representing an area should be empowered to make decisions for that area
- Secondary workshops or teams that focus on a specific area's needs or a specific category of requirement(s) may be appropriate for large projects

21.4.2 Requirements Planning Pitfalls

Certain aspects of the requirements-capture process require particular attention, including:

- A common understanding of the requirements among team members should be established
- All required levels of the business should be involved during requirements capture
- Ambiguous requirements should be avoided and, where possible, requirements should be measurable
- Requirements should be classified to ensure that appropriate focus is given to critical requirements
- Functionality that will not be used should be avoided
- Scenarios that unduly hold up requirements capture and may lead to other key requirements being missed or misunderstood should be avoided
- The original scope should be maintained; extending the scope should be possible only through a formal change-control process

- An effective and efficient change-management process should be implemented, incorporating an impact assessment of changes based on risk and formal version control
- Multiple requirements within a single-requirement statement should be avoided

The need for increasing levels of requirements detail and functionally detail is addressed in Appendix M4.

22 Appendix D2 (Retired)

Appendix D2 Functional Specifications has been retired and the content incorporated into Appendix D1 Specifying Requirements.

23 Appendix D3 – Configuration and Design

23.1 Introduction

This appendix provides guidance for defining the required configuration of system components and for system design.

Based upon the type of system (e.g., configurable or custom, on-premise or SaaS), Configuration and Design Specifications (CS and DS) provide a detailed, technical expansion of the RS (see Appendix D1). They explain how the system will do what is defined in the RS.

23.1.1 *Changes from GAMP 5 First Edition*

This appendix has been updated in line with changes made to Appendix D1 Specifying Requirements, and also to align with information in new appendices Appendix D8 Agile and Appendix D9 Software Tools.

23.2 Scope

This appendix applies to the production of all CS/DS.

Separate documents may not always be needed to adequately define configuration and design aspects. A hierarchy of specifications may be required for larger more complex systems, while specifications may be combined for smaller, simpler systems or system classed as low risk. Furthermore, a system may be used to record and track configuration and design information including required approvals or acceptance.

23.3 Guidance

23.3.1 *Overview of Configuration and Design*

23.3.1.1 *Configuration*

Configuration specifications should be provided for configured products and cover the appropriate configuration of the software products that comprise the system to meet specified requirements. This includes the definition of all settings and parameters supporting the intended business process.

It may be possible to maintain configuration information electronically in systems with robust configuration management, e.g., audit trails. Such an approach should be clearly documented.

23.3.1.2 *Design*

Custom applications require design of hardware and software, and also may require CS.

Hardware design defines the hardware components of a system, e.g., system or component architecture, or interfaces.

Software design occurs at two levels. At the higher level it defines the software modules (sub-systems) that will form the complete software system, the interfaces between these modules, and the interfaces to other external systems including the data flow. This high-level design should also be traceable to the RS, where appropriate. At the lower level the design describes the operation of the individual software modules. These specifications should be unambiguous, clear, and precise.

The regulated company or cloud service provider should have a unified approach to the specification and verification of infrastructure that supports the system design, and such activity should not be repeated for each system; see the *ISPE GAMP Good Practice Guide: IT Infrastructure Control and Compliance (Second Edition)* [49].

23.3.2 General Guidelines

The use of tables and diagrams to illustrate CS and DS is highly recommended. If such tables or diagrams are produced elsewhere then these should be cross-referenced in the appropriate specification. Standardized tables can help ensure that all relevant parameters and settings have been defined. Diagrams can be helpful in software design to clarify and explain data flow, control logic, data structures, interfaces, and mapping to corresponding business processes. Diagrams in hardware design can aid understanding of architecture and connectivity. The diagrams should aid in the assessment of the system modules and sub-modules during the course of the defect fixes and enhancements.

Configuration and design should cover both hardware and software aspects. Depending on the risk, size, and complexity of the system this may be covered by a single specification or may require a hierarchy of specifications covering software and hardware separately. In the case of tier of specifications each specification should be uniquely referenced and traceable back to its appropriate higher-level specification. A system may be used to record and track configuration and design information including required approvals.

All specifications should be structured in a way that supports traceability through the life cycle from individual requirements to associated testing.

Where a system is hosted by a service provider the CS/DS are typically produced and maintained by the service provider. The regulated company should leverage the service provider effort and where necessary produce a supplementary specification based upon the system configurations required to support the intended business process.

If deliverables from the vendor are being leveraged, the regulated company needs to have oversight through a quality agreement. With evolution in technologies, assessment of supplier documentation should be driven by critical thinking as elaborated in *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20].

See Appendix M5 for further information on design review and traceability.

23.3.3 Information Required

The topics described in this section should be covered by each appropriate specification; not all information is required for all types of system. The level of detail should be based on risk, complexity, and novelty. This may be covered by a single document, by a hierarchy of documents, or within a system.

The guidance provided is intended to be neither prescriptive nor exhaustive.

23.3.3.1 Introduction

The introduction is common to all types of specification and should contain the following information:

- Ownership of the document
- Who produced the document, under what authority, and for what purpose
- The contractual status of the document (if applicable)
- Relationship to other documents (RS, other configuration or design specifications, etc.)

- If the CS/DS is created and maintained in a system, the introduction information may be included in a planning or process document.

23.3.3.2 Overview

The overview should briefly describe the configuration and/or design as defined in the document, including the storage and in some cases, location of the records and the involved systems. Depending on the complexity of the system, this may cover the complete system, hardware, software, functions, data, and/or interfaces. The overview should not contain detailed design information.

The overview may be illustrated using diagrams.

23.3.3.3 Configuration

The required configuration of components to be provided as all or part of the solution should be defined. This includes but is not limited to:

- Required configuration settings or parameters
- Reason for setting, with reference to controlling specification
- Tools or methods that will be used to set the required options
- Dependencies and impacts on other modules or systems
- Infrastructure items such as operating systems and layered software
- Security of settings

For simple systems it may be possible to incorporate this information into the RS.

23.3.3.4 Hardware Design

The Computer System

The overall architecture of the hardware required should be defined. At a high level this may be illustrated by means of an annotated block diagram showing both the functions of the parts and their functional relationships. The individual aspects and level of detail may vary greatly depending on the type of system, e.g., on-premise solution, XaaS, etc. The following should be covered as appropriate:

- Main computer system
- Organizational setup with areas of responsibilities, e.g., for XaaS setups
 - This should describe the primary hardware components of the main computer system(s), e.g., Central Processing Unit (CPU), memory, bus type, clock accuracy
 - This should describe the Virtual Machine (VM) components, e.g., VM software used, processing unit (CPU), memory, bus type, elasticity of setup, other infrastructure as code elements
- Storage including location of data

This should describe all proposed storage devices with their maximum storage capacities, e.g., hard disk, compact disk writer, public/private cloud storage.

- Peripherals including sensors, wearables, mobile devices, and BYOD¹⁶ setups
- Interconnections/networks including protocols/encryption

This should describe all interconnections of the hardware components and any connections to other equipment, devices, and computer systems. The following elements may be included in this description:

- Cable specifications
 - Connector specifications
 - Screening/shielding requirements
 - Drawing schedules
 - Network and other external connections including wireless connections, bandwidth capabilities, and availability e.g., for mobile connections
- Configuration (unless covered in separate CS)

This should cover configuration details such as Dual In-line Package (DIP) switch settings, device addresses, pin assignments, encryption, mobile device lockdown, or preconfiguration as appropriate.

As noted above, it may be possible to maintain configuration information electronically in systems with robust configuration management, e.g., audit trails. Such an approach should be clearly documented.

- Embedded systems (within process equipment)
 - Layout diagrams to detail control panel and interior and exterior arrangements
 - Location diagrams to indicate where sensors and other devices are installed on the equipment
 - Electrical wiring diagrams
 - Piping/Process and Instrumentation Diagram (P&ID) drawings
- Reference to relevant standards

Inputs and Outputs

Input and output formats should, where necessary, be specified. These may include digital and/or analog signals or outputs.

For external equipment the following elements should be considered:

- Accuracy
- Isolation
- Range of current and voltage
- Type and numbers of interface cards

¹⁶ Bring Your Own Device

- Timing
- Encryption and security

Environment

The operating environment for the hardware should be defined. The following topics should be considered:

- Temperature
- Humidity
- External interference
- Physical security
- Shielding against radio frequency, electromagnetic, and/or UV interference
- Hardening against physical hazards such as dust or vibration
- Location

Electrical Supplies

The electrical supply requirements for the configured hardware system should be described. The following elements should be considered:

- Filtering
- Loading
- Grounding protection
- UPS
- Power consumption and/or heat emission to calculate the necessary capacity of the air conditioning or Heating, Ventilation, and Air Conditioning (HVAC) system

23.3.3.5 Software Design

Software should be designed in accordance with recognized design standards where appropriate.

DS covering software design are required for custom applications. This is not normally required for configurable products, where software design is normally reviewed or evaluated as part of the supplier assessment and is often proprietary.

Software Description

The modules that will form the system should be described, briefly stating the purpose of each. A list of all interfaces between modules, and any interfaces to external systems should be given. A system diagram is recommended. For SaaS the delivery model, e.g., single-instance-multitenant should be described.

System Data

System data and the major data objects should be defined. The data should be characterized in a hierarchical manner with complex objects being built up of simpler objects. The objects may include the following:

- Instances and tenants
- Databases and collections of files
- Files
- Records

A description of the data objects will include such things as:

- Data types (integers, floating point numbers, characters, Boolean, string, object, etc.)
- Data format (alphanumeric or numeric, field length, date, etc.)
- Data precision
- Data accuracy

Each file and data structure should be uniquely identified. The use of formal data description methods such as Entity Relationship Models or similar should be considered. In multitenant environments the method of data separation and security should be described.

It is acceptable to have all system data defined separately, such as in a data dictionary. If data is defined separately then this should be clearly explained and documented.

Module Description

For each module the following should be covered:

- Module operation: the description may take the form of pseudo code or a flow chart
- Interfaces to other modules: these may refer to the system diagram, if one is produced
- Error handling and data checking
- Data mapping to each module
- Software module data (see System Data section above)
- Data load and module performance aspects

For each sub-program in the software module, the following should be covered:

- Sub-program operation: the description may take the form of pseudo code
- The steps involved in each process to be performed and the inputs to and outputs from each step

- Parameters: each parameter should be identified as one of the following:
 - Input parameter
 - Output parameter
 - Input and output parameter
 - Algorithms
- Each parameter should be identified as:
 - Pass by value
 - Pass by reference
- Any side-effects of the sub-program
- Language, including version
- Reference to any programming standards
- Description or examples of all display screens (may be in user documentation, e.g., operator manual, and may be referenced)
- Sub-program data (see System Data section above)
- Description or examples of all implemented reports, their meaning and handling, and when they are generated

This level of detail may be provided in separate specifications.

23.3.4 Glossary

Definitions of any terms that may be unfamiliar to the readership of the document should be provided.

24 Appendix D4 – Management, Development, and Review of Software

24.1 Introduction

This appendix provides guidance for the management, development, and review of software. It also includes general guidance on coding practices. Software tools to assist with these activities are available, and in general, automation of these processes will lead to better compliance, efficiency, and reproducibility when compared to manual approaches.

Software development should be performed within a defined life cycle. The criteria for releasing developed and tested software should be understood and defined. Software released into the GxP environment should be based on, and meet, defined requirements.

This appendix is not intended to constrain the choice of development methods and models in any way. Suppliers should select and use the most appropriate methods and models for their use. This Guide is not intended to place any constraints on innovation and development of new concepts and technologies. Any examples used, or technologies assumed, are intended to illustrate general principles, and are not intended to be restrictive.

See Appendix D8 for additional guidance on software development using Agile, incremental software development methods.

24.1.1 Changes from GAMP 5 First Edition

This appendix has been updated in line with changes made to Appendix D1 Specifying Requirements, and also to align with information in new appendices Appendix D8 Agile and Appendix D9 Tools.

24.2 Scope

This appendix may be used to define procedures and standards for software development and review for GxP regulated computerized systems.

While the main focus is on software development using high-level and low-level languages, the principles of this appendix also may be applied to various techniques used for process control system software development/configuration. For further information on this topic see *ISPE GAMP® Good Practice Guide: A Risk-Based Approach to GxP Process Control Systems* [61].

24.3 Guidance

24.3.1 Software Development

The software development method should be defined and regardless of the model used, should also define coding standards, code repository standards, software development tools, and naming conventions to be followed. The configuration and use of build and deployment systems and tools should also be defined and controlled with documented procedures.

Source code should be subject to review with the approach and extent of such reviews based upon the coding language, tools, and risk to patient safety, product quality, and data integrity. The review should take place according to the software development life cycle used; for some life cycles this may be before the testing of the software commences; for others like Agile, this may be during or after testing the individual change within the sprint. See Section 24.3.2 for further details.

Software should be subject to a defined software configuration management process that includes version control of software code, scripts, and any relevant artifacts required by the technologies in use. Such change and version control is especially important during software development, as mistakes in this regard can lead to problems such as reintroduction of previously removed defects. The use of tools can provide high levels of control and assurance. In configuration management, for example, orchestration software provides automated assurance of working and tested code, and provides a defined workflow for moving code between environments; code repository software reduces the chances of configuration management issues, particularly where multiple developers may be working on the same code, and enables traceability from requirements to code level automatically.

Any software development tools used should be assessed for suitability and fitness for purpose. See also Appendix D9.

See Appendix M8 for further details on configuration management.

24.3.1.1 General Design and Documentation Principles

Software design artifacts should be sufficient to initially develop and subsequently support working software. The degree and nature of the design will be based on the life cycle model used, software complexity, code/configuration language, and interfaces. Design artifacts may be in documentation form or as records within software development tools.

With linear life cycle models, typically, the detailed design is developed from preceding functional specifications and user requirements. With iterative software life cycle methodologies such as Agile, the design is developed within the sprint cycles based on the product backlog items (e.g., requirements in the form of epics, user stories, etc.) and design artifacts will vary. For example, acceptance criteria developed at an early stage in the sprint may be used to help drive development.

In general, it is good practice to include an Software Bill of Materials (SBOM) that provides the list of components/modules that make up the overall software and can also be used to identify/track inter-dependencies between the individual items.

Software should ultimately:

- Meet requirements
- Be reliable
- Be robust
- Be maintainable
- Be capable of handling error conditions
- Be well designed
- Be sufficiently commented

24.3.1.2 Software Module Identification

Each software module should include the following attributes:

- Module name
- Reference/traceability to associated design artifacts

- Constituent source file names
- Module version number
- Project name/reference (if applicable)
- Brief description of the module (if module name is descriptive, that may be adequate)
- References/dependencies to other software modules
- Any specific command files required to compile and build the module

24.3.1.3 Software Module Change Traceability

All software module changes should be clearly identified according to the chosen software development life cycle and technology used.

Additions should be identified by the use of commentary that references the reason for the change, e.g., relevant change request number or sprint cycle.

- Deletions should reference the relevant change request (for major changes, e.g., when a significant amount of code is removed) as part of the version history.
- It should be possible to obtain a history of changes made, which should include the following information for each change:
 - Change request number(s) (if applicable)
 - Applicable version number(s)
 - Dates change(s) made
 - Identity of person(s) changing the code
 - Summary of the changes

The use of appropriate tools can greatly assist change management.

Where software development tools are used, the features of these tools, for example configuration/version/change control, should be used to manage these processes in the most efficient and effective way.

24.3.1.4 Maintainability

Software should be written and structured in a way that a competent person (other than the original author) would be able to understand and modify it safely. The software should be created using the coding standards relevant to the technology

24.3.1.5 Open-Source Modules

Where modules are imported from an open-source code repository, updates as and when the open-source code is updated needs to be determine. In addition, occasionally open-source modules may be retired, in which case future vulnerabilities will not be addressed, which increases risk.

24.3.1.6 Removal of Dead Code

Dead code is code that cannot be executed due to the logic of the program, and should be removed. It is usually a symptom of poor maintenance, and may have been left over by accident from development or code changes.

Code that has been included for purposes of testing or for later diagnosis during support work, and which can be configured on or off, is not regarded as dead code. Any such code should be clearly documented.

If the code is configurable or general purpose code that may be used in many different projects, each with different configurations of options, the unused options should not be removed. The software review and the testing processes, however, should demonstrate that the correct options have been selected and that they work, and that the deselected options have been correctly deselected and do not function.

Code that has been properly commented-out during the operational change process is not regarded as dead code and may be an appropriate technique for minor changes during the operational phase. The cleanup of commented-out code prior to formal testing of the initial and subsequent major releases of software should be considered, as an aid to code maintenance.

24.3.1.7 Reliability and Error Recovery

The software should either recover from incorrect data or incorrect operation of any equipment, or fail in a safe and predictable manner. “Defensive” coding techniques should be employed. For example, input values should be checked, and subroutines should check that the values of the received parameters are within range. In the event of a failure, a safe, documented error recovery or a controlled shutdown should be performed, with an informative error message.

24.3.2 Source Code Review

Source code reviews have three objectives:

- To ensure that programming standards, such as those described in Section 24.3.1, are consistently and correctly applied
- To ensure that the code is written in accordance with the design
- To identify and enable removal of defects

The review aims to ensure that the code is fit to release, and that the code can be effectively and efficiently maintained during the period of use of the application.

The review should be performed by at least one independent person with sufficient knowledge and expertise, in conjunction with the author of the code. Further guidance on code inspections and walk-throughs may be found in Myers et al [62].

Automated source code review comparison tools are particularly useful when making small changes to existing code.

24.3.2.1 Static Analysis of Software

“Static analysis involves a set of methods, supported by tools, used to analyze software source code or object code to determine how the software functions and establish criteria to check its correctness.” [63]

These include [63]:

- Control Analysis – analysis of the controls used in calling structure, control flow, and state transitions

- Data Analysis – ensures proper operation is applied to data objects and data structures
- Fault/Failure Analysis – identifies incorrectly specified or built components and incorrect behavior of components
- Interface Analysis – checks the accuracy of interface structure, and the ability to prevent errors during user interaction

Formal Verification methods may also be considered for high-risk areas of code using formal mathematical methods to provide “proof” of algorithms under multiple input conditions.

A decision based upon known risks should determine the extent of software review required for a given project. The criteria for making this assessment, and the outcome, should be documented. Factors that may influence the extent of the software review include the criticality of the application or specific module, the complexity of the design, and the experience of the developers.

Problems found during the review should be recorded and corrective actions defined. Agreed actions should be resolved prior to testing or retesting.

Reviews should be documented as appropriate. Note that this may be as records within tools and not necessarily within a document.

24.3.3 Third-Party Software

While the requirements noted in Sections 24.3.1 and 24.3.2 represent good practice, documentation of these activities may not be readily available for software obtained from outside suppliers or as free and Open-Source Software (OSS). The need for documented evidence of these processes, or alternative controls such as testing, and the degree of effort to be expended, should be risk-based.

24.3.3.1 Purchased Software

In most cases evidence of software control can be obtained through standard mechanisms for supplier assessment, such as supplier audit; see Appendix M2. Conclusions of the assessment should be documented. If inadequacies are noted, the regulated company may require remedial actions by the supplier, possibly enforced via a Service Level Agreements (SLA) or other contractual means, or may decide that further testing is required.

24.3.3.2 Free and Open-Source Software

OSS often plays an important role in the infrastructure of regulated companies. Examples include the Linux operating system, R environment for statistical computing, and many web design, programming, and support tools.

Note that commercial software may contain open-source components incorporated into it. It is becoming more common to request a SBOM when evaluating new commercial software or validating in-house developed systems.

Evaluation of software control for free and OSS is similar in scope and approach to that of purchased software (following established GAMP software categories). However, the nature of OSS is different. Although there is potential for less control, the output (open-source code) is public and can be reviewed. This may not be possible with commercial software. Furthermore, required changes may be implemented more rapidly than they could be by a commercial supplier, due to the availability of several possible coding authors. However, it is up to the organization using the OSS to monitor for, import, and apply those updates. Lack of diligence in this process implies an increased risk if the updates were addressing security vulnerabilities, as those vulnerabilities would be widely published. Often organizations will contract support services through third parties, or become involved in the support community for this purpose.

Some organizations that manage OSS take great care to manage code structure, commenting, version control, and release notes. Others do not. Because open-source libraries often rely on other open-source libraries, they must be actively managed; while documentation for the current implementation may be adequate, it does not mean that the future updates will be. Companies considering using OSS should thoroughly investigate these factors and seek out OSS from well-established and robust communities. Any one-off modification will result in custom code that will have to be managed by the regulated company.

To determine whether there is an adequate level of software maturity and control, a risk assessment should be performed to determine the business and GxP risks of using OSS products, and to identify appropriate controls.

If a decision is taken to use OSS, then such software should be implemented in accordance with the appropriate life cycle activities as for commercial software, and with consideration of the following aspects:

- Knowing the size and sustainability of the open-source community to ensure the community has the staying power to support the future.
- Checking for development standards and good documentation – reading the documentation is an indicator of the quality of the software development cycle.
- Knowing what version is in use – if a local distribution of the software is used, verify that the copy matches the version to be installed. Ensure download is from a reputable source (preferably directly from the repository) and take steps to ensure the code was not altered along the way (often done with a check sum).
- Understanding the governance model for updates and patches. If the company is implementing or connecting to a decentralized and distributed software, such as a public blockchain, make sure that the governance model for that network is understood and have a plan should that network become compromised.
- Keeping up-to-date with patches and updates; vulnerabilities are often exposed in software using outdated versions of open-source libraries, and unless the company is compiling the program itself, this is not always apparent.
- Participating – open source works best with a broad community; the best way to get new features that will make your business better is to ask for them. Having this connectivity becomes part of the IT culture will help to ensure that the company stays in front of any major changes/disruptions.

25 Appendix D5 – Testing of Computerized Systems

25.1 Introduction

This appendix covers the testing of GxP computerized systems. It is not intended to offer a prescriptive approach, that is, it does not require that testing must be done a certain way in a particular sequence. Instead, this appendix identifies the influences on testing (types of systems, different architectures, development models, corporate policies, regulatory requirements, etc.) and the key activities involved, and offers information on the types of testing and testing approaches that can be used selectively and in combination for optimal assurance that system is fit for its intended use.

The application of critical thinking is vital to achieve effective testing that adds value and fulfills objectives such as:

- Identifying defects so they can be corrected or removed before operational use
- Reducing the risk of failures that might affect patient safety, product quality, or data integrity
- Demonstrating that the system as installed and configured meets its requirements
- Providing evidence that the system is fit for its intended use
- Offering a basis for user acceptance
- Meeting a key regulatory requirement
- Ensuring that risk management controls are effective

Testing should be carried out in accordance with an appropriate test strategy based on the risk to patient safety, product quality, and data integrity. Testing may occur both in the vendor's development life cycle and in the regulated company's implementation life cycle.

25.1.1 Changes from GAMP 5 First Edition

This appendix has been revised to emphasize that:

- Critical thinking should be applied when planning testing efforts such that the level of effort is commensurate to the risk acceptable within the organization as defined in its policies, procedures, and plans. The regulated company determines the assurance activities based on their own need to ensure systems are fit for intended use. The key is to determine the regulated company's level of risk acceptance, based on the intended use of the system, factoring in the technical or procedural controls that are currently in place or that will be put in place.
- Testing by any means and in any part of the life cycle and in any environment (development, validation, production, DevOps, etc.) all contributes to finding defects and confirming the system is fit for intended use.
- Testing should not be limited to detailed and prescriptive step-by-step scripted protocols. The use of exploratory testing and other unscripted techniques is encouraged to extend test coverage and improve defect detection. Unscripted testing must be documented and can then be leveraged as part of the overall verification stage. Using automated testing brings benefits to test coverage, repeatability, and speed.

- Modern approaches may rely on records, information, and artifacts in automated tools in place of formal specification and test documentation. Either approach is acceptable, provided the information is complete, accurate, available, and adequately demonstrates that the system is fit for intended use and maintained in a validated state throughout its operational life. For this reason, many of the sections detailing test documentation (test plans/strategies, test specifications/protocols, test scripts, test summaries and reports) from the previous edition have been replaced with Section 25.5, discussing instead testing activities.

25.2 Scope

This appendix applies to testing of all GxP regulated computerized systems. It covers the following key aspects:

- Roles and responsibilities
- Influences on test activities
- Types of testing
- Test environments
- Leveraging previous testing including supplier testing
- Testing applied to different software categories

25.3 Roles and Responsibilities

The regulated company should define roles and responsibilities covering testing. Such definitions typically are included in an appropriate planning document, test strategy, or company procedure. Traditional roles and responsibilities may include:

- Process Owner – responsible for the business process supported by the system and for the approval of the test strategy and overall test coverage
- System Owner – responsible for the system in question and for ensuring system prerequisites are satisfied to allow testing to proceed
- SMEs – essential for input to the test strategy and test case design
- Test manager – plans testing approaches and types of testing
- Test analyst – responsible for developing test cases – see explanation below
- Tester – testers should be as independent as possible, as it is difficult to objectively test one's own work. They should not be authors of the function or feature to be tested as that would typically only confirm the system does what the author expected it to do
- Test reviewer – responsible for reviewing test cases and completed testing; they should not be the same person that executed the specific test case
- Quality oversight, if applicable and necessary

- Supplier – should have already completed substantial testing in their development life cycle. Supplier or third-party representatives may act as test managers, testers, etc., depending on contractual agreements. Final responsibility for compliance remains with the regulated company irrespective of any outsourcing.

Test cases may be created by the developers as part of the coding, by the software testers as part of exploratory testing, by the business users as part of “day in the life” testing, and by a test analyst (usually for manual scripted tests). The quality of the test cases and the functionality they cover directly impact the effectiveness of the testing and therefore, no matter who is creating the test case, they should be an SME in the associated field. Some examples include:

- The software tester should be experienced in software testing techniques
- The business user should be highly skilled and knowledgeable in the business process that the system will support
- The test manager should have an expert understanding of GAMP principles, good software engineering practices, and critical thinking

The focus should be on ensuring that the responsibilities for testing are met no matter what approach is used or who is creating and executing the test cases.

25.4 Influences on Test Activities

Testing is one part of ensuring that a computerized system is fit for its intended use, aimed at verifying that the system selection, implementation, configuration, and use in the operating environment collectively constitute a system capable of supporting the business process. As such, test coverage and approaches are influenced by the:

- Criticality of the business process being supported by the system, including the GxP impact of the system
- Requirements for the system
- Supplier’s development life cycle and quality system
- Regulated company’s internal computerized systems validation policy
- Risk culture within the company
- Use and existence of additional controls elsewhere in the business process
- Level of test automation available.

Table 25.1 outlines the key activities specifically relating to testing and the influences on those activities. The table does not include scaling system controls and configuration based on risk, which are an essential part of system implementation. This table is not intended to replace the detailed descriptions of the life cycle activities found throughout the body and appendices of this Guide. The Key Objectives/Focus of the activities in this table are those related to testing, and not all life cycle activities.

Table 25.1: Activities and Influences around Testing

Activity	Key Objectives/Focus	Influenced By
Defining Requirements	Identifies what the system must do to support the business process and the data integrity technical controls needed. Ultimately the requirements form the specifications against which test cases will be run and provide the basis for determining if the system meets its intended use. Requirements capture is discussed in detail in Appendix D1.	<ul style="list-style-type: none"> • Applicable GxP Regulations • Business Process Map • Data Flow Diagram • Process Risk Assessment
Supplier Assessment	Evaluate if the supplier can provide a system of acceptable quality that will meet the requirements. Extensive well-executed supplier testing can reduce the amount of testing that the regulated company will need to complete. Where such testing is to be leveraged, the regulated company should be able to demonstrate an appropriate supplier assessment process and conclusions upon request during regulatory inspection. Supplier assessment is explained in Appendix M2.	<ul style="list-style-type: none"> • Supplier Qualification Process • Requirements • SLA
Validation Planning	Defines the GxP impact of the system, the life cycle and test strategy, and the decisions made based on the outcome of the supplier assessment. See Appendix M1 for more detail on validation planning.	<ul style="list-style-type: none"> • Applicable GxP Regulations • Computerized Systems Validation Policy • Requirements • Supplier Assessment • GxP Risk Inherent in the Business Process
Functional Risk Assessment	Assesses and prioritizes the risks from the system functionality with the potential to impact patient safety, product quality, and data integrity. For some complex system development projects, more formal and detailed assessments such as Failure Mode Effects Analysis (FMEA) may identify failure modes and drive improved controls. For implementation projects, assessing at the functional or requirement level is typically sufficient. The functional risk assessment is one component of the wider QRM approach covered in Chapter 5 of the Main Body and in Appendix M3.	<ul style="list-style-type: none"> • Requirements • Risk Assessment Procedure or Methodology
Test Planning/Coverage	Defines the depth and rigor of test coverage and documentation needed based on the assigned risk priority of the function, including references to any supplier testing to be leveraged. Agile approaches and the use of end-to-end automated tools may provide much of the test planning and coverage; however, additional manual cases may be needed to verify workflows and overall system fitness for intended use. This activity is discussed in more detail in Section 25.5.1.	<ul style="list-style-type: none"> • Requirements • Validation Planning • Supplier Assessment • Functional Risk Assessment

Table 25.1: Activities and Influences around Testing (continued)

Activity	Key Objectives/Focus	Influenced By
Test Management Processes	Often captured in a test specification or test plan in a manual testing approach, but alternatively recorded in an SOP for more automated and Agile approaches, a test management process needs to define the testing approach and execution instructions. Section 25.5.2 focuses on test management processes.	<ul style="list-style-type: none"> • Computerized Systems Validation Policy • Validation Planning
Creating Test Cases	Test types include, but are not limited to, scripted or unscripted, manual or automated, exploratory, ad-hoc, regression, end-to-end regression, etc. Except for ad-hoc testing, most tests have as a minimum a defined objective, a pass/fail outcome or criteria, and a reference to the requirement or aspect under test. See Section 25.5.3 for more information on test cases.	<ul style="list-style-type: none"> • Test Planning/ Traceability Matrix • Configuration Specification
Executing Tests	Test execution should meet the objectives of the test cases and generate test records/capture test evidence to demonstrate the system’s fitness for intended use or allow defects to be addressed. This is discussed further in Section 25.5.4.	<ul style="list-style-type: none"> • Test Management Processes • Scripted and Unscripted Tests
Reviewing and Reporting Completed Tests	<p>The focus of the review is to confirm if the functionality operated correctly rather than to scrutinize the testing to check for minor errors. Any test failures should be assessed to confirm that any defects have been fixed or have been dispositioned satisfactorily.</p> <p>Overall review should confirm the desired test coverage has been achieved; if any gaps are identified then new tests will need to be created, executed, and reviewed. Section 25.5.5 expands on reviewing and reporting.</p>	<ul style="list-style-type: none"> • Test Management Processes • Test Planning/ Traceability Matrix • Executed Tests
Validation Reporting	<p>Confirms if the validation strategy was implemented as planned and if all the requirements have been met and verified based on risk. Document the impact of any non-conformances or outstanding deviations. Identify where the test records and evidence are maintained, for example:</p> <ul style="list-style-type: none"> • Generated in the system under test • Maintained in the system audit trail • In automated tools • Any completed paper protocols • Supplier test evidence (and how it is assessed or accessed if needed) <p>The full requirements for validation reporting are detailed in Appendix M7.</p>	<ul style="list-style-type: none"> • Applicable GxP Regulations • Validation Planning • Requirements • SLA

25.5 Test Planning/Coverage

Test planning is needed to ensure sufficient test coverage of the system functionality, including adequate verification of GxP functions based on the regulated company’s risk tolerance. Fundamental to the risk-based approach is an acceptance that not all functionalities will be challenged and consequently not all defects will be found. Risk management, when done well, should ensure that the unchallenged functionality and undiscovered defects are low in terms of impact on patient safety, product quality, and data integrity. Test coverage for the regulated company should be limited to the intended use of the system, however, the level of test documentation required will vary depending on the risk associated with the feature/function to patient safety, product quality, and data integrity. Traceability between test cases and requirements or specifications is essential to quantify the test coverage.

For a linear-sequential system development and implementation, as shown in Figure 25.1, the overall test coverage of a system is typically a combination of:

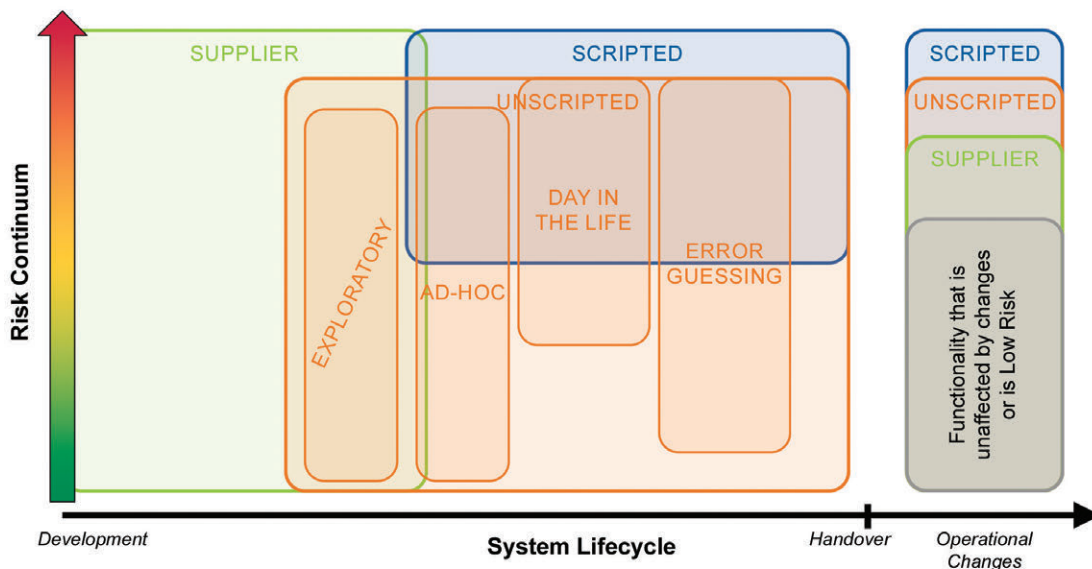
- Supplier testing (confirmed during supplier assessment as robust and suitable for leveraging)
- Unscripted testing (including exploratory, ad-hoc testing, day in the life testing by a business process user, and error guessing) for maximum defect detection across the system functionality
- Scripted testing (manual or automated) to verify the correct functionality of the system controls for risks at the higher end of the risk continuum
- User acceptance testing to formally demonstrate fitness for intended use

There may be some degree of overlap between the testing activities, but this should be minimized as there is no benefit to repeating testing already performed in a controlled environment. The testing in the bulleted list above may have occurred in multiple environments, however, it all contributes to the overall defect detection and assurance of fitness for intended use. This is discussed in more detail in Section 25.7.

A subset of scripted and unscripted testing may be repeated after changes in the operational phase to provide assurance that the system remains fit for intended use.

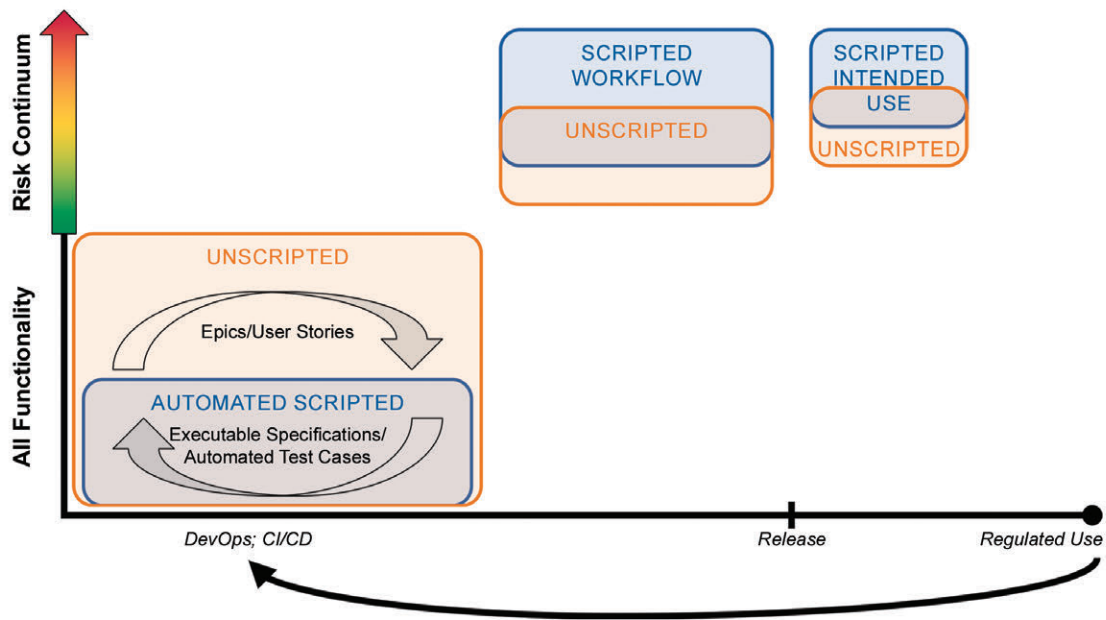
The Figures 25.1 and 25.2 are intended to be illustrative of general possible approaches, and not intended to be definitive or prescriptive.

Figure 25.1: Test Coverage for Linear-Sequential System Developments



Agile development approaches are now widely used for both on-premise software delivery and for SaaS offerings. In Agile, automated test cases can be generated as the requirements are coded. The need for additional testing will be determined by the robustness and coverage of the automated test cases. For example, even when the test cases comprehensively cover the detailed functionality of the system, there may be a need to separately create unscripted and/or scripted test cases to challenge the end-to-end workflows and intended use within the system, as shown in Figure 25.2.

Figure 25.2: Test Coverage for Agile Approaches and SaaS Offerings



Scripted workflow testing may be manual or automated and completed by the provider. Scripted intended use testing focuses on the individual regulated company's intended use of the software as defined by their configuration and internal workflows, and would be scaled based on risk.

The DevOps or CI/Continuous Deployment (CD) output may go straight to regulated use without the need for scripted workflow or intended use testing if the verification needs already have been addressed in that framework, for example, when:

- The regulated company's in-house IT are the developers and therefore can factor in the intended use to their automated scripted testing
- The software solution is entirely standard (i.e., not configurable), and is used as-is by the regulated company so the intended use exactly matches the provider's testing

For subsequent releases, depending on the scope of the sprint, exploratory testing or ad-hoc testing may be combined with regression testing by automated test cases to verify that sprint developments have not impacted functionality from previous sprints.

See Appendix D8 for more information.

25.5.1 Test Management Processes

Early and efficient testing, during development as well as during the verification phase, improves defect detection and reduces the occurrence of defects surviving into the operational phase. A combination of direct evidence (e.g., manual or automatic recording of results (Pass/Fail)) and indirect evidence (e.g., reports, notifications that can only exist if the proving step is completed) can be used to establish that requirements have been fulfilled in support of the intended use.

“Errors that impact product quality should trigger more detailed investigations compared to a catch and correct in the moment approach for less significant typographical errors.” [20]

This is consistent with how both *ISPE GAMP* Guidance [64] and the FDA CDRH Case for Quality program [9] promote taking a risk-based approach to testing [36].

“Test strategies should ensure sufficient system testing to detect defects, and that test management processes are robust enough to maintain control of the system during the testing activities.” [20]

The critical thinking rationale behind the test strategy must be documented.

A test management process may include:

- Any special prerequisites to testing
- Identifying what combination of scripted or unscripted, manual or automated testing will be used
- The personnel required for each test or group of tests
- The expectations regarding test evidence (including the recording and reporting of test pass/fail results)
- How the environment in which the testing is to occur will be identified and recorded
- How the progress and outcome of testing will be monitored
- How test cases and completed tests will be reviewed and approved

25.5.2 Test Cases

The level of detail for test cases can vary depending on the test strategy and the risk posed by the function or feature being tested. The *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36] and the *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20] introduced the concept of scripted and unscripted testing, as summarized below in Table 25.2.

Table 25.2: Acceptable Assurance Approaches and Records

Adapted from ISPE GAMP RDI Good Practice Guide: Data Integrity by Design [36].

Assurance Approach	Test Plan	Test Results	Record (Digital Acceptable)
Unscripted Testing: Ad-hoc (with least-burdensome documentation)	Testing of requirements or functions with no test plan	Details regarding any failures/deviations found	<ul style="list-style-type: none"> • Summary description of requirements or functions tested • Issues found and disposition • Conclusion statement • Record of who performed testing and date
Unscripted Testing: Error guessing	Testing of requirement or function failure modes with optional listing of expected failure modes in advance	Details regarding any failures/deviations found	<ul style="list-style-type: none"> • Summary description of failure modes tested • Issues found and disposition • Conclusion statement • Record of who performed testing and date
Unscripted Testing: Exploratory Testing	Establish high-level test plan objectives for requirements or functions (no step-by-step procedure is necessary)	<ul style="list-style-type: none"> • Pass/fail for each test plan objective • Details regarding any failures/deviations found 	<ul style="list-style-type: none"> • Summary description of requirements or functions tested • Result for each test plan objective – only indication of pass/fail • Issues found and disposition • Conclusion statement • Record of who performed testing and date
Unscripted Testing: Day in the Life Testing	Establish high-level test plan objectives for normal day to day activities to be challenged (no step-by-step procedure is necessary). Test using business process experience and knowledge, and against SOPs where available.	<ul style="list-style-type: none"> • Pass/fail for each test plan objective • Details regarding any failures/deviations found 	<ul style="list-style-type: none"> • Summary description of activities or business operation challenged • Result for each test plan objective – only indication of pass/fail • Issues found and disposition • Conclusion statement • Record of who performed testing and date

Table 25.2: Acceptable Assurance Approaches and Records (continued)
Adapted from ISPE GAMP RDI Good Practice Guide: Data Integrity by Design [36].

Assurance Approach	Test Plan	Test Results	Record (Digital Acceptable)
Scripted Testing: Limited	<ul style="list-style-type: none"> Limited test cases (step-by-step procedure) identified Expected results and values Independent review and approval of test plan 	<ul style="list-style-type: none"> Pass/fail for test case identified Details regarding any failures/deviations found and disposition regarding fails 	<ul style="list-style-type: none"> Summary description of requirements or functions tested Result for each test case – any critical values and indication of pass/fail Issues found and disposition Conclusion statement Record of who performed testing and date Record of who reviewed testing and date
Scripted Testing: Robust	<ul style="list-style-type: none"> Test objectives Test cases (step-by-step procedure) Expected results and values Independent review and approval of test cases 	<ul style="list-style-type: none"> Pass/fail for test case Details regarding any failures/ deviations found and disposition regarding fails 	<ul style="list-style-type: none"> Detailed report of assurance activity Result for each test case – any critical values and indication of pass/fail Issues found and disposition Conclusion statement Record of who performed testing and date Record of who reviewed testing and date

Scripted testing is dependent on detailed test cases that predefine the step-by-step instructions to be followed by the tester. Meanwhile, unscripted testing captures less detailed information. For example, with exploratory testing, the test case is limited to establishing high-level test plan objectives for features and functions (no step-by-step instructions) and supporting that with what was tested, by whom, when, and any issues found as a contemporaneous record. Even for ad-hoc testing, where there are no predefined objectives, there will still be a record of what was tested, by whom, when, and any issues found. Unscripted testing does not mean undocumented testing.

Unscripted testing techniques allow the behavior of the system to determine the path forward and are included in Section 25.6.1. Exploratory testing leverages the Plan – Do – Check – Act cycle. Testers simultaneously learn about the product and its defects, plan the testing work to be done, design, and execute the tests, and report the results. Good exploratory tests are planned, interactive, and creative. As the tester learns the application, they are better able to explore and test the application without preconceived notions about its behavior that come from scripted test cases.

Although unscripted testing test types have less detail in them as they do not contain step-by-step instructions, they can increase test coverage as the tester will create and evolve the test cases organically as they progress.

Unscripted testers should be qualified as testers and knowledgeable about the system in question to ensure the testing is focused on challenging the application in the context of the business process.

A comparison of unscripted and scripted testing approaches is provided in Table 25.3, which is an evolution of earlier Computer Software Assurance (CSA) work [20].

Table 25.3: Comparison of Unscripted and Scripted Testing Approaches

Adapted from ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management [20].

Unscripted Testing	Scripted Testing
<p>Used to supplement scripted testing to improve defect detection on high-risk functions.</p> <p>Critical thinking could support using unscripted testing to challenge functions with low and/or medium risk, and using it as part of a spectrum of tests challenging high-risk functionalities.</p>	<p>Used alone or in combination with unscripted testing approaches to challenge functions with high risk to patient safety, product quality, and data integrity.</p>
<p>Used to uncover software defects or errors associated with poorly defined/implemented specifications. The dynamic test design during execution allows functionality to be explored against the system specifications and other artifacts.</p> <p>Repeatability is lower due to the absence of a detailed step-by-step test script.</p>	<p>Tests against specifications to confirm fitness for intended use using test scripts that can be reviewed and approved in advance. It tests the known functionalities and may miss issues not identified at the time of test design. The detailed scripted nature of the testing discourages exploring areas of risks identified during test execution, e.g., the unknowns that may appear when the system is being tested.</p> <p>Scripted testing, manual or automated, can provide the basis for regression testing to capture the impact of changes or updates.</p>
<p>Relies on the tester's intuition, knowledge, and testing experience to explore and challenge the functionality of an application using the specifications, user manual design documents, etc.</p>	<p>Test cases use the specification as the standard to be verified.</p> <p>By its very nature scripted testing may not uncover defects arising from poorly defined specifications as scripted testing follows the specification verbatim.</p>
<p>Aims to test both expected and unexpected user/system behaviors.</p> <p>More sensitive to inexperienced testers and/or lack of system knowledge than scripted testing, as the scenarios and challenges are created dynamically during execution.</p>	<p>Tests are designed to confirm expected user/system behaviors.</p>

Each scripted manual test case should, where possible, include the following:

- Unique test reference
- Cross-reference to controlling specification
- Title of test
- Description of test, including the test objective
- Test steps – a step-by-step description of the actions to be performed by the testers along with the expected results
- Acceptance criteria – the defined set of expected results that should be met for the test to be deemed to have passed

- Pretest steps – including any test prerequisites or setup
- Data to be recorded – a description of the test-specific data to be collected and recorded. This can be input, output, or descriptive data and should include serial numbers of any test equipment used and supporting calibration certificates where necessary. Any requirement for screenshots and other electronic documentation should be specified.
- Post-test actions – this optional section details those actions required to return the system to a known state. Examples include resetting process parameters, putting the system in a safe state, or rebooting the system.

Each unscripted test case (except for ad-hoc testing) should, where possible, include the following:

- Unique test reference
- Cross-reference to controlling specification
- Title of test
- Description of test, including the test objective
- High-level instructions on how test objective can be met

25.5.3 Executing Tests

The quality of testing is significantly impacted by the knowledge and skill of the tester, therefore:

- Day in the life testing needs to be executed by a skilled business process user.
- All other manual testing needs to be executed by a skilled software tester; where unscripted testing is used, knowledge of the system under test is also essential.
- All staff responsible for test execution, including end users, should be trained in test procedures and should be able to demonstrate sufficient confidence in operating the system under test.
- Training should be documented.

The identity of the tester should be recorded for all executed test cases. It is disproportionate and unnecessary to use test witnessing or to require the tester to initial every test step to affirm they followed the instructions.

Automated test tools require a documented assessment of their adequacy prior to use. Detailed guidance on automated tools and their use in Agile approaches is available in *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20]) and in Appendix D9.

If calibration of necessary test equipment is required, it should be performed and documented. Calibration equipment should be certified, traceable to national standards and referenced in accordance with the customer's procedures; see *ISPE GAMP® Good Practice Guide: A Risk-Based Approach to Calibration Management (Second Edition)* [65].

Testing activities should be recorded following good documentation practice irrespective of whether manual or automated testing is used. Test records and evidence should be retained for subsequent review and inspection. Expectations regarding test evidence, including the recording and reporting of test pass/fail results, can be defined at a policy level within the QMS or within the test-management process. Excessive hard-copy test evidence continues to be an area for efficiency improvement. In many cases, prolific screen shots do not add value and are unnecessary.

It is recommended to apply critical thinking when defining the need for test records and evidence, to:

- Ensure that even where testing is not scripted in detail, the objective of the test remains clear, and the results record adequately what testing was carried out. Evidence of a test case result can be as straightforward as the tester indicating if a test passed or failed.
- Ensure that test evidence is only collected for proving steps that are not inherently covered by evidence from another test.
- Adopt an exception-reporting approach to recording detailed results. That is, if the system response matches the expected results, a simple “Pass” can be recorded. However, if the system responds in an unexpected manner, the tester records both a “Fail” and a description of how the response differed from the expected results, as this additional detail is helpful in determining the root cause and corrective action. This eliminates time wasted capturing elaborate test evidence (e.g., excessive screenshots or recording detailed descriptions of the observed response) when a simple pass or fail (or recording the value or initialing a design specification statement) is adequate to confirm that the test has been completed.
- Recognize that routine requirements for screenshots as objective evidence bring little value to verification activities. There are however certain situations where they offer a practical benefit over manual recording when detail is needed, for example, when there is a specific need for a before and after comparison of detailed or complex data, such as an audit trail or report.

The use of automated testing tools is encouraged to minimize manual collection of evidence as the artifacts within the tools are themselves evidence.

25.5.4 Reviewing and Reporting Completed Tests

Completed testing should be reviewed to verify that the computerized system is fit for its intended use. While test evidence should be reviewed to confirm completeness and accuracy of testing activities and conclusions, there is little value in:

- Looking for minor errors in documentation that have no impact on patient safety, product quality, or data integrity
- Scrutinizing test evidence in extreme detail to assess whether the tester has exactly followed the test instructions for non-critical functionality

Unusual patterns of test failures associated with particular authors and/or individual testers should be investigated to determine whether there are any wider testing implications on the rigor of completed testing.

Test reports should be produced that summarize activities and findings and state the final conclusions. Many automated test management tools can present a dashboard showing the test cases completed, any outstanding test cases, and any failed test cases, as an alternative to manually created test reports.

25.6 Types of Testing

This section discusses the types of testing that should be considered when developing the test strategy. The test cases for the different types of testing can be created as scripted or unscripted cases, as discussed in Section 25.5.3.

25.6.1 Test Types Unique to Unscripted Testing

There are some types of testing which are unique to unscripted testing:

- *Ad-hoc testing* is unscripted testing performed without planning or predefined documentation. It is aimed at finding defects as early as possible.
- *Error guessing* is a testing technique designed to expose anticipated and potential defects based on the specialist tester's knowledge and experience of failure modes.
- *Exploratory testing* is experience-based testing where the tester spontaneously designs and executes tests based on existing specialist testers' knowledge and experience, prior exploration of test items (including results from previous tests), and typical common software behaviors and types of failure and defects.
- *Day in the life testing* allows users of the system to carry out their normal day to day activities in the system (as they would post go live) to uncover issues not identified by scripted testing. These testers need real-life business process experience and knowledge, and may be drawn from the intended routine users of the system.

25.6.2 Other Test Types

All other test types can be created as scripted or unscripted test cases. Two general types of testing activities may be identified:

- *White box testing* is also known as clear-box testing, code-based testing, or structural testing. Test cases are identified based on the internal structure of the component or system. Sources of the internal structure include source code knowledge, knowledge of detailed design specifications and other development documents.
- *Black box testing*, also known as functional testing, is based on an analysis of the specification, either functional or nonfunctional, of a component or system without reference to its internal structure.

Specific types of testing should be considered, depending on the complexity and novelty of the system and the risk and supplier assessments of the system to be tested, including:

- *Normal Case testing* (Positive Case or Capability testing) challenges the system's ability to do what it should do, including triggering significant alerts and error messages, according to specifications.
- *Negative Case testing* (Invalid Case or Resistance testing) challenges the system's ability to correctly prevent actions that the specifications state or infer should be prohibited or blocked.
- *Repeatability testing* challenges the system's ability to repeatedly do what it should, or, when dealing with real time control algorithms, confirming the algorithms continuously and correctly fulfill their function.
- *Performance testing* challenges the system's ability to do what it should as fast and effectively as it should, according to specifications.
- *Volume/Load testing* challenges the system's ability to manage high loads as it should. Volume/Load testing is required when system resources are critical, and is typically automated.
- *Structural/Path testing* challenges a program's internal structure by exercising detailed program code.

Automated testing has gained traction since the previous version of this Guide, and in Agile approaches it is now common for test cases to be coded earlier. This allows for prolific use of:

- *Automation First*: automating testing (e.g., functional, acceptance) as part of the implementation and validation effort
- *Smoke testing*, that is a superficial level of testing to check critical functionality as a measure of the stability of each system build
- *Regression testing* to challenge the system's ability to still do what it should after being modified according to specified requirements, and that portions of the software not involved in the change were not adversely affected. Regression testing using automated test cases is used in Agile approaches to verify the build prior to release.
- Software tools, including automated testing tools, are discussed in detail in Appendix D9.
- Testing approaches continue to evolve and improve, and more advanced approaches such as using artificial intelligence to drive testing may become more widespread in the future.

25.6.3 Contractual or Acceptance Testing

There may be a need for specific tests to satisfy contractual requirements, which are typically called acceptance tests. Typically, acceptance tests are a predefined set of functional tests that demonstrate fitness for intended use and compliance with user requirements, and are related to meeting project milestones, the supply of key deliverables such as training materials or user manuals, and other commercial issues.

This approach is often used for automated equipment and process control systems. For further details see the *ISPE GAMP Good Practice Guide: A Risk-Based Approach to GxP Process Control Systems (Second Edition)* [61]. In such circumstances the test strategy and planning should leverage these tests to satisfy GxP verification requirements and avoid duplication.

Acceptance may be carried out in two stages, factory acceptance and site acceptance.

- *Factory Acceptance Tests (FAT)* are performed at the supplier site before delivery to show that the system is working well enough to be installed and tested on-site.
- *Site Acceptance Tests (SAT; sometimes called System Acceptance Testing)* show that the system is working in its operational environment and that it interfaces correctly with other systems and peripherals

These acceptance tests may be leveraged in support of, but do not replace the need for, process qualification including Process Performance Qualification (PPQ) where relevant.

The environment for acceptance testing (e.g., validation or production) should be defined.

25.7 Test Environments

Testing may take place in different environments during a project, which may include:

- Development environment where prototyping or programming takes place
- Formal testing environment
- Operational environment where the system is in its target environment

The use of multiple environments provides the opportunity to assess changes before they are rolled out to production and to provide a target for restore testing of backup data, thus reducing risk to the live operational environment and data.

The DevOps environment used in Agile approaches may be replicated into multiple test environments on demand to accommodate the needs of different groups and even SaaS consumers. The test environments, whether on-premise or hosted by the SaaS provider, need to be functionally equivalent for the testing to be representative and effective. Differences should be assessed and mitigated to ensure the testing is representative for testing and verification purposes. For example, missing endpoints in a formal test environment can be mitigated using service virtualization to emulate the endpoint, manual review of the inbound and outbound transaction, etc.

It is essential that a record is kept of the environment in which a test case was run, and the current build and configuration of that environment. Note, this record may reside in an automated tool (e.g., continuous integration orchestration). As stated in the *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36] and in Section 25.8.4, there is no benefit to repeating testing already performed in another, but still adequately controlled environment just because it was not the formal test environment.

It is the scope, rigor, and coverage of the testing that ensures the quality of the system as implemented; the quality and business gains are achieved by leveraging the full range of completed testing. Any significant differences between the environments used for testing should be assessed to determine the equivalency of test results, that is, to justify that similar results would have been achieved in the operational environment. Critical thinking would note this is analogous to how change controls are handled in production. There is no 100% retesting of the system when a change is applied; instead the change is assessed for its impact on the validity of testing.

25.8 Leveraging Previous Testing Including Supplier Testing

There are many different life cycle models for software development, including:

- Waterfall
- V
- Spiral
- Prototyping
- Agile

These are all equally acceptable. Whatever model is used, the supplier should define the implementation of the model, including necessary quality controls, and describe the way it is used to demonstrate that the computerized system meets their specifications.

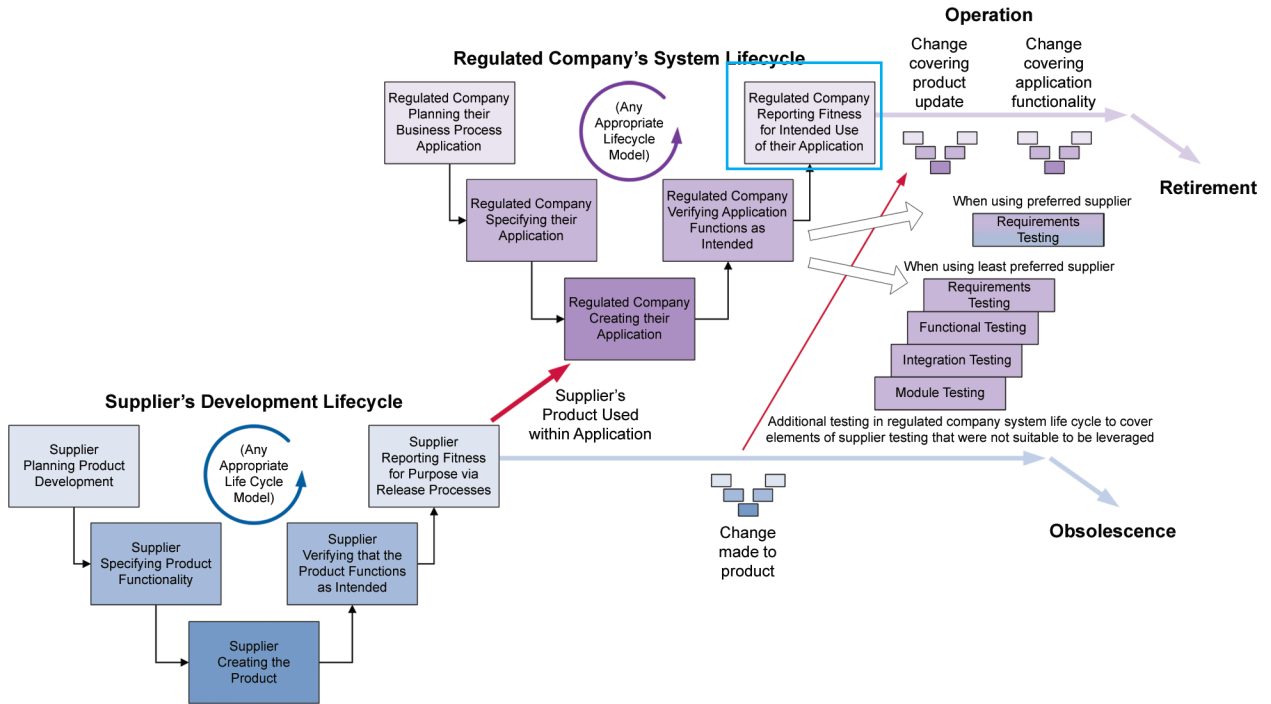
Testing strategies should be established, implemented, and reported accordingly. More information on software testing is given in ISO 90003 [66], various IEEE standards [67], and other publications. The International Software Testing Qualifications Board® (ISTQB) [68] offers formal training and accreditation in software testing.

25.8.1 Supplier Testing

The more extensive and effective the testing done by the supplier during their life cycle, the less testing the regulated company may need to do. If the supplier testing is confirmed as sufficient to demonstrate that the system functions correctly against supplier specifications, then the regulated company verification activity may be limited to a subset of requirements testing, as shown in Figure 25.3.

Figure 25.3 Supplier and Regulated Company Interaction

Adapted from ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management [20].



Common test types executed during a supplier development life cycle are:

- *Acceptance Testing for Purchased Hardware and Software:* Purchased hardware and software should be subject to acceptance testing before being used for the system's development.
- *Unit/Module Tests:* Stand-alone tests for software components, ensuring readiness for further integration into the complete system. Such tests are best developed at the same time that the software is developed.
- *Integration and Functional Tests:* Testing of integrated software components, sub-systems, and the complete system.

In order to leverage supplier testing, the regulated company should first consider:

- The extent, coverage and quality of the supplier testing to determine how much reliance can be placed on the supplier testing and what additional testing will be needed. An SME review of supplier test cases and records may be required to justify this.
- How supplier test records can be made available for appropriate assessment and review by the regulated company
- How traceability between the regulated company's requirements and the supplier's test cases will be established and maintained, if required

Access to supplier test records can be complicated when such records reside in requirements management tools, automated test tools, etc., rather than existing as discrete documentation. This should not, however, be used as a reason to insist the information is recreated in a document format, and an agreement should be reached on how those records can be viewed when needed.

25.8.2 Ongoing Life Cycle Maintenance

As Figures 25.1, 25.2 and 25.3 show, new releases of the software by the supplier will result in the need for the regulated company to actively take measures to maintain the validated state after upgrading. Critical thinking and risk management should be used to determine the extent of the effort required.

When supplier testing of new functionality and supplier regression testing of existing functionality will be leveraged by the regulated company, provision may need to be made for access to this testing as part of accepting the new software release.

As a minimum, the regulated company will need to verify the updated software remains fit for their intended use. Key to the regulated company determining the necessary testing required for a change and their acceptance of a change is their ability to understand and risk evaluate the changes in light of the criticality of the business process supported by the computerized system. To be able to efficiently evaluate the change, the supplier should provide a clear understanding of the changes being implemented. Regression testing of the configuration and workflows – whether manual or automated – may be implemented by the regulated company to confirm the latest software version implemented still provides a system fit for the regulated company's intended use, as part of operational change management.

25.8.3 Testing in Multiple Environments

Testing in environments other than the validation or production environment contributes to finding defects and confirming the system is fit for intended use.

The differences between the supplier's development environment, any initial exploratory testing in a sandbox, or regulated company's development environment, and the production environment should be assessed to determine its impact on the validity of the testing. This will be specific to the type of testing; for example, load testing conducted in a test environment with limited data and users will not be representative of the true load experienced in the operational environment.

However, where there is no impact to the particular test types and cases arising from the difference in the environments (for example, testing of audit trail functionality), the testing should be leveraged no matter what environment it occurred in, and there is no benefit to repeating that testing in the formal test or production environments. See Section 25.7 on functional equivalency of environments.

25.9 Testing Applied to Different Software Categories

This section provides practical considerations when planning testing.

25.9.1 Aspects that Apply to all Systems

25.9.1.1 Installation Testing of Hardware/Software

Although the term is not used in this Guide, which refers to installation verification activities in general, many companies still use the traditional process validation term Installation Qualification or IQ. The purpose is to verify and document that system components are combined and installed in accordance with specifications, supplier documentation, and local and global requirements. Installation testing provides a verified configuration baseline for subsequent verification and validation activities, and verifies any installation methods, tools, or scripts used. This forms the basis for configuration management of the installed system. This "static" approach to installation verification is typically applied to process control systems and automated equipment.

For cloud-native or virtualized software, the use of automated installers combined with configuration management tools and service management tools will be more effective than conventional installation approaches, and if they provide dashboard functionality then a review-by-exception approach can replace many of the static verification activities.

Whatever the approach used toward installation verification, the availability of appropriate technical information should be verified. The necessary information will vary widely between different types of systems, but examples include:

- User and technical guides
- SOPs
- Training schedules
- SLAs
- Security procedures
- Log books
- Hardware inventory
- Instrument lists
- Specification sheets
- Certificates and calibration procedures
- Loop sheets
- Piping drawings/P&ID
- Equipment list and specification sheets
- Software inventory (including installation procedure, system software list, application software list, data list, initial data settings for start-up)
- Preventive maintenance program
- List of critical spare parts

25.9.1.2 Points to Consider for all Systems

The following is an aide memoire only and does not replace the need to apply critical thinking and a risk-based approach to the scope and rigor of testing. It should be used simply as a reminder to help ensure appropriate test coverage of the installed system.

Test coverage may include:

- Power failure testing, especially
 - Prevention against loss of critical data or loss of control action
 - Ease of controlled restart

- System access and security features
- Data audit trails for creation, modification, and deletion of GxP records, and mechanisms for logging of other critical actions including manual interactions
- Manual data entry features, input validation
- Electronic signature features
- Alarms and error messages
- Critical calculations
- Critical transactions
- Transfer of critical data into other packages or systems for further processing
- Interfaces and data transfers
- Backup and restore
- Data archival and retrieval
- Ability to deal with high-volume loads especially if the system is accessed by many users as part of a network application

25.9.2 Typical Testing Activities for a Standard Product

These are software products that are used off-the-shelf (i.e., which are either not configurable for a specific business process or those that offer limited configurations using factory-provided values or ranges (also called parameterization)) and are typically classified as GAMP Category 3.

Testing should focus on:

- Installation testing as described above
- Requirements testing that demonstrate fitness for intended use; this may include testing the system functionality against requirements depending on risk if not adequately addressed in the supplier testing. For very simple measurement systems regular calibration may substitute for testing.
- Any further or more rigorous tests as a result of risk and supplier assessments
- Acceptance of the final system information from the supplier, including specifications, manuals, and drawings, if not already covered
- Any other relevant aspects listed in Section 25.9.1.2 not already covered

25.9.3 Typical Testing Activities for Configured Functionality

A common type of computerized system involves the configuration of commercially available software products running on standard hardware components. Configurable software components enable configuration of user-specific business processes into one or more workflows, specific to methods, or products, or processes, etc. Software products that are configured for a specific business process typically are classified as GAMP Category 4, even though they will also contain components of Categories 1 and 3, and possibly Category 5 components if there are some customized elements. Appendix M4 provides more detailed discussion around multi-category systems.

As the configuration is typically specific to the individual company's intended use, there will be less opportunity to leverage supplier verification and more need for regulated company verification activities.

Testing should focus on:

- Installation testing as described above
- Configuration testing – verifying that the system has been configured in accordance with the agreed specification. The tests could take the form of inspections or check of configuration documentation.
- Functional testing – challenging functionality that supports the specific business process based on risk and supplier assessments. The regulated company may not need to complete some or all of this testing if the supplier testing has been assessed and judged to sufficiently verify the functionality.
- Requirements testing that demonstrate fitness for intended use – this may include challenging the configuration and user-defined workflows to ensure the needs of the business process have been met in the configured system
- Acceptance of the final system information from the supplier, including specifications, manuals, and drawings, if not already covered
- Any further or more rigorous tests as a result of risk and supplier assessments
- Any other relevant aspects listed in Section 25.9.1.2 not already covered

25.9.4 Typical Testing Activities for a Custom Application

Some computerized systems or software components are developed to meet individual user requirements, where no commercially available solution is suitable. Such software is classified as GAMP Category 5, although it is important to understand that even a fully customized development will leverage standard software modules and libraries to which the approach in Section 25.9.2 may be applied.

Most modern developments, especially for large or complex applications, will use an Agile development approach as detailed in *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20] and in Appendix D8.

Traditional linear-sequential developments may still be used for small or simple applications, such as interfaces and macros. Linear-sequential developments may also be used for other reasons. For example, where systems integrators are engaged by the regulated company with the need for both parties to work collaboratively on the activities, the limitations of requirements' management/traceability tools. etc., may preclude such collaboration.

In such cases, and based on satisfactory supplier and risk assessments, a testing approach based on the four levels of module (unit) design, integration, functionality, and requirements typically is applicable.

Testing should include the approaches detailed in Section 25.9.3, with the addition of:

- Code review for new code required as a result of risk assessments, see Appendix D4
- Software module testing to test software modules defined in the design specification
- Software integration testing to test that the modules work when operating together

26 Appendix D6 – System Descriptions

26.1 Introduction

This appendix provides guidance on the contents of System Descriptions.

EU GMP Annex 11 [32], requires that there is an up-to-date description of critical GxP regulated computerized systems:

“For critical systems an up to date system description detailing the physical and logical arrangements, data flows and interfaces with other systems or processes, any hardware and software pre-requisites, and security measures should be available.”

The need for a system description may be covered by one or more existing specifications or other documents, or a separate document may be produced.

The principal use of such a document is to help new users and regulators understand what the system does, and as such is written in non-technical language as far as possible.

26.1.1 Changes from GAMP 5 First Edition

This section has been updated to reflect current ISPE guidance document format.

26.2 Scope

This appendix covers information that may be required for a wide range of GxP computerized systems. Not all the information will be required or relevant for all systems. The level and detail of information should be based on system risk, complexity, and novelty.

26.3 Guidance

26.3.1 General Guidelines

The system description should be maintained up-to-date throughout the life cycle of the system.

For complex systems spanning multiple departments or sites (e.g., Enterprise Resource Planning (ERP)), a separate document may be appropriate. For simpler systems it is common practice to include the system description in another specification or other document.

The development of the system description may begin early in the life cycle and evolve iteratively as the development process progresses. A complete system description meeting regulatory expectations should be established before the system is released for operational use.

The system description should be subject to change control and periodic review.

26.3.2 Contents of the Document

Listed below are topics that may be required in the system description depending on the nature and type of system. The guidance provided is intended to be neither prescriptive nor exhaustive.

The system description should cover only the main features of the system. Detailed information on specific topics should be included in other specifications and not repeated.

26.3.2.1 Introduction

This should explain the context of the system within the business process and regulated company in general. This should be considered from the following perspectives as appropriate:

- Departmental
- Site-wide
- Division-wide
- Global

26.3.2.2 Main System Functionality

This is a description of the key functions of the system, both GxP and non-GxP (many of which could be business critical). The functions may be grouped to maintain the description at a high level. The use of diagrams is encouraged to explain relationships between key functions.

26.3.2.3 Regulatory Impact

This should include a description of the key GxP functions of the system. The impact that the system has upon patient safety, product quality, and data integrity should be considered.

Other regulatory requirements should also be covered as appropriate.

26.3.2.4 The Computing Environment

This may be covered by a high-level diagram of the architecture supporting the system covering, as appropriate:

- The infrastructure that supports the system (e.g., server configurations, storage arrangements)
- Interfaces to users
- Interfaces to equipment
- Interfaces to other systems
- Interfaces outside the company
- The flow of data through the interface
- Security features such as firewalls

26.3.2.5 System Components

An indication of the main hardware and software components should be provided. This may include information regarding servers and storage devices, as well as the operating systems, databases, and application layers. It should make reference to any configuration documentation relevant to the system. A detailed inventory of all components is not required here.

26.3.2.6 System Interfaces

This is an overview of the key interfaces to other systems and equipment, and the data flowing to or from the system.

26.3.2.7 Access Control

This is an overview of the access control features of the system, both physical (if relevant) and logical.

26.3.2.8 Security Controls

This is an overview of the established system security controls both physical and logical. These should include software for the protection of data and records, e.g., virus protection software.

26.3.2.9 Electronic Records and Signatures

An indication of the types of electronic records created and managed by the system, and the type of electronic signatures used, should be provided, if relevant.

26.3.2.10 Glossary

Definitions of any terms that may be unfamiliar to the readership of the document should be provided.

27 Appendix D7 – Data Migration

27.1 Introduction

This appendix provides guidance for the migration of electronic data in the regulated environment. This covers planning, execution, and reporting of the activity.

Migration activities should be focused on aspects critical to patient safety, product quality, and data integrity and their extent should be commensurate with risk, complexity, and novelty.

27.1.1 Changes from GAMP 5 First Edition

Minor changes have been made to clarify some basic concepts and principles.

27.2 Scope

This appendix can be used by, or on behalf of, regulated companies to define quality and compliance-related procedures and standards for planning, executing, and verifying data migration activities.

This appendix does not cover routine transfer of data from one system to another as part of an ongoing business process. Such situations should be covered by normal specification and verification activities.

27.3 Guidance

27.3.1 General Guidelines

Data migration is the activity of transporting electronic data and associated metadata from one system to another, or simply the transition of data from one state to another. Data migration is an activity that can occur often during and/or at the end of the life cycle of the various computerized systems used by regulated companies. In practice, data migration efforts can vary greatly in scope, complexity, and risk, example:

- An *in-place* version upgrade of a database or application
- Data conversion (e.g., from one supplier's database to another)
- *Same system – different platform* migration (e.g., transporting an application's data from one server platform to another server platform, such as from an on-premise platform to a cloud service-provider platform)
- Migration from one source system to a different target system
- Migration from multiple source systems to a single target system

The complexity and risk of the data migration effort also may increase significantly if rules are used to select a subset of data from the source system, or if data is transformed (e.g., data type conversion, filtering, cleansing, aggregation, renormalization) prior to being inserted into the target system. The ultimate goal of any data migration is to have data that remains usable and retains its contextual meaning. Quality management controls should be in place to ensure that data migration efforts are successful, compliant, and repeatable.

Each data migration should be managed within the framework of a data migration plan and report, or through following an established data migration procedure.

27.3.2 Quality Management

27.3.2.1 System Life Cycle

Data migration may take place multiple times during the life cycle of a single computerized system, such as during:

- Initial system development and deployment
- Application upgrades
- System retirement

Despite this, organizations that have defined a system life cycle may not have a defined or documented data migration process. As with other life cycle phases and activities, data migration efforts will be more consistent and successful if the life cycle provides guidance for the requirements and expectation of data migration including roles and responsibilities, documentation requirements, quality and compliance controls, technical and verification activities, and project management and oversight. A data migration SOP may be the best method for describing and documenting the process expectations, including quality and compliance requirements. Just like every other life cycle activity, it must utilize a risk-based approach to ensure the effort is appropriately rightsized.

27.3.2.2 Risk Management

The life cycle should include an established risk management process with guidance for assessing risks that are specific to activities related to computerized systems. In addition to risks typically found with technology projects, the following should be evaluated when migrating regulated data:

- The inherent quality and compliance risk associated with the data being migrated, including but not limited to:
 - Impact upon patient safety, product quality, and data integrity
 - Risk associated with the business processes that the computerized systems involved are supporting
 - Risk associated with incomplete source data
- Risk to the business due to systems being unavailable or data being unreliable
- The level of complexity (e.g., multiple source or target systems; multiple phases; a high degree of data transformation)
- Technology risk due to the use of complex or leading-edge systems or tools
- Risks associated with the use of system vendors and third-party contractors to support and execute the migration

27.3.2.3 Configuration Management and Change Control

Migration of electronic data in the regulated environment should be performed under change control. Similarly, all appropriate data migration project documents and tools should be controlled using configuration management and good documentation practices, including ALCOA+ principles.

During a data migration project, system changes unrelated to the migration should be prohibited. This is because the success of a data migration effort depends on various characteristics of the systems (e.g., software versions, database schemas) remaining unchanged during the project. Unrelated system changes can increase the complexity of the data migration effort, which will in turn increase the overall project risk.

The change management process, as well as the migration plan, should also ensure that adequate communications occur with the impacted business areas and users of the system. These communication activities should be defined and include information about the timeline, scope, entry criteria, and completion criteria of the migration. It should include activities that need to be completed prior to the start of the actual migration process to ensure the reliability and accuracy of the source data, such as in-process and incomplete data cleanup prior to the migration, unavailability of the system, and activities required post-migration to confirm and/or monitor success. Proper communication with the impacted business areas and users prior to, during, and post-migration can have a significant impact on the perceived and actual overall success of the migration and should not be overlooked.

Typical good practice is to use one or more intermediate staging areas for data that has been extracted from source systems, prior to being loaded into the target system. A typical migration effort will include at least three system platforms or areas that should remain under configuration management during the entire project: the source system(s), the staging area, and the target system. It is also good practice to establish a rollback strategy in the event of a significant failure of the migration activities. This should include a full source-system backup prior to any migration activities so that the source-system data can be recovered if necessary.

27.3.3 Areas of Concern

27.3.3.1 Fitness of Software Tools for Intended Use

Data migration efforts typically involve the use of software tools to automate some or all of the extraction, transformation, loading, and verification activities. These tools tend to be GAMP software Category 1 infrastructure tools (e.g., database migrators, and verifiers purchased from a software supplier) or Category 5 custom application (e.g., SQL scripts, in-house developed programs).

The infrastructure tools and custom applications should be fit for intended use. The rigor of related specification and verification activities should be commensurate with associated risks. Depending upon the scope, complexity, and customization of the software tools being used, required deliverables may range from evidence of basic testing to full software specifications and formal verification. An SME should ensure that appropriate life cycle activities and deliverables are identified and executed. The quality unit should review and approve key documentation in accordance with company procedures.

For software tools that move or transform data, there are three principal areas of risk:

- Data will be moved or transformed incorrectly or incompletely (incorrect data format, accuracy and/or completeness of the information, loss of information, orphan records, etc.),
- Data residing in the target system will be harmed (deletion and/or overwriting of data, breaking and/or modification of data links, loss of metadata, etc.)
- In the case where not all data is being migrated, residual data in the source system is adversely affected by the removal of the migrated data

In theory, these risks would require that a high level of effort be applied to demonstrate the fitness for intended use of software migration tools. In practice, however, through the use of data verification the risks can be significantly mitigated and therefore the rigor of software migration tool verification can be reduced. It should be noted that the development and approval of a data mapping table (i.e., fields from the source-system data model mapped to the target-system data model) is still necessary when using software migration tools.

27.3.3.2 Data Verification

Data should be verified each time it is moved (either within a system platform or from one system to another) or its state is transformed. There are two general types of post-migration data verification: test environment verification and operational environment verification.

In test environment verification a target test system is initially populated with data, then a migration test run is performed, and finally data in the target test system is verified to show that all required data migrated successfully and without adversely impacting existing data. This verification provides objective evidence that the data migration software tools are fit for intended use, and also provides a level of confidence in the overall migration process. A typical approach during this step is to work with a relatively small amount of data, which can then be completely verified to ensure that no data errors occurred. The results of this verification should be appropriately documented to provide evidence of the success of the process and retained as part of the migration process.

The intent of operational environment verification is the same: to verify the outcome of the migration process on both migrated and existing data. The amount of data involved, however, is typically very large and, therefore, more difficult to verify. Automated software tools can be used to verify 100% of data in the target environment. The suitability of such software tools should be rigorously determined through appropriate testing and clearly defined verification processes with the outcomes clearly documented and maintained as part of the migration activities.

Data mapping and transformation are not the only issues to be addressed. An important part of data migration is confirming that all of the required data has been migrated. Verification techniques, such as checksum and checksum in two dimensions, can be used to corroborate complete data transmission and to locate failed records.

Objective evidence of data verification should be generated. Verification scripts and data sheets, screen shots, error logs, and hardcopy reports should be created when appropriate and feasible.

27.3.3.3 Reliability of Source Data

The reliability of the source-system data must be considered as part of the migration process. If the source system is used and maintained in compliance with regulatory requirements with appropriate source system's controls, the reliability of the source data prior to migration should exist. Through the use of a well-defined and well-controlled data migration process, sufficient assurance of the accuracy and integrity of the migrated data (i.e., reliability of the migrated data) can be obtained. (If the process for extraction of the source data from the source system to the staging area is not performed using standard controlled and validated processes from the source system, additional verification activities should be defined as part of the migration plan to ensure the reliability and accuracy of the source data.) To document this, the data migration plan should reference the appropriate source-system documentation.

If the reliability of the source system and data are unknown then two problems exist: first, the veracity of data migrated from the source system may not be verifiable; second, the migrated data will mix with reliable data already in the controlled target system. After the migration effort is complete, the trustworthy existing data and the questionable migrated data will be indistinguishable unless steps are taken to identify the migrated data as such (e.g., differences in record dates and notations in user-defined fields). If this is not possible, then the possible data inconsistencies should be documented, explaining the controls in place on the source system and justification of why the migrated data should be trusted.

27.3.3.4 Usability of Migrated Data in Target System

There are four main issues to consider relating to usability of migrated data on a target system:

- Target-system functionality does not allow performance of tasks previously carried out in the source system
- Lack of completeness of migrated data affects the usability of the data
- It may not be sufficient to migrate the data. Separate migration of metadata or configuration of the target system also may be required. For example, the source data has some access rights defined for it, such as user groups and user rights. Migrating the data may not normally migrate this metadata, which is normally separate from the data. However, these user groups and rights also may be required on the target system.

- Need for multiple migrations through various versions of a system to ensure full compatibility in the final target system (version). Consultation with the vendor(s) should occur to understand the changes associated with the sequential versions of system to appropriately understand and define the required migration plan to ensure final usability of the data in the target system.

27.3.3.5 Audit Trails

Audit trails can be problematic for data migration efforts. If the target system has an audit trail but the source system does not, documentation should be created reflecting that auditing for migrated records began when they were loaded onto the target system. If possible, these records should be distinguishable from records that were created on the target system (e.g., using a notation in a user-defined field).

If both the source and target systems have audit trails and it is technically feasible, the audit trail should be migrated along with the audited data. If it is technically unfeasible to migrate the audit trail (e.g., due to data transformation) or if it would constitute too great a risk to do so, then the original audit trail should be archived in a format that can be retrieved and used for support or investigational purposes.

When possible, a computer-generated audit trail should be created during the movement and transformation activities associated with the data migration effort, because this audit trail serves not only as a verification tool for the migration team but also as a historical record of changes to the data. If there are instances where the audit trail must be turned off during the migration effort (i.e., negatively impacts the reliability, accuracy, or integrity of the migrated data), the rationale for this decision should be appropriately documented and maintained as part of the migration process documentation. In these cases, the audit trail should be archived in a format that can be retrieved and used for support or investigational purposes.

27.3.3.6 Use of System Vendors and Third-Party Contractors

The use of system vendors and third-party contractors to perform data migrations is becoming more prevalent in the regulated environment due to the complexity and proprietary data base structures and data integrity controls. System vendors and third-party vendors should be assessed/ audited to ensure they are capable of performing the work and that tools utilized are validated for their intended use. Master Service Agreements (MSAs), SLAs, Statements Of Work (SOW), and/or quality agreements should be established to clearly define the project requirements, as well as the oversight and roles and responsibilities associated with the use of the vendors, including whose procedures will be utilized, the tools utilized for the project and accountability of validation of those tools, the training requirements, etc. All of this information should be maintained as part of the migration process documentation.

27.3.4 Data Migration Plan

Different data migration efforts can require very different activities and deliverables. This appendix is not intended to provide specific project or technical methodology guidance. However, every data migration project should have a data migration plan as a required deliverable. Similar to plans for computerized system development projects, the data migration plan serves as a high-level roadmap that guides project team members in performing a compliant and successful technical effort.

The data migration plan should describe the entire migration process, including as a minimum:

- Migration project purpose and scope, including if the migration will be completed in waves or sub-projects due to the overall size and complexity of the project
- System description(s)
- Roles and responsibilities, including the impacted business areas, migration team, vendors, etc.
- Required deliverables, including who is responsible for creating, managing, and approving them

- Risk-management strategy, including any risks associated with the use of vendors and third parties
- Configuration management strategy, including the source, staging, and target environments
- Software tool overview and strategy for ensuring compliance and fitness for intended use
- Migration steps, technical activities, and impacted business area activities
- Data mapping and modeling activities
- Transformation rules
- Data verification strategy and acceptance standards
- Cutover plan, paying special attention to impacted business area activities, especially pre- and post-migration that ensure the success of the migration project. The cutover plan should also include communication activities to ensure impacted areas are aware of timeline, scope, entry criteria, completion criteria, and unavailability of the system.
- Rollback strategy

The data migration plan should be approved by the process owner, system owner, quality unit, and SMEs as appropriate.

It may be appropriate to reuse the same data migration plan more than once. An example of such reuse is the deployment of an EDMS: the approved data migration plan can be used multiple times as different functional groups or geographic sites migrate their electronic documents into the system. Each time the plan is executed, a data migration report should be created. If the migration process is something that will be routinely executed, such as part of software and configuration updates, consideration should be given to the use of a procedure as the migration plan to ensure an efficient and consistent outcome.

27.3.5 Data Migration Report

The data migration report summarizes the activities that were conducted during the data migration effort. It describes any anomalies or deviations that were encountered and lists the results of verification activities (including objective evidence as appropriate). Because the report will be used to establish the reliability of the migrated data, it should clearly state the overall outcome of the migration activity (e.g., full success, partial success, failure).

If the migration is being performed in waves or sub-projects, the need for wave or sub-project migration reports should be considered, especially if the migrated data will be utilized in production prior to the completion of the entire migration project. A final migration report at the conclusion of the entire migration project should also be considered to completely close out the migration project.

The data migration report should be approved by the same individuals who approved the migration plan (e.g., process owner, system owner, quality unit, and SMEs as appropriate).

In the case where data migration is being performed as part of a computerized system project, such as a system replacement or upgrade, the data migration report can be documented as appropriate within the project, and may not need to be a separate document.

28 Appendix D8 – Agile Software Development

28.1 Introduction

Agile software development practices are widely adopted by many industries, including medical devices. Agile approaches focus on delivering quality and value to the customer at speed, and in an incremental fashion, enabling technical innovation and flexibility.

FDA’s Technology Modernization Action Plan (TMAP) [69] identified Agile software development and DevOps as foundational requirements for a modern FDA technology infrastructure. Such approaches are very well suited to the development of GxP regulated systems if correctly applied by trained and qualified practitioners supported by appropriate tools.

This appendix provides a summary of the principles underpinning Agile and illustrates how it can be implemented in a way that is aligned with GAMP 5 and GxP principles. The focus is on how to use well-implemented standard Agile processes to deliver software for GxP applications and does not advocate in some way modifying Agile for GxP, for example, by superimposing linear (V-model) activities. A more detailed description of how to apply Agile in a GxP environment is included in the *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20].

The planning, specification, verification and reporting activities within this Guide are not inherently linear, and the approach described is designed to be compatible with a wide range of other models, methods, and schemes including incremental, iterative, and exploratory models and methods. Figure 28.1 demonstrates the approach described.

Figure 28.1: Iterative and Incremental Approach to Achieving Compliance and Fitness for Intended Use [20]



The key principles behind Agile software development are that of discovery and of iteration (ongoing changes) as opposed to waterfall software development, where there is a linear flow of defining/collecting all requirements before transforming these into a complete set of functional and design specifications that are then configured/coded before testing commences.

With Agile software development, requirements are collected/discovered and then moved into development/configuration, testing, and release in iterative cycles. This typically requires the ongoing involvement of cross-functional teams including end users and business process owners.

There are a number of differing frameworks that support Agile, underpinned by the values within the Manifesto Statements for Agile Software Development [70]:

“Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan”

These principles need to be taken in the context of emphasis rather than a binary choice. For example, tools are invariably used, and form an important part of Agile software development, but the manifesto stresses that team collaboration is more important than simply tools/processes. [20]

28.1.1 Changes from GAMP 5 First Edition

This is a new appendix.

28.2 Scope

Agile software practices can be applied across many types of computerized systems, including custom application development, and configuration of configurable products in an incremental manner.

Where the regulated company is looking at less clearly defined scope/requirements, especially for customized developments, Agile is likely to be the preferred approach to commence with a discovery phase to develop the initial backlog requirements and enable faster initial system deployment and subsequent incremental development.

GAMP 5 discusses the interaction of supplier and regulated company life cycles (see Figure 2.1 in the Main Body). These may be both Agile or a mixture of Agile and linear approaches.

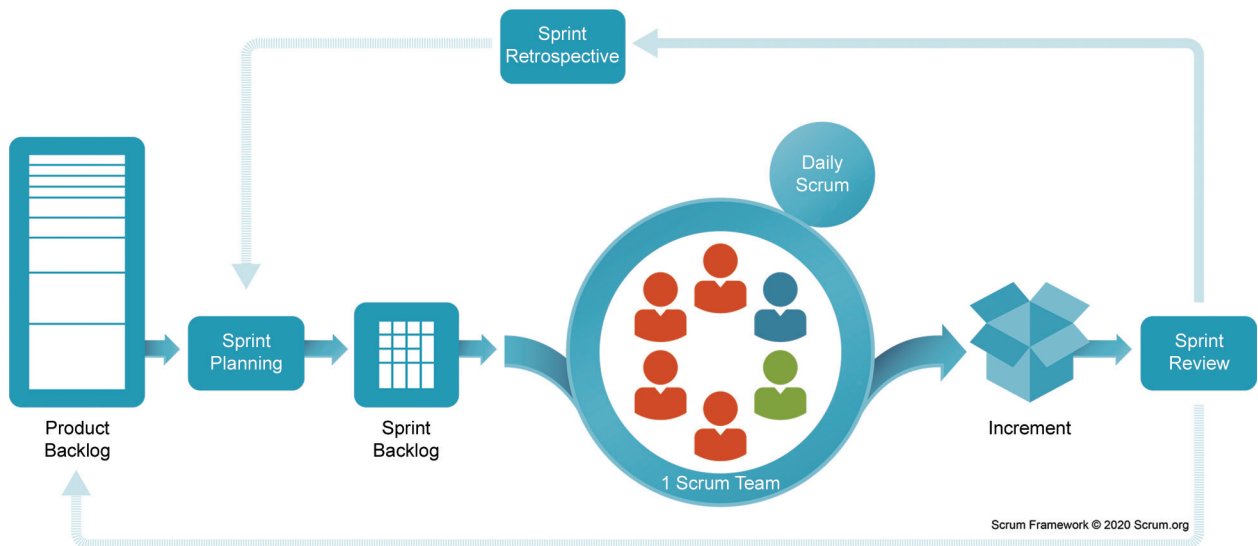
28.3 Guidance

28.3.1 Agile Basics

There are many different frameworks for Agile, but to illustrate the general principles the popular Scrum framework [71] will be used as the reference. The choice of framework can depend on factors such as scale, complexity, and dependencies, for example, larger programs of work may adopt more formality. Agile is not a reason to accept a lower level of quality or GxP compliance. The regulated user must ensure that the computerized system is fit for intended use. Evidence of this may be generated and maintained in tools and supporting systems, rather than in traditional documentation. Figure 28.2 illustrates the process for Scrum.

Figure 28.2: Scrum Framework Model [71]

Used with permission from Scrum.org, www.scrum.org.



- A Product Owner is responsible for collecting requirements into a product backlog, typically containing larger sets of requirements (epics) which are further split into sets of (user) Stories.
- The Scrum cross-functional team takes a subset of the backlog items and develops/delivers and tests these during a sprint. This includes backlog refinement activities to ensure the items are sufficiently defined and ready to be included.
- A review is conducted of the results of the sprint, called the sprint retrospective, which evaluates the business value of the completed sprint, and drives continuous improvement in the working methods and process.
- The cycle then repeats with another subset of the backlog.

Sprints are governed by time and designed to be a short period, 2 to 4 weeks, for example. Backlog items unable to be completed during the Sprint return to the backlog.

Scrum teams typically are small, multidisciplinary, and are responsible as a whole for delivering to time and quality. The Scrum product owner provides the customer link (e.g., to the business process owner) and the Scrum master provides coordination and helps ensure the team adheres to the rules, governance, and processes agreed to by the team.

Regulated company quality roles typically provide oversight and (in line with the critical thinking and risk-based approach) subject matter expertise on regulations and potential areas of product quality, patient safety, and regulated data impact associated with the business process the system will be supporting. Quality should work with the Scrum product owner, Scrum master, and business process owner to determine how and when they should be engaged. Ultimate accountability for quality and compliance lies with the regulated company. [20]

In combination with the model in Figure 28.2 are the concepts of a Definition of Ready (DoR), Definition of Done (DoD), and also perhaps a MVP, to determine the activities to be completed before a requirement (epic/user story) can be considered complete and to agree/define the minimum set of epics/user stories that need to be completed for the initial release of the product.

Due to the iterative nature of Agile software development, there is a risk of new functionality impacting previously developed and tested software, and therefore regression testing, often performed using automated test tools, is typically applied to ensure stability of the system within sprints.

It may be tempting to try and map artifacts created using Agile methods to the traditional life cycle documents (for example, mapping user stories/epics to the RS); however, it is more effective to start from established good practice Agile artifacts and not force-fit the traditional documents. Mapping against the high-level project stages as illustrated in Figure 28.1 is sufficient, and many activities such as a supplier assessment, user training, and periodic review, are conducted irrespective of the development method used.

28.3.2 User Requirements

Many regulated companies expect to see a fully formed URS early in the development life cycle when developing a GxP regulated system. With Agile, by planning and storing requirements differently and following Agile principles, teams can successfully deliver effective and useful software in a controlled way that is compliant with GxP regulations, and where there is a clearly identifiable set of fulfilled requirements at the point the system is released into the live GxP operational environment.

Ultimately the delivered description of system functionality is provided by the list of completed Agile artifacts, such as epics and user stories, which can be obtained directly, or as reports, from the Agile software development tools, including audit trails, version, and history.

28.3.3 Mindset and Culture

Mindset reflects how culture, values, and priorities manifest themselves throughout the organization. A traditional mindset is one with fixed standards, workflows, and expectations; a discovery mindset encourages acting, learning, and rapid continual improvement.

There is a perception that full and accurate sets of requirements can be defined at an early stage and end dates can be fixed for large linear programs. As a result, final testing phases are often squeezed and reduced because earlier phases took longer than planned.

For Agile, a shift in culture and behaviors according to the following principles is required [72]:

- *“An agile-first mindset and ways of working, using a Discovery Mindset, rather than a Certainty Mindset*
- *A culture of innovation and continuous improvement*
- *Empowered teams with the ability to deliver across geographies*
- *A consistent and repeatable way of agile delivery*
- *Much-improved alignment across all stakeholders”*

The discovery mindset is a powerful approach to changing the way teams plan and deliver. This also applies to business process owners and quality groups. With the traditional approach they are involved at the beginning (requirements) and end (acceptance testing). Agile is a continuous cycle of involvement and the resourcing and planning impact of this needs to be taken into account.

28.3.4 Tools Instead of Documents

Agile is typically used with sets of tools providing control over the product backlog, configuration, testing, and release activities enabling shorter cycle times, higher quality in delivery, and better user engagement. From a quality and GxP compliance perspective, tools perform an essential role in demonstrating that the system is fit for intended use, functionality can be traced back to requirements, and testing completed.

Tools manage a variety of Agile activities such as user-story risk assessments, testing, and traceability, and a comprehensive toolset can provide an integrated solution to manage them all. With tools in use the need for documentation and manual sequencing of events in many cases will be eliminated or, at a minimum, be significantly reduced, for example, by workflows with stage gates to ensure one activity must be completed before the next step in a process can be started. This also eliminates the risk of transcription errors, versioning errors, content inconsistencies, or anomalies where parallel electronic and documentation-based systems are in use.

A backlog with continuously changing user stories is extremely difficult to manage on paper, thus, demonstrating control (a key GxP principle) is equally difficult. A tool can provide an efficient and effective means to manage user stories in a backlog throughout updates and changes.

Reports may also be generated from tools, providing baseline status at points in time (for example, of user stories), and used, for instance, during regulatory inspections to provide evidence of status.

Tools provide benefit in measuring, managing, and achieving high-quality results. Tools can provide integrated bug/defect management and can reduce the chance for human error. Table 28.1 shows a representative listing of tools in common use with an indication of how they support the development of high-quality software. New and enhanced tools continue to be developed and used over time.

Table 28.1: Common Agile Tools

Adapted from ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management [20].

Tool	Quality
Backlog Management Software	Provides real-time status of requirements including traceability (e.g., to testing), and approval status (for example accepted, done), and includes an audit trail for changes and reporting functionality.
Testing Management	Provides evidence of test status, using automation and regression test functionality to provide additional assurance of thorough testing. Can provide functionality to perform negative and stress testing and reduces the likelihood of test script errors that occur when tests are developed manually.
Orchestration Software	Provides automated assurance of working and tested code and defined workflow for the code moving between environments.
Code Repository Software	Reduces the chances of configuration management issues particularly where multiple developers may be working on the same code, enables traceability from requirements to code level automatically.
Large Binary File Registry	Enables accurate configuration management and reduces complexity where multiple repositories are required.
Knowledge Management Software	A context repository utilizing taxonomy which assists team members in locating information quickly and accurately.

The tools used as part of Agile are covered within the scope of Appendix D9 and **do not** require computerized system validation, but should be subject to risk assessment, assessed for adequacy by appropriate SMEs, used by trained and qualified individuals, and appropriate controls applied according to the intended use, for example, to ensure that acceptance test results stored within a tool are trustworthy, complete, and available.

28.3.5 Approvals and Acceptance

Both the backlog refinement and sprint planning sessions are key governance and approval steps to ensure *we build the right system*, with the roles and responsibilities for approvals defined as part of the planning.

There is a misconception that approval means sign, and by extension that there is the need to comply with the life-sciences regulatory requirements on electronic signatures, for example US FDA 21 CFR Part 11 [73]. This is not the case and is only applied where the approval is equivalent to a traditional handwritten, legally binding signature required by a predicate regulation. For software life cycle deliverables, this is only the case for SaMD or software embedded in a regulated medical device. Approval can also be achieved by many other means such as status change, email, audit trails, and in many cases evidence of acceptance rather than an approval is sufficient.

Modern tool sets frequently have immutable change logs built into them. These logs record exactly *who*, did *what*, and *when* against each product backlog item, and can be configured to enforce security rights over exactly who can do what.

28.3.6 Software Development and IT Operations (DevOps)

DevOps is an extension of Agile where the development and maintenance activities are combined within a single team and set of practices so that focus can be placed on important/urgent tasks, silos in the team are avoided, and team members have a chance to develop through a broader set of activities.

In a DevOps environment, systems will undergo many changes/releases, and this initially seems to be difficult to reconcile in a GxP regulated environment. This concern can be mitigated with the level of control and oversight of the process. Also, there is no knowledge loss associated with transition from project to operation support teams, together with the tools and accountability of the DevOps team for the quality of the product. However, there is still a level of risk/impact assessment required for higher-risk GxP functions undergoing change.

Continuous Integration (CI) and Continuous Deployment (CD) (sometimes also referred to as Continuous Delivery) is an extension of DevOps where there is increased use/reliance on automated software analysis tools; code is delivered into code repositories and is verified (tested) and integrated using these enhanced tools.

DevOps teams provide CI and CD with faster turnaround times for increments (greater velocity).

DevOps provides for closer integration of the product and the associated toolsets that in turn provide additional controls such as:

- No development before an accepted user story
- Source code review and automated testing as part of the CI process to help avoid the release of bad code
- Ensuring close alignment and constant engagement with the product owner as feedback on proposals comes almost instantly

It is important that DevOps teams feel that they have accountability within the team for compliance; if compliance is perceived as something outside, then the team sticks together and stakeholders outside such as security, compliance, and auditors are perceived as strangers. [20]

28.4 Agile Approach to Quality

By following the agreed Agile process and associated ceremonies then a state of control can be demonstrated and quality built into the product.

When constructing epics and stories and their acceptance criteria, the same considerations for attributes of a good traditional linear set of requirements should be considered in terms of data, interfaces, environment, performance, availability, regulatory, maintenance, data migration, and security. Epics/user stories can be assessed against a set of DoR criteria before being accepted into a sprint backlog. For example, with Agile there are techniques that can be adopted to help check for good attributes of stories such as “INVEST” [74] where:

- **I** – Independent
- **N** – Negotiable
- **V** – Valuable
- **E** – Estimable
- **S** – Small
- **T** – Testable

A typical GxP regulated system has functional requirements that have GxP impact along with nonfunctional requirements, for example security, that need to be in place to support GxP regulations including data integrity. The regulated company should identify these based on the intended use and ensure that these are included within the MVP for the initial release of the system into productive use.

Where a third-party supplier develops a solution using Agile, take into consideration that the supplier may have business process and GxP expertise; therefore, the regulated company should work with the supplier if the areas with GxP impact have already been identified and leverage this.

Acceptance criteria are vital to help the product owner and team have a better understanding of the epics and user stories and should be captured within the epic and user story. To avoid duplicating the content of the actual tests, they should be kept at a high-level and succinct.

Demonstrating that a system functions according to its intended use is a basic GxP requirement, and largely demonstrated with linear models through layers of testing activities ultimately leading to a set of user acceptance tests. For Agile, testing is always performed within the sprint. There must be a “potentially shippable product at the end of each sprint;” an untested product is not shippable.

Agile testing uses a combination of exploratory testing and test automation with regression testing to verify that sprint developments do not impact correct functioning of software developed in previous sprints. Other test practices include solution walk-throughs or demonstrations where the potential solution is demonstrated to business users early. Any final acceptance testing should be minimized, and test activities and records are managed and stored within tools.

Agile is by its essence about tightly controlled management of change, and may seem to present a challenge in the context of GxP and maintaining a state of control. However, this is where Agile, when operated correctly, shows its strength; to deliver working software Agile processes need to be robust, well managed, and sprints under control with team accountability and with tools to provide additional control/oversight. Adherence to the defined Agile process (for example, Scrum ceremonies such as sprint planning, daily (Scrum) meetings, sprint reviews, and sprint retrospectives) drives quality, ensures timelines can be adjusted quickly, and provides a focus on learning by addressing issues as they appear. [20]

29 Appendix D9 – Software Tools

29.1 Introduction

This appendix describes the recommended risk-based approach and considerations when using tools supporting computerized systems life cycle processes, IT processes, and IT infrastructure processes. Such tools do not directly support GxP regulated business processes or GxP records and data directly supporting the regulated product life cycle.

The tools described in this appendix are GAMP software Category 1. Classification as GAMP software Category 3 – 5 applies only to business applications (either GxP or non-GxP). Tools, like systems supporting IT processes, and infrastructure are all GAMP software Category 1. Such tools should be managed through the application of good IT practices, within a defined IT management framework, such as that defined by ITIL [54].

As part of their technology and data modernization activities [75], US FDA have described their adoption of industry standards and good practices including ITIL [54] processes across all operating areas. They noted that increased automation, sharing, reuse, and consistent, repeatable processes are significantly improving the agility and cost effectiveness of IT activities.

The term software tool or simply tool in the context of information technology can be very broadly used, and this appendix seeks to identify tools and distinguish these from use cases where the software is actually a component of a computerized system supporting a GxP regulated process.

GAMP guidance and the FDA CDRH Case for Quality program [9] strongly encourage the use of software life cycle management tools and automation, which can bring great quality benefits, and have low GxP risk.

Such tools and systems supporting system life cycles, IT processes and infrastructure should not be subject to specific validation, but rather managed by routine company assessment and assurance practices and good IT practices. Compliance and regulatory resources should not be focused on audit and review of activities related to systems with little or no direct impact to patient safety or product quality.

EU GMP Annex 11 [32] requires that automated testing tools and test environments should have documented assessments for their adequacy. These should be supported by the application of appropriate controls following good IT practices.

29.1.1 Changes from GAMP 5 First Edition

This is a new appendix.

29.2 Scope

This appendix applies to software tools used as part of software development, support, and maintenance activities.

This appendix does not apply to software components/applications that are part of a GxP regulated business process, or can directly impact GxP data/records directly supporting the product life cycle; in this case risk-based computerized system validation is required to demonstrate adequate controls are in place. For example, software used to migrate or convert GxP records would require computerized system validation, however a tool used to mask or obfuscate data to be used as system test data would not.

Test tools are another commonly used toolset; they perform an important role by helping to assure thorough testing, often automated and including regression and boundary/stress test features. Such tools also often store the associated test results but even though these records may support the computerized system life cycle, the test tools themselves do not require validation. The records they maintain should be preserved and protected, and the tools need to work reliably, but this can be achieved through a basic risk assessment and application of normal IT practices.

System tools used for monitoring/performance management of systems or tools associated with IT security, do not require computerized system validation, but a risk assessment approach and basic IT good software support practices such as defined by ITIL [54], including having an inventory of the tools and where they are deployed.

29.3 Guidance

29.3.1 Selection of Tools

Some basic assessments should be performed as part of tool selection, this does not need to be onerous, but the effort may vary depending on the intended use and risk.

The key criteria are whether the tool is the appropriate tool to support the life cycle process we are trying to automate/support: does it meet the need of the SMEs who are going to use the tool and do the work, and is it fit for purpose?

Note that it is not envisaged that any tool would need to have anything other than a desktop assessment, unless the use was considered business critical.

Where tools are to be shared by a supplier and the regulated company then consideration is required around access to data/records over the operational lifetime of the system, and any handover of the tools and data once the project phase has completed.

29.3.2 Risk Assessment

Some of the factors to consider as part of risk assessment are:

- Is the tool an industry standard one in common use?
- What data/records are generated/stored by the tool, and does the tool have adequate controls to maintain data integrity?
- What is the required operational lifetime of the tool, and how will this be supported?
- Is the tool open source, are the download locations verified to avoid cybersecurity issues?
- Will the tool require any level of workflow/configuration, including any key activities that need to be manually performed?
- For automated tools, for example, when considering performance monitoring, how are any alerts monitored and acted upon?
- If the tool fails, could it impact an operational computerized system or infrastructure operation?
- Could the tool have any potential negative impact on system or network performance?
- Could the tool introduce any cybersecurity risks, e.g., facilitate the introduction of malware?

29.3.3 Inventory/Deployment

As part of good practice, it is sensible to have some form of inventory, for example, within a Configuration Management Database (CMDB), of what tools are used by the organization, where they are deployed, and the nature of any data/records potentially supporting GxP (e.g., test records) and required retention period. Identifying a single point of contact/owner for the tools is also recommended.

29.3.4 Installation and Configuration

Tools, even when off-the-shelf, often will require some degree of configuration or parameterization in order to correctly install and establish any workflows and associated aspects such as (project) naming conventions or access levels, etc. These are basic good software engineering practices that need to be applied along with version and appropriate change control.

29.3.5 Life Cycle Management

Tools, as with all software, typically have a defined lifetime, upgrades, and eventually are retired and/or decommissioned.

One person or role should be designated to be the owner of the tool to provide the required level of oversight; the oversight effort will depend on the level of criticality, scope of use of the tool, and complexity of the tool.

In general tools, that have the ability to impact system or network performance and hence impact users require more oversight than those used as part of software development activities.

It is best practice to ensure tools in use are at versions supported by the supplier.

29.3.6 Security Considerations

Cybersecurity considerations should be included for tools. Tools that are deployed across the network for example, for performance monitoring, may require more detailed security analysis as the consequences could be severe if a vulnerability were to be exploited.

Security considerations should be included as part of the initial risk assessment for the tool and also as part of the life cycle management for example, ensuring tools are maintained as patched and up-to-date/supported releases and also only downloaded from authorized sources. Access privileges for tools should be set as low as possible to allow the tool to perform the task.

29.3.7 Data and Records

Where tools are storing information/records that support the validated status of the system, for example contain system requirements, design, code or test results, then controls for maintaining the integrity, and in particular the security and availability, should be determined and applied. Note also that increasingly tools, and the records/data within them may be cloud based and therefore assuring data integrity and access throughout the retention period should take account of this.

Note that the full set of data integrity controls that would be applied to primary GxP records are not required as these tools, by definition, do not maintain GxP regulated records; most tools do support technical and transactional logging supporting integrity and investigations so for example, audit trail event analysis may be useful in the event of fault finding.

29.3.8 Approvals

For tools within the scope of this appendix (i.e., not directly supporting a GxP regulated business process or maintaining required GxP records) any approvals or acceptances of transactions or results associated with the tool are not subject to regulatory predicate rules and therefore are not subject to regulated electronic signature requirements.

Where an approval or acceptance is required, there needs to be traceability to the person approving or accepting, and also approval roles need to be defined. Many tools will have this traceability and functionality built in as standard.

29.3.9 Tools over Documentation

There may be a temptation, particularly where tools store test execution records for example, to export information into controlled GxP documentation when there is no quality benefit in doing so. This approach is not recommended, it is inefficient and there is the risk that documentation becomes out of step with the information/records within the tools. Tools should have sufficient controls to preserve the data/records (see Section 29.3.6).

Where tools have reporting functionality, this can be useful for producing brief summary reports to provide baseline status, for example, a summary report of tests executed, passed and failed would be included in a validation or test report for the system being tested by the tool.

There is the potential for regulatory inspections of computerized systems to request access to information in tools, for example, system requirements test results. Process owners need to be aware of what information relating to their computerized system life cycle activities is managed within tools, and regulated companies and IT departments need to be prepared to show the information if requested. There may be value in determining in advance how this can be readily achieved, for example it may be possible to pre-configure reports to extract test results relating to areas of GxP functionality.

Tools and systems supporting life cycles, IT processes, and infrastructure (rather than directly supporting product life cycle processes) are not themselves GxP regulated systems and should not be subject to specific validation but managed by routine company assessment and assurance practices and good IT practices. The activities performed and the records maintained for these supporting tools should not be viewed as requiring the same rigor with respect to controls as systems that directly support GxP product life cycle activities. [76]

30 Appendix D10 – Distributed Ledger Systems (Blockchain)

30.1 Introduction

Distributed computing using a decentralized infrastructure is rapidly advancing, specifically blockchain and other distributed- ledger technologies that leverage a combination of cryptography, consensus, smart contracts, and replication to capture and secure data from multiple participants within the network.

Blockchain technology has many use cases that help to establish trust and transparency among parties and eliminate silos by providing an end-to-end solution such as track and trace within the supply chain.

As more members of the healthcare technology environment choose to participate in the network it may span entire supply chains from starting material manufacturers to pharmacies. The network effects of this technology may put pressure on pharmaceutical manufacturers to participate for both regulatory and business purposes. Even if the regulated company chooses not to participate, it is very likely that one of their suppliers or customers will.

Given the unique and emerging nature of this technology, this guidance is intended to provide considerations for organizations to apply when relying on blockchains to support GxP processes. This guidance is not intended to define the technology or discuss the appropriateness of the technology to meet business needs.

30.1.1 GAMP 5 Context and Application

GAMP 5 sets out a risk-based approach that allows an organization to conclude that a computerized system is compliant and fit for intended use. If the entire blockchain is treated as one system, it may be difficult to identify the intended use; most blockchains by nature are designed to support many use cases. Therefore, applying critical thinking and modern development methodologies, e.g., Agile, are essential to maintain control over the qualified state of the blockchain network.

Based on the intended use, risk to patient safety, product quality, and data integrity, as well as the degree of novelty, complexity, and configuration or customization, a framework and regulated data life cycle model can be developed that encompasses traceability and considerations of use.

It is important to understand the intended use of the application and its components that leverage unique features of blockchains so that a risk-based approach can be applied. For example, if a blockchain is considered to be a network layer that also has some functions of a database and an operating system, it falls into the classic categorization of infrastructure, which tends to carry a lower risk and shares many attributes with a virtual-computing environment. Some blockchains have the ability to leverage smart contracts, which operate like small computer programs on the network; these could be considered customized software (if built for a specific purpose) or configured software (if built from a recognized standard). When smart contracts begin interacting with other smart contracts, blockchains become capable of complex business logic. As this can be scenario dependent, it is important to consider the business process, how the data flows, and the criticality of the decisions made using the information derived from a blockchain network.

Most blockchains are not intended to be large data stores, but the data contained wherein or the log of transactions captured may be critical to the organization's application. In these cases, the concepts laid out in the ISPE GAMP Guide: Records and Data Integrity [35] apply, particularly the concepts of data retention and retrieval, as blockchains are designed to be permanent records and highly available.

30.1.2 Changes from GAMP 5 First Edition

This is a new appendix.

30.2 Scope

30.2.1 Objectives

This guidance is written from the perspective of an IT quality professional whose primary concern is ensuring that the usage of computerized systems does not introduce new risks to patient safety, product quality, and data integrity. It is assumed that the regulated company has already determined blockchain to be an appropriate solution and selected a use case and a blockchain protocol to work with.

This guidance is not intended to influence the design of blockchain networks or applications, but to educate the GxP practitioner in the things to consider when relying on these systems. At the time of production, blockchain technology is evolving and not widely implemented in regulated life science environments. Whereas most GAMP guidance stems from lessons learned, pragmatic experience, and response to regulatory findings, these blockchain considerations are based on proactive insights gained from early proof of concepts and challenges faced in other industries.

As blockchain involves interactions among multiple parties such as regulated companies and suppliers working together on the development of the system, an Agile methodology and critical thinking can be applied to support an iterative and exploratory approach for planning, testing, and continual maintenance.

This appendix focuses on large-scale public blockchain implementations that are sufficiently decentralized (for example, not controlled by a small group of entities) and potentially used for multiple use cases, both GxP and non-GxP. Small-scale private blockchains can be configured in many ways, but they will either be a subset of public blockchains or more akin to a shared proprietary system or a shared computing environment, which is covered in existing GAMP guidance documents (for example, cloud computing).

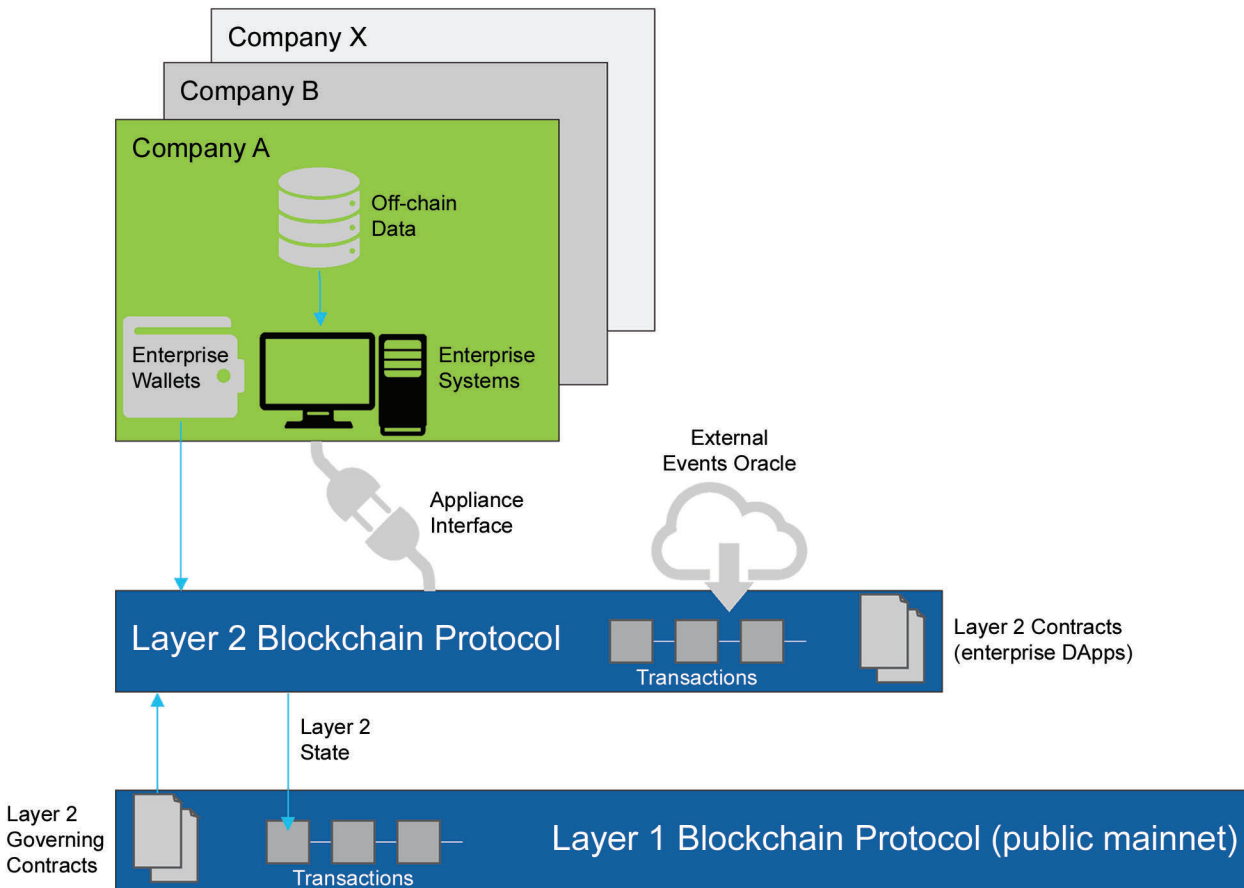
Addressing public blockchains also addresses the open-source nature of their development and governance, considerations around the balance of privacy and transparency, the distribution of external stakeholders responsible for managing the network, the incentives for participants to behave as “good actors,” and ancillary systems that are necessarily a part of the blockchain ecosystem. These include existing systems that may or may not be regulated, the interfaces to those systems (often referenced as Application Programming Interfaces (APIs)), traditional databases and/or distributed storage, and Layer 2 solutions that run on top of a primary (Layer 1) blockchain.

A Layer 1 network refers to blockchain, while a Layer 2 protocol is a third-party integration, secondary framework, or protocol that can be used in conjunction with a Layer 1 blockchain.

30.2.2 Governance of Large-Scale Decentralized Networks

With public networks there is often the need for Layer 2 solutions that provide for more efficient transactions, privacy, and other use-case-specific requirements. A Layer 2 solution can also be an open public network, in which case it is similar to relying on a Layer 1 public blockchain. Or the Layer 2 solution can be private/permissioned networks, which can be treated as shared computing environments with proper change controls and governance, similar to the way an organization would rely upon a cloud-computing provider, but with the added benefit of security and decentralization provided by the underlying Layer 1 solution.

Figure 30.1: The Blockchain Technology Stack Across Multiple Enterprises



When a network is sufficiently decentralized there are enough independent network operators to make it improbable that a single entity or a small group of entities hold influence over the core protocol (the primary rules) that operate the network. When changes are needed, it requires coordination of the majority of network operators. This is a familiar concept for those that rely on OSS. Generally speaking, core protocol updates positively impact the operations of the network (e.g., more efficient processing, network operator incentives, better encryption).

Governance also applies in how an entity and its business partners choose to interact with the blockchain. While some of this is controlled through the smart contracts the entity connects with, much of it has to do with the off-chain data standards and processes that trigger events on the blockchain, as well as the API through which they connect.

There is no preventive mechanism to stop an entity from posting inaccurate data or triggering fake events. However, because blockchains are ecosystem-level systems, collusion with other parties would be needed to continually propagate false information, which is unlikely. In addition, there would always be a clear log of transactions. To this end it is important to know exactly who the business partners are (that is, the addresses they control on the blockchain) and to recognize that the system is more secure when looking at the sum of the parts. Because it is most likely that enterprises will interact with blockchains through APIs, GxP considerations must be applied to the blockchain network as well as the ancillary systems that push and pull information to it.

30.3 Guidance

This appendix outlines the current best practices that should be used during the system life cycle when deploying a blockchain-based system in a production GxP use case. Once the intended use of the system is established, system planning includes a holistic approach for development and deployment. Although there is not a “one size fits all” approach for documentation requirements to qualify and maintain a blockchain network, applying methodologies such as Agile and critical thinking can help identify appropriate controls needed and mitigate the associated risks within the regulated environment per Appendix D8 and *ISPE GAMP RDI Good Practice Guide: Data Integrity by Design* [36].

30.3.1 Concept Phase

Blockchains are often considered to be an ecosystem-level network. From the perspective of an individual organization there are likely to be external stakeholders that should be involved in all phases of the project, including the concept phase. Understanding the incentives of each stakeholder as well as their role in the GxP use case is critical to developing a solution that can meet a broad set of objectives.

During the concept phase of the project, it is important to apply critical thinking and a risk-based approach to determine exactly which use cases within the blockchain solution are GxP and what ancillary systems and data elements are required by the blockchain solution. Business-process mapping can help to define those interactions by documenting:

- The intended use of the blockchain solution
- The ancillary systems that the blockchain will pull data from and push data to
- The high-level risks associated with patient safety, product quality, and data integrity
- The business logic managed within smart contracts (if applicable)
- The role blockchain plays in the overarching system
- The data being managed across the network (on-chain, off-chain, metadata)
- The type of blockchain protocol and any special considerations

The sequence of process steps, high-level risks, and business activities can be addressed as part of mapping the business process. It illustrates the manual and computerized aspects of the system and associated risks, such as manual operations, access controls, and interfaces. Business-process steps enable the IT quality professional to understand data integrity risks per the regulated data life cycle and assess and evaluate where data is generated, processed, used, retained, and removed. Business process mapping can support data quality by facilitating the assessment of data criticality and vulnerability, and identification of system interfaces.

Once high-level risks and business-process steps are understood, data flow can be generated that show the inputs, outputs, and flow of data through the system. Similar to understanding the integration points for an ERP system or Manufacturing Execution System (MES), it is helpful to create a swim-lane diagram for the systems that are part of the overall solution. The individual components and systems may have to be treated differently with different states of control. Depending on the implementation, the blockchain may be a single swim lane acting as a data repository, or multiple swim lanes capturing the functionality and interactions of smart contracts.

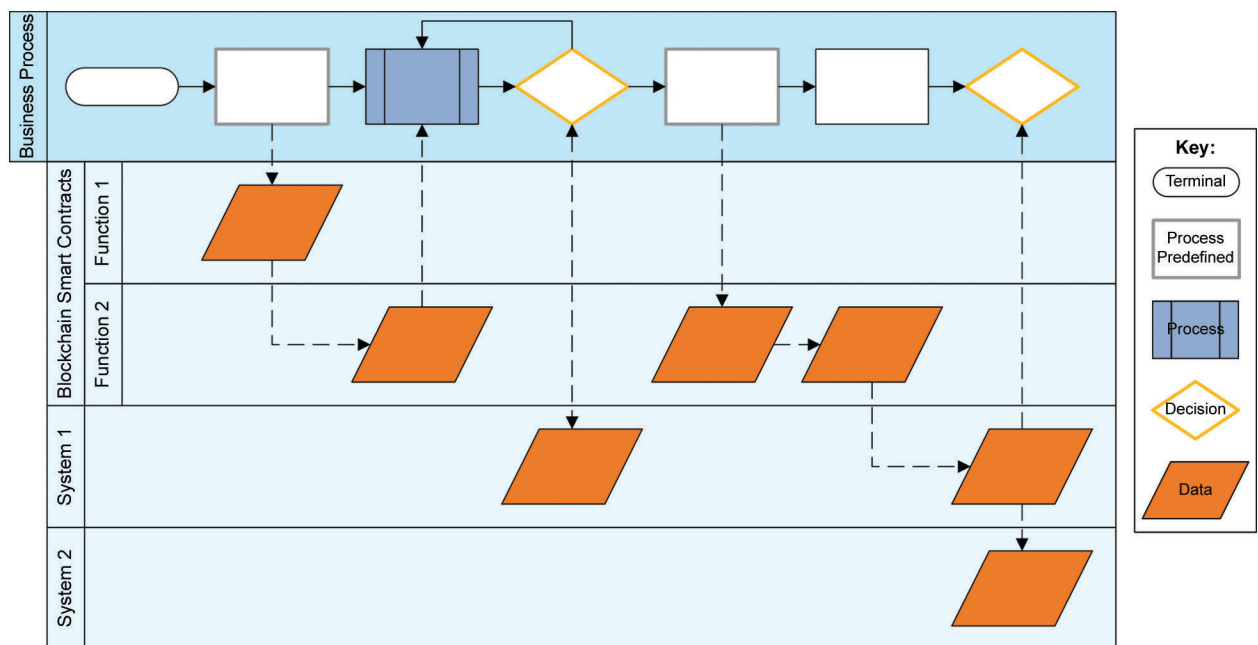
When deploying and validating a blockchain solution, it is recommended to take a strong systems-engineering approach, looking at where the various technology systems interact and where those interactions are based on APIs, or smart contracts, or potentially other technologies.

Data flow can be used to map interactions for an overview of the system landscape and to identify areas of high risk. Utilizing this approach can provide an integrated-risk overview, whereas a fragmented system-by-system approach may miss key integrations and interface points of contact between the various systems. An example data flow map is displayed in Figure 30.2.

Smart contracts are used within blockchain networks to automate actions based on defined quantitative variables that the contract monitors. The contract then executes actions when the defined criteria are met. An analogy is the familiar logic trees within PLC programming.

Smart contracts should be assessed as part of the overall-system life cycle of the blockchain network or application. As they utilize custom coding, standard code-review processes should be used to review the integrity, security, and integration of the inputs and outputs of the smart-contract logic.

Figure 30.2: Example Data Flow for a Blockchain-Integrated Solution



30.3.2 Project Phase

30.3.2.1 System Planning, Stakeholders, and Service Providers

The blockchain network is likely to have many stakeholders, both internal and external. A typical stakeholder/steering committee approach may be difficult to implement across an entire ecosystem, therefore establishing a governance structure early in the project phase is recommended. (Refer to Section 30.3.2.6 for additional considerations.)

There are often third parties involved in developing or hosting the blockchain solution. In some cases, a Blockchain as a Service (BaaS) provider will be selected to manage the integration to, or run nodes on behalf of, the organization. In these cases, supplier assessments and testing should be leveraged in accordance with Appendix M2. However, the blockchain-specific elements should also be considered, such as:

- Is there a process for identifying, adding, and/or removing a node from the network?
- Is the BaaS also providing off-chain storage and/or transaction caching? If so, what are the data controls around those services?

- What is the process for managing blockchain addresses? How is ownership and custody over the private and public keys managed?

30.3.2.2 Data and Process Driven Approach

Ultimately blockchains are just systems that humans or other systems use to push and pull information in support of business processes; therefore, a data and process-driven approach is a suitable method for relying on a blockchain network in a GxP environment during the project phase. A suggested method for validating the environment is outlined in the following sections.

30.3.2.3 Understand Data from Source of Origin, Source of Truth, and Ownership Standpoint

The blockchain will likely serve as a backbone connecting multiple sources of data, and in many cases becoming the source of truth for which data represents the current state of the overall system. (This trust and shared ownership of data is often a primary reason for using a blockchain.) However, blockchains are not generally the source of origin for new data other than identifiers (for example, token IDs), accounts, and timestamps. In these cases, data-quality controls should be in place to keep the data recorded in the blockchain in sync with data generated in the source of origin. Data mapping and checks of ALCOA+ requirements can help to identify deficiencies, for example, potential problems associated with the lack of long-term access to data. The blockchain may provide evidence of which account(s) signed a transaction on the network, but the organization may need to understand which other organization(s) control that account, implying the need for a registry or some form of verified credentials.

Illustrative example:

A serialized medicine has the following data attributes:

```
{ "GTIN (01)": "00855245005019",  
  "EXP (17)": "112023",  
  "LOT (10)": "XYZ123",  
  "S/N (21)": "447018182632" }
```

A token is generated on a blockchain that represents this single unit; for privacy purposes it is given a random token ID:

```
0x778gha0ca303305a92d8d028704d65e4942b7ccc9a999164cf
```

Included in the transaction that minted the token is a Uniform Resource Identifier (tokenURI) that contains a hash of the data attributes and a pointer to where the metadata is stored (in some storage systems these may be one and the same). For example:

```
1220D5CC15B22D5606BF3DC255979552A4214E29054CD2E60A1446FE570F53F18C6B
```

(this hash was generated from the data above by using a SHA-256 hash algorithm in a multihash format).

In this example an investigator can access the un-hashed metadata attributes by following the tokenURI. They can pass the metadata through the same hash algorithm (this is why using a multihash format is important) and if the hashes match, they have confidence that the un-hashed data was the same as the data used to mint the token. If the investigator has access to the source system where the product data was generated, they can query the source system and verify the individual attributes.

30.3.2.4 Identifying the Resulting Controls

After identifying the critical functionality by performing a risk assessment of the business-process logic and the data flows, define a set of controls that will help to ensure the system is operating as intended. Some items to consider when defining control sets are:

- Functionality
 - Input Controls: Events used to trigger transactions and the data posted to the blockchain during a transaction are typically managed outside of the blockchain itself. However, once a transaction is posted it is practically immutable; therefore the integrity of the data initially posted should be considered.
 - > How are events that trigger blockchain transactions identified in each of the systems that feed data to the blockchain?
 - > For data that is captured either on-chain or off-chain, are sufficient data quality controls in place?
 - > If off-chain data changes for legitimate reasons (for example error corrections, updates, etc.), how are on-chain references updated as hash reference may no longer be valid?
 - > Is there any transformation happening through the API when data is pushed to the blockchain?
 - Output Controls: Extracting data from a blockchain can be done in several ways, but often the on-chain data is only a portion of the information needed to make business decisions. The emphasis on output controls is to demonstrate the accuracy of the information on-chain to the off-chain data through reconciling and displaying the completeness of the extracted data by making the query replicable.
 - > Is there a mechanism for checking the integrity of off-chain data (for example, hash matching)?
 - > Is there any transformation happening through the API when data is queried and extracted?
 - > Are there controls in place to ensure that the representation of the data (for example, data visualization software) shows an accurate and complete picture of the data available on the blockchain?
 - > Which specific data elements are relied upon from the blockchain layer (e.g., timestamps, account IDs, transaction IDs) versus data and events used to generate the transaction?
 - > How is the returned data analyzed to determine if transactions were skipped or omitted?
 - Processing Controls: Ensuring that the blockchain solution is continuously used as intended, or if there are changes to the network that introduce new risks, a set of processing and monitoring controls is required.
 - > How are the APIs that integrate systems with the blockchain monitored? Can events recorded on the blockchain be reconciled back to source systems?
 - > How is change control managed for the underlying blockchain protocol? (see Section 30.3.2.5)
 - > How are failed transactions monitored and addressed?
 - > If applicable, how are transaction costs managed? Is there a risk that the posting accounts will have insufficient funds to cover the fees?
 - > How is the stability of the blockchain monitored to determine if it is still viable?

- Access Controls: In a blockchain the participants are identified by their blockchain address. Addresses (sometimes called wallets) can be owned by individuals, entities, systems, or smart contracts; and it is the address owner who is responsible for posting transactions under that address. Access controls on a blockchain revolve around who has access to use which addresses, and what a given address is allowed to do.

Operating under the assumption that the blockchain will be used for many different use cases, the addresses involved in the GxP application will likely be a subset of all active addresses. The IT quality organization should be involved in the design of access controls between users, ancillary systems, and the blockchain network. The function responsible for routine administration of access should notify and discuss any major changes to the level of access that might impact GxP applications with the ITQ organization. Additional points to consider include:

- > Under what authority are addresses generated and managed?
- > Are the addresses attributable to an entity within the organization, an individual, or a source system?
- > How are users (or underlying systems) authenticated before posting transactions?
- > Is there a process in place for managing custody to the private keys of the transacting addresses?
- > If off-chain data bases are separate from existing systems, have they been implemented following the organization's security procedures?
- > If privacy solutions (e.g., zero knowledge proofs, roll-ups, mixer contracts, etc.) are used to obscure transactions on-chain, who has access to view the transactions and how is that managed?
- > If allow/deny listing (or other similar preregistration approach) is used to grant access to call functions in a smart contract, how is this being managed?
- > Are the ancillary systems and their APIs that access the blockchain secure in preventing unauthorized transactions to be posted?

30.3.2.5 Change Management

The change management of the blockchain solution should be looked at for the entirety of the GxP use case. It is likely that many of the source systems, off-chain storage, and APIs used to interact with the blockchain network can be managed using traditional Agile change-management approaches. For public networks, it is likely that the core protocol used to propagate the blockchain network is open sourced and managed by a community. Refer to Appendix D8 for guidance on relying on OSS.

30.3.2.6 GxP Considerations around Governance of Public Blockchain Networks

Specifically for blockchains, there are additional considerations above those of general OSS. Namely how the governance of the running network is managed by the node operators.

Occasionally there will be significant changes to the core protocol in the network. For those changes to be adopted, the majority of the node operators must update their software¹⁷. Most often the updates are done seamlessly with no disruption to the network; however, there is a possibility of some nodes choosing not to update their software. In these cases the network splits (or forks) into two separate blockchains. While this is less likely with improved governance and becomes even more difficult with larger more decentralized networks, it is still a possibility that should be considered. As is typical with managing reliance on OSS, it is important for organizations to keep abreast with updates and changes and determine the GxP impact of updating (or choosing not to update) their applications.

¹⁷ For an organization running a node, this would be an action the IT function would take.

30.3.2.7 Composability of Smart Contracts

Often several smart contracts are strung together to create a Decentralized Application (or DApp). When considering the governance of large-scale decentralized networks, it is governance of the DApps that carry a higher risk. These contracts are often deployed with the intent of being composable, which means that the functions within them can be called by other smart contracts. While this is key to enabling complex logic, it also creates a network of dependencies that must be understood.

By changing references when certain functions become deprecated, a smart contract can be upgradable, which implies the need for change-management controls. Because the nature of blockchain within enterprises is such to encourage broad ecosystems, these contracts are often designed for a broad range of participants. They also have the capability to ingest data from external sources (referred to as oracles) to trigger certain events within the smart contracts code.

For example, if a public blockchain was being used to track pharmaceutical supplies with the intention of facilitating product recalls or managing stockpile levels, there would likely be several smart contracts involved; for example, one that creates the product on-chain and one that auto-orders replenishment stock. There could be one that listens for an external signal from a manufacturer or regulator that might flag inventory that has been recalled. Assuming that the underlying blockchain protocol remains stable throughout this process, the greater risk is in the way that the smart contracts interact or vulnerabilities that those interactions might introduce.

It must also be possible to differentiate between smart contracts that are core to basic use case operations (likely leveraging standards with very little modification) and smart contracts that have been custom developed for a specific DApp. The former can be assessed as part of the overall reliance on the blockchain network; the latter would require specific considerations within the scope of the DApp.

30.3.3 Retirement or Migration

Most blockchains are designed to be highly available in perpetuity; however, it is possible that networks may become stagnant and be abandoned by their participants. As the level of decentralization decreases, many of the strengths of relying on a blockchain solution may be eroded. Considerations should be made for how to retain data in the event that the blockchain network is retired, in particular where those records are subject to regulatory data-retention or legal-hold controls. This may be resolved by requesting the organization's or service provider's IT function to maintain a full node of the network and anchoring the last block as data in another secure location (for example, another blockchain) to demonstrate data integrity until the organization is ready to retire the solution.

Further, real-life use cases of blockchain to support regulated processes have not been around long enough to fully assess the implications on long-lived data; it is suggested that organizations take a "cryptographic agility" approach allowing them to evolve as the technology matures. Considerations include:

- The impact of replacing cryptographic methods (the underlying security guarantees are not automatically preserved)
- The impact of storing sensitive data (either encrypted or unencrypted) on the network as future vulnerabilities may be found in the current state-of-the-art cryptography
- The ability to maintain and extract data from a blockchain node even if that blockchain is no longer active

It is also possible that the organization may choose to migrate to a different blockchain protocol, in which case many of the existing procedures and strategies for records migration would apply (refer to *ISPE GAMP Guide: Records and Data Integrity* [35]). However, there would be several blockchain-specific considerations:

- The new network is likely already running; therefore, historical records may not be recreated in a new network with the original transaction information and ordering

- Blockchain addresses are inherent to the protocol that generated them; it is not likely that wallet addresses and contract addresses will be identical. This will require mapping access controls into the new environment.
- Much of the blockchain security that is rooted in cryptography relies on the network running continuously; migrating to a new network will impair that security for historical transactions

30.4 Other Considerations

Additional considerations may need to be taken when using blockchain networks. These fall outside the direct software-development life cycle but form a critical component of evaluating and demonstrating that a blockchain-based system is under control and suitable for use in a GxP regulated environment.

30.4.1 Use of Emerging Technologies

When choosing to use an emerging technology that may not have reached a state of maturity, it is important that a practitioner takes steps to ensure that a strong evaluation of the technology is performed prior to selection and that the often rapid pace of architectural change is accounted for in the operation and use.

Key considerations that should be accounted for may include:

- Maturity of the blockchain:
 - Is the blockchain supported by a large developer community?
 - How long has the blockchain been in operation?
 - Has the blockchain been subject to multiple forks?
- Vendor support :
 - Does the blockchain have a commercial support model?
 - Is there a landscape of technology vendors supporting the blockchain?
 - Does the vendor operate a QMS and/or have software-development and management-process controls in place?
- Pace of change:
 - Does the blockchain have a mature change-governance process?
 - Are architectural changes well communicated and documented to users?

30.4.1.1 Use of Standards

Standards are still emerging in this area and are largely driven by the development communities as opposed to regulators as of this publication. When looking at a blockchain from the perspective of an entire ecosystem, standards become important to enable interoperability. These will most likely be data standards relevant to all members of the ecosystem and should be built from existing standards sets.

Standards are also important from a security perspective. Smart contracts deployed on a network execute business logic when called upon. Similar to any technology, there are multiple attack vectors by which those smart contracts can be exploited to cause unintended outputs. Where possible, it is recommended to use standard deployments of smart contracts that have been in use on public networks for some period of time where they may have been hardened from previous attacks. It is also recommended to utilize third-party specialists to review smart-contract logic prior to deployment and/or to deploy contracts in a public testnet environment prior to launching in production.

30.4.2 Reliance on Cryptography and Proofs in Place of Trusted Systems

Using the cryptographic principles of hashing and asymmetric encryption, blockchains generate practically immutable storage for time-stamped data, which can be attributed to and accessed only by authorized participants. The level of trust (or assurance) depends on:

- Linking of blocks based on hash functions that are both easy to verify and computationally intensive to create
- Asymmetric encryption where knowledge of the public key allows:
 - Storing information that only an authorized recipient can decode using the (secret) private key
 - Verification of signatures generated using the sender's private key

Much as the use of duplication and error correction in RAID storage eliminates the need for high-level reliability on hardware and environment conditions (e.g., vibration free), blockchain technologies replace the need for a qualified hardware (storage) from a mature supplier by mathematically sound algorithms.

From this perspective the validation strategy for the source system may be focused more on data quality than data integrity. It should also be noted that while an in-depth understanding of the mathematics and cryptography may not be required, organizations should check that generally accepted or well-known algorithms are used for the process (similar to the way one might inquire about the security standards used for data encryption).

30.4.3 Role of Cryptocurrencies and Incentives on Public Networks as They Relate to GxP Applications

In public blockchain networks the inclusion of new data in the blockchain is done collaboratively by all participating nodes in the network. Their incentive to process specific transactions on behalf of others is the transaction fee voluntarily donated to the participant(s) attaching the respective block to the blockchain. Consequently, to post a transaction on the blockchain, some value (in the form of the chain's inherent currency) must be transferred at the time of record creation. Transaction fees (sometimes referred to as gas) are also required to deploy and interact with smart contracts on the network (both of these activities are transactions). The fees are typically paid by the address initiating the transaction, implying that the address may hold some value in addition to its access rights, which could create additional risks for those addresses to be compromised. The following should be considered when relying on public blockchain networks that have transaction fees:

- Fees are typically in correlation to the amount of data in a transaction. In addition to incentivizing network node operators, they also incentivize efficient use of the network. Adding additional data to a transaction "just because" can quickly make blockchains cost prohibitive.
- To increase throughput and reduce transaction fees Layer 2 solutions are often deployed; these can be considered as separate networks that inherit some of the security of the Layer 1 network, but they should be carefully assessed.
- The value of the cryptocurrency held by the addresses (both individually and cumulatively across an organization) can fluctuate wildly, and there may be accounting and tax implications.

30.4.4 Achieving Privacy and Confidentiality on Public Networks

A fundamental design principle of blockchains is the distributed ledger, that is, the ability for all participants to quickly come to agreement on the current state of the entire dataset. Early blockchain networks achieved this by having a fully-open and transparent ledger that is visible to all participants. This approach is often diametrically opposed to a particular organization's desire to have private transactions between known business partners. Even if sensitive data is not included within a transaction, the pattern of transactions between accounts may be considered confidential. This visibility also extends to the logic of smart contracts operating on the network.

Privacy and confidentiality is a rapidly evolving facet of the technology, as it has been widely recognized as a barrier for enterprise adoption of public blockchains. There are multiple technology solutions that leverage cryptographic measures (zero knowledge proofs, optimistic roll-ups, etc.) and scaling solutions (Layer-2 scalability, Layer-1 alternatives, sharding, etc.) that are required to provide the on-chain compute power to support these methods.

Though the technology is evolving there are fundamental elements to consider:

- Avoiding the storage of sensitive data on-chain
- Encryption and obfuscation controls for on-chain data
- The maturity of the cryptographic procedures being used
- Processes for cryptographic key management
- Mechanisms used to anonymize transactions or shield the logic in smart contracts

These areas are actively discussed in the ISPE GAMP Community of Practice [77].

31 Appendix D11 – Artificial Intelligence and Machine Learning (AI/ML)

31.1 Introduction

Artificial Intelligence (AI) and Machine Learning (ML) are transforming the way in which the life sciences industry is doing business and processing data. The industry is increasingly relying on such innovative technologies to automate many functions previously performed by humans. As computer systems become more integrated and datasets become more extensive, computer science is advancing our ability to learn from that data and draw conclusions. Underlying algorithms are sophisticated enough to begin making robust decisions in the form of AI.

The use of such AI, along with the subdiscipline of ML, presents the life sciences industry with a challenge in maintaining the overall quality and regulatory compliance of such IT systems, applications, and/or solutions. This appendix provides a basic understanding for these digital solutions and guidance on how to ensure compliant integration and fitness for use in a validated environment. Additional information is available in *ISPE GAMP® RDI Good Practice Guide: Data Integrity by Design*, Appendix S1 – Artificial Intelligence: Machine Learning [36].

Additionally, this appendix:

- Provides a basic understanding of AI and the use of static/dynamic ML sub-systems in industry
- Presents an overview of a risk-based regulatory compliant AI/ML life cycle framework (e.g., model) in alignment with GAMP 5 principles and phases (concept, project, operation, retirement)
- Describes the importance of data integrity to the overall quality of AI/ML in addition to presenting an understanding of inherent risks
- Acknowledges the iterative nature inherent in developing AI/ML as a sub-system of the overarching IT application and/or business solution

31.1.1 Changes from GAMP 5 First Edition

This is a new appendix.

31.2 Scope

This appendix seeks to describe a life cycle framework for ML for use within the life sciences industry and a GxP regulated environment. It positions and contextualizes the life cycle and management of the ML sub-system or components within a wider GAMP system life cycle, at the level of general descriptions and guidance.

This appendix applies specifically to the ML component or sub-system, which is typically embedded within a wider IT system, solution, or application.

31.3 Guidance

There are many similarities in best practice between ML and more traditional algorithmic programming. Successful implementation of ML requires thorough business analysis and process understanding (e.g., by data scientists), effective planning, and the application of good software development, engineering, and maintenance practices. (See Guiding Principles 1 and 2 in Joint US FDA/Health Canada/UK MHRA Good Machine Learning Practice for Medical Device Development: Guiding Principles, [78].) Where this starts to diverge from previous strategies is that data management must be implemented during the concept phase because the data is the key resource required to progress with ML.

Performance metrics are important in the design of any ML sub-system. They act as the technical specifications for the acceptability of the ML model. These metrics also drive the inherent iterative training, evaluation, and improvement stages of a ML operation. As the process may have been previously performed by a human, the performance expectations and grading may shift, and it is important to understand this shift early on by asking questions such as:

- Will existing Key Performance Indicators (KPIs) be used to grade against the change?
 - If so, how would they be impacted? What change would show success? For example, is quality expected to remain at the same level or actually increase?
 - If not, how are the new metrics determined and what insight do they provide to confirm the system is operating as intended?

Another key aspect of ML development is the tight integration of data and metadata into the development process. The term data-centric development is sometimes used to reflect this. As a result, data should be managed with the same care as the code itself with the consequence of additional overheads and controls for data acquisition, selection, classification, cleansing, and augmentation.

Like other software system development, ML development has business, technical, or project risks commensurate with the complexity and novelty of the system. Managing these risks requires good process/business analysis, risk analysis, and cost/benefit analysis at all stages of development to recognize issues and to decide whether to take mitigating or rectifying steps, or to terminate the project.

Development also requires consideration of human factors (e.g., usability challenges such as alert fatigue), cybersecurity, and legal liability. This requires transparency, and an understanding of the ability to reproduce outcomes, adequately interpret the results, and understand the relevance for how the models will be applied without bias.

An in-depth knowledge of the business process and the associated data flows and decisions are essential for successful implementation of AI/ML. Such an understanding of the business process will also enable critical thinking to be applied on where and how to scale quality and compliance activities, based on identifying and understanding the potential risks to patient safety, product quality, and data integrity. (See Guiding Principle 9 in Joint US FDA/Health Canada/UK MHRA Good Machine Learning Practice for Medical Device Development: Guiding Principles [78].)

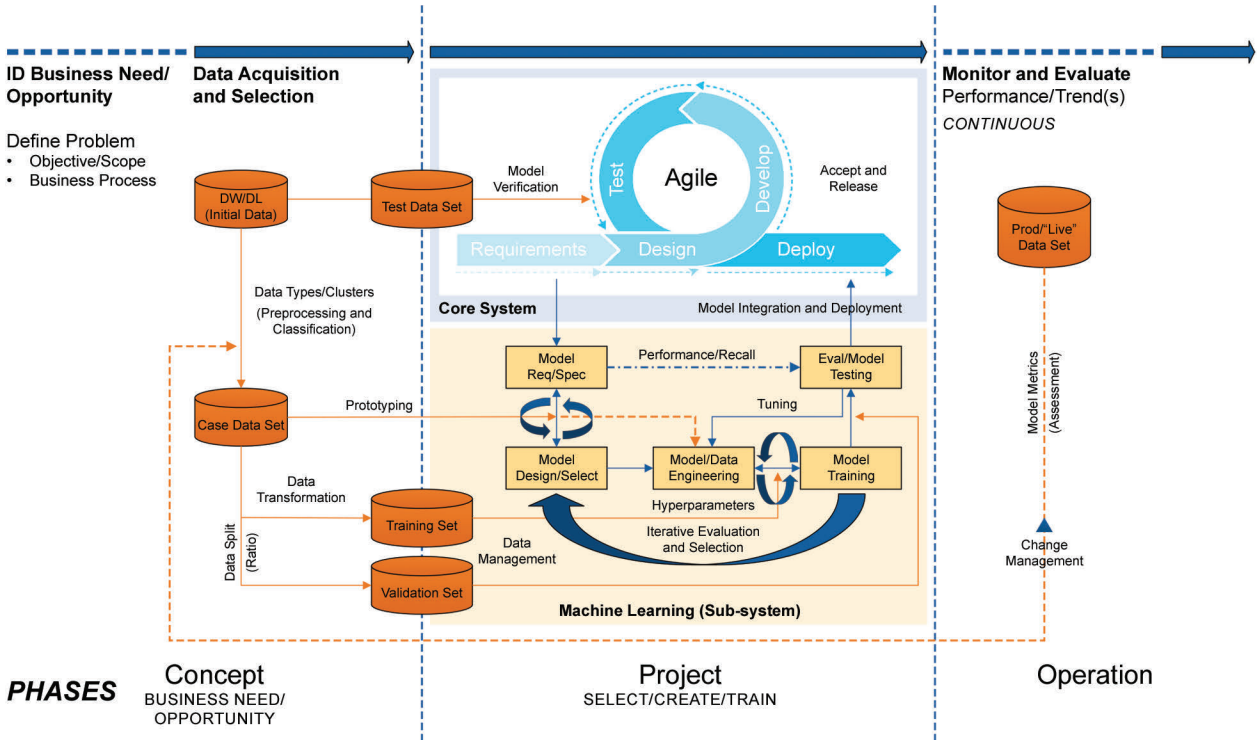
31.3.1 Life Cycle Approach

For most systems that use ML, many aspects of the traditional computerized-system life cycle and compliance/validation approach are fully applicable (e.g., those related to specification and verification of user interface, reporting, security, access control, data-integrity controls, and data life cycle management).

Operational ML sub-systems provide different outputs as they evolve, but the verification and validation approach should be designed to cope with these changes; this must include change and configuration management, version control, and monitoring.

Figure 31.1 is a high-level model that may be used to position the primary actions and activities of an ML sub-system within an overarching GAMP life cycle.

Figure 31.1: GAMP 5 Life Cycle with ML Sub-System



In the **Concept Phase** the business need or opportunity is identified, clarified, and agreed. The specific problem to be solved is defined, and initial data is identified (e.g., from a data warehouse or data lake), selected, acquired, and prepared. Prototyping allows the assessment and selection of suitable algorithms and hyperparameters, and preliminary hyperparameter values. Data management also begins in this phase when the original dataset is collected.

During the **Project Phase**, following a defined plan, the selected technologies and technical architecture are specified. Formal risk-management activities commence, as well as other supporting activities including project-based configuration and change management. Project phase activities for the ML sub-system are typically iterative and incremental rather than linear. These iterative activities include model design/selection, engineering, model training, testing, evaluation, and hyperparameter tuning.

Data management should be active during the project phase, including the acquisition of new data, secure storage and handling, preparation (including labeling), and partitioning into test, training, and validation datasets. The training and validation datasets are used for model training and unbiased evaluation respectively. The test dataset is excluded from all training and hyperparameter tuning activities and used to provide an unbiased evaluation of the final model within the overarching core system. This may be different from approaches in the past where data could be reused multiple times. Here the understanding needs to be known as to which data can be used multiple times versus which can only be utilized once. Successful separation of this data needs to be laid out and explained in order to give confidence that the data is not creating an homogenous state that could only learn from specific values.

In the **Operation Phase** the performance of the system is monitored and evaluated. As new (also known as live) data becomes available, further configuration/coding, tuning, training, testing, and evaluation are performed. There is likely to be a tight and iterative loop of alternating project and operation activities as the availability of new data and ongoing performance evaluation and quality checks lead to opportunities for improved performance, both proactive and reactive, or changing scope of use. This requires effective change management, configuration management, and model versioning. (See also Section 31.3.5.3 on change management).

31.3.2 Concept Phase

31.3.2.1 Business Opportunity and Definition/Data Dependencies

The concept phase to implement ML is where the large shift from management of project phases comes about. Until now, many projects did little or even skipped the concept phase as the business already had a tool in mind. The transition to embracing the concept phase can be cognitively difficult for people on project teams to adjust to at first, but it is vital to the success of an AI/ML-based process. Understanding what is within the capabilities of a company must be determined in the concept phase or the assumptions will create issues on an exponential scale later.

This phase should include research and investigation of which ML algorithms should be considered for development based on cost and risk of development and expected performance. Likewise, algorithms can also be purchased and should be scrutinized to ensure the business needs are met. Information to consider from the source is what process and standards were the algorithms developed against and what evidence will be provided with the purchase.

During the concept phase, the business need is developed and analyzed, the overall process and workflow is defined and agreed, and how the proposed AI/ML application will support the process is identified. This analysis will help determine constraints, such as availability of data, deployment hardware, legal liability, and regulatory and Intellectual Property (IP) factors. Detailed data-related factors such as source, structure, format, and segmentation should also be determined.

This requires consideration of five primary principles:

- Transparency – or how the model(s) will behave
- Reproducibility – or how the outcomes may be trusted
- Interpretability – or what may produce the outcome(s)
- Applicability – or how the model(s) shall be applied
- Liability – or who will be ultimately liable

Once this is accomplished, the primary task is to build a dataset based upon what the organization is trying to achieve, not forgetting to identify assumptions. However, in order to build an appropriate dataset, an organization must understand what information is important to them (i.e., meaningful) and find where that data/information is available, along with how to exclude unwanted/irrelevant data. This is defined by the desired features and/or parameters that will be directly involved in the ML, which can be small and/or even limited in quantity.

This process should begin with, and/or be supported by, a robust data-governance strategy and a realization that not all data is created equal. This includes awareness of data size, quality, and prevalence (see also Section 31.3.5.2 on data governance).

31.3.2.2 Data Acquisition and Selection

There are many places an organization can begin to look at in order to acquire data. The first place is typically internal to their company and given operations. Often this is within an existing data warehouse or data lake. Ideally, the company's data is organized and not siloed. It may not be possible to converge all siloed data streams, as such it is usually in an unorganized format and requires preparation and labeling to become useful to an ML model.

It is also possible to obtain data external to the organization to serve as the basis for learning. This data may come from various sources, some of which may be structured, unstructured, and/or semi-structured. Regardless of the format, there should exist a data selection and governance strategy that aims for a diverse and non-biased dataset. The set should also aim for a large number of data points, keeping in mind a reduction of overall data complexity.

One must consider data classification, clustering (e.g., including data privacy needs), regression, and ranking, as well as what values and/or metadata are critical and which simply add more complexity. It is important to not just have the appropriate or relevant data but also have it in the correct form or format, which is one of the major challenges in the use of external data sources.

If obtaining and/or using external data (i.e., open source or public information), an organization must ensure that they have the right to use such data and that Personal Information (PI)/Personally Identifiable Information (PII) considerations are taken to comply with regulations such as the EU GDPR [79]. If this is not the case, then certain data attributes may need to be anonymized.

Data selection considerations include:

- Source: internal, external open source, external purchase
- Privacy and controls: data classifications, data use, risk, mitigation controls
- Structure/format: unstructured, semi-structured, structured; unorganized or organized
- Segmentation: clustering, priority, or tiering, necessary versus nice-to-have

An initial set of data, free from bias, will be collected from the existing business activities, or needs to be gathered to provide a starting point for the prototyping. Once identified, this stage identifies the activities required to prepare the data for the training and evaluation of the models, including data transformation: formatting, cleaning, and feature extraction. It is also likely that the data needs to be labeled to provide the training inputs and evaluation against which the prototype sub-system will be evaluated. At this phase, it is not expected that the data be complete, as subsequent stages will identify the need for additional data and the plan for acquiring and labeling it. It is, however, important to partition the case data into training and validation sets to avoid compromising future evaluations.

31.3.2.3 Data Types, Preprocessing and Classification

Data preparation includes, but is not limited to:

- Profiling (e.g., formatting)
- Cleansing, transforming (i.e., homogeneous)
- Anonymizing (e.g., for EU GDPR [79]/privacy)
- Augmenting to diversify

Datasets are often not immediately usable, thus the need to prepare it prior to making it available for ML activities. The quality and integrity of the data is critical (the quality of output is determined by the quality of input) and must be suitably classified and labeled (i.e., assigning tags to make it more identifiable for predictive analysis). The data needs to be free of bias to certain regions, demographics, etc. This is one of the most time-consuming efforts in the process and ideally should be done by dedicated data scientists that possess distinct domain knowledge and expertise of the data. This aids in their ability to decide upon relevant data structuring, cleaning (i.e., removal or replacement of missing values), labeling, annotating, and preparation for further processing and use. The data preparation must also be documented and supported with evidence. This is imperative in order to achieve intended outcomes and regulatory compliance, as the understanding and preparation of data is directly attributable to proper selection, build, and testing of models.

31.3.2.4 Data Transformation

Putting together the data in an optimal format is known as feature transformation. This includes the format (e.g., differing files), data cleaning (e.g., removing missing values), and feature extraction (e.g., identifying features or data elements most important for predicting speed and accuracy). Normalizing the dataset also helps to improve these circumstances by reducing dimensions.

31.3.3 Project Phase

The output from this phase is an implementation of the AI/ML sub-system integrated into the overarching IT system together with extensive performance evaluation measures. Integral to this is the development of the training and performance evaluation infrastructure that supports training, tuning, and evaluating the models during the operation phase. Additional tools, supporting model construction or data preparation, may also be developed during this phase that are not be passed onto the operation phase and hence may not be subject to the same regulatory or operation requirements.

This phase follows an iterative approach where successive versions of the AI/ML sub-system are specified, designed/selected, implemented, trained, tuned, and evaluated. This phase consists of a series of experiments that iteratively improves the design, implementation, and hyperparameter selection of the sub-system to optimize performance.

31.3.3.1 Model Requirements and Specifications

At this stage, the initial set of requirements are defined that drives the development and defines the functionality required from the system and ML sub-system including performance.

Nonfunctional requirements such as integration and deployment constraints should also be considered at this early stage to inform the choice of ML algorithms. Nonfunctional requirements also include a set of performance metrics. These are a detailed description of the output of the ML sub-system and how they compare to the gold standard to provide quantitative measures of how well the sub-system performs.

31.3.3.2 Model Design and Selection

A ML model is selected based upon the question it is expected to answer (see Guiding Principle 6 in Joint US FDA/Health Canada/UK MHRA Good Machine Learning Practice for Medical Device Development: Guiding Principles [78]). Common types of models include:

- Classification – categorize data into two or more classes
- Clustering – recognize patterns by attribute to determine targeted action
- Outlier – identify anomalous data
- Forecasting – analyze patterns to predict the future

For every type of model, there is a large selection of model subtypes or architectures. AI/ML projects can benefit significantly from deploying algorithms and techniques developed and applied to other applications and use cases. For a new system, it is unlikely that the choice of algorithm is so clear-cut that a decision can be made to fully specify the component and proceed to development at this stage. In order to determine which algorithm is most suitable and how it should be trained and evaluated, the candidates should be evaluated against the operational, performance, and if relevant, regulatory requirements.

31.3.3.3 Model and Data Engineering

This stage involves constructing the model architectures and the surrounding infrastructure for data input and evaluation that enables training and hyperparameter tuning of the models. The choice of model architecture determines the set of hyperparameters. Tasks include selecting and preparing the data for training iterations and recording results to allow comparison between trials of different hyperparameters and results from different versions of the architecture. Once setup, the infrastructure is then employed to execute a series of trials in which the model hyperparameters are altered to determine the set that result in the best model performance. Subsequent iterations of the development process refine the architecture driven by model test results.

31.3.3.4 Model Training

Training a series of model instances by varying hyperparameter values and recording the results is performed during this stage. Hyperparameter optimization may involve manual selection and altering of the parameters after each iteration, or automated processes using exhaustive search or the more efficient Bayesian optimization of the hyperparameter space.

Most ML algorithms possess many hyperparameters and hence, define a large hyperparameter space over which to optimize. However, the parameter space can be greatly reduced by applying knowledge of the algorithm and problem domain to identify the subset of hyperparameters whose values can be predetermined and fixed. Although libraries and infrastructures exist that allow for automated hyperparameter tuning, data scientists are advised not to take a completely hands-off approach to this activity. Dividing the hyperparameter search space experiments into smaller regions, by allowing only a subset of the hyperparameters to optimize for each experiment run, can provide useful insights on the effect of hyperparameters on the model training and performance, and lead to a more efficient tuning stage.

During model training, the training dataset is supplied as input to the model and the resulting output is compared to the training data ground truth to measure the model's performance and update the model parameters. The validation dataset is used to evaluate how well the model was trained and assists in fine tuning the model(s). For instance, does the defined input generate the correct output? This may require human verification and/or the use of tools. Validation is performed to provide evidence that the accuracy of the model and associated algorithms has detected/defined per output expectations. It is important to consider the need for multiple datasets as the data used may alter the original data and make it unavailable to revalidate. There must also be consideration for dealing with bias due to erroneous assumptions in data selection. This may happen, for instance, when the training dataset(s) is not large or representative enough.

The output from this stage is a trained model using all the training data and an optimal or near-optimal set of hyperparameters. This is considered the best model given the existing fixed architecture and parameters that have been evaluated using the validation data. The iteration of model design – model engineering – hyperparameter tuning – model training – model evaluation reveals insights into the performance of the latest and previous models. This yields further evidence as to how the model architecture and training options may be altered to improve the performance; hence, a redesign or selection of an alternative model may be performed and evaluated.

31.3.3.5 Evaluation and Model Testing

The evaluation and model testing stage is when the best performing models from the previous training and selection iteration are subjected to performance evaluation. The performance output from the models is compared to the gold standard labeled data to produce a set of aggregate and indicative performance metrics, or scorecards. These scorecards inform on the current performance and determine if additional iterations are required. (See Guiding Principle 8 in Joint US FDA/Health Canada/UK MHRA Good Machine Learning Practice for Medical Device Development: Guiding Principles [78].)

Many ML libraries incorporate the validation data evaluation into their training functions, thus automating much of this process. Data scientists, however, should be wary of relying on the quantitative measures for model evaluation. Visual qualitative evaluation of the validation results often leads to better insights on how the model is performing, allowing common error modes to be identified and addressed, and permit crucial refinement of the performance metrics to provide better alignment with the required outputs.

When target model performance is achieved and/or no further changes to architecture are identified, then the best performing ML models are selected as the candidates for integration into the overarching IT system and deployed.

31.3.3.6 Model Integration and Deployment

During this stage, the AI/ML algorithms and models are migrated from the development environment code, which supports fast prototyping and experimentation, to deployment target code that is cleaned and better fit for deployment and long-term maintenance. The process involves removing much of the code designed to support prototyping of candidate algorithms and experimentation. This includes identifying the parameters and algorithm choices to be adopted and removing candidate algorithms that did not yield the desired properties or performance.

Key to this phase is modularization: to isolate the inference module of the code from the remaining code. Inference are the components of the code relating to the forward passing of the test or previously unseen data as input through to the output of the AI/ML sub-system. This excludes any function relating to validating the output against the ground truth, or code involved in altering the model parameters or hyperparameters.

Integration also requires the specification and implementation of the interface between the AI/ML sub-system and overarching IT system.

31.3.3.7 Verification, Acceptance, and Release

The final infrastructure for release, maintenance, and performance verification of the AI/ML sub-system is developed during this stage. Processes relating to the development, release, and maintenance of the AI/ML sub-system are defined to specify if, how, and when the functions of the developing AI/ML algorithms are verified. Choices must be made as to whether the training and possibly tuning of the ML models are included in these processes. For example, it may be decided to run the complete model training, hyperparameter tuning, and model performance cycle on the test data at regular intervals to verify functions of the code. Alternatively, it may be deemed that the model training and hyperparameter tuning are not part of the core code or infrastructure and is excluded from the verification process.

The training data, excluded from all training and validation activities to date, is used as an independent dataset to provide confirmation of acceptable performance (i.e., verification) within the overarching application's User Acceptance Testing (UAT) prior to deployment. This data provides the assurance that certain performance scoring and/or defined outcomes are achieved.

At a minimum the process should specify and document how verification of the AI/ML sub-system shall be performed.

The execution of such processes should result in the release of the first version of the AI/ML sub-system.

31.3.4 Operation Phase

31.3.4.1 Monitoring and Continuous Evaluation

During the operational phase, the AI/ML sub-system should be continuously monitored and maintained. Such monitoring may result in the need to change or modify the AI/ML sub-system (see Guiding Principle 10 in Joint US FDA/Health Canada/UK MHRA Good Machine Learning Practice for Medical Device Development: Guiding Principles [78]).

For static systems, changes must be made in adherence with the organization's change-management process (see Appendix M8), leveraging risk-based evaluation(s) with consideration on the impact to current and future production data to ensure it is robustly tested before deployment. For dynamic systems, appropriate metrics/KPIs and acceptable parameters/tolerance limits must be established to ensure that the algorithm is still operating as intended.

A typical request might be that the system is poor at generalizing to a specific subclass of input data. A typical solution is to acquire and integrate data of this subclass into the training dataset. The integration of additional training data must be systematic in which every change in the performance is measured, validated, and understood.

For example, upon acquiring the additional data, a portion of it could be assigned to the training set and another excluded from all model-training activities. Model training would proceed with the augmented training dataset with the realization that the additional subclass of data may result in an overall drop in performance due to the inclusion of more challenging data. Once trained, tested, and tuned, performance of the revised model should be staged by initially executing the evaluation processes with the original model on the augmented test dataset with an expectation that the performance may drop because of the increased challenge. Then, execution of the evaluation process with the revised model should take place, with an expectation that the performance measures achieve the desired acceptable level.

31.3.4.2 Performance/Trending

In the operational phase it is possible to integrate, enrich, and prepare new data to further refine existing models, develop new predictive models, and establish performance monitoring measures to track ongoing effectiveness. There must exist quality control standards and/or measures to ensure acceptable performance throughout the model(s) usable life. [78]

31.3.5 Supporting Processes

This section describes additional processes needed to support the life cycle of AI/ML systems. The processes presented should be used during all project phases.

31.3.5.1 Risk Management

Risk management helps to drive much of the process and controls needed because it provides a direct indication of the possible harm that could happen in different scenarios. ML is about increasing efficiency while decreasing potential harm. It is imperative that possible scenarios and gaps from the traditional risk-management approach are determined; this means performing risk assessments and then assessing the additional risks of having ML intervention. This also needs to account for the volume and scope of the initial dataset in ensuring that all scenarios, features, etc., were properly trained, and that the training data was correctly labeled prior to use, as this could lead to additional requirements, etc., to be evaluated.

Other risks for consideration include:

- Use of external vendors/suppliers
- Retainment of sufficient technical and scientific knowledge within the organization
- Model maturity and extent of previous usage
- Changes to production data and/or errant data
- System performance, downtime, etc. (i.e., consider business continuity and potential data-loss scenarios)

See Chapter 5 of the Main Body and Appendix M3 for additional details on risk management.

If the ML sub-system is **static** (i.e., offline where there are controlled and identifiable changes to the case data and algorithm) and all data acquisition and annotation are **supervised** (i.e., validated prior to being included in the case data), then standard data validation change-management processes should be sufficient.

Dynamic ML sub-systems (i.e., online learning may be deployed to continually update the model parameters during operation as additional data is acquired) or the use of **unsupervised** data (where there is no ability to validate every single piece of data as it is included in the case data) will require additional controls such as continual and robust performance evaluation.

In all cases, periodic review and monitoring of the model performance is needed to identify potential bias, overfitting, etc. The timing of reviews may be determined based on a defined interval and/or by activity such as the incorporation of new case data.

31.3.5.2 Data Governance

Data governance is critical to the success of ML. Since all data is not created equal, the strategy should include awareness for data size, quality, and prevalence.

Prior to the project phase kick-off, it should be determined if additional data will be needed and whether a separate project is required to obtain that data, determine its appropriateness for intended use, and prepare it for AI/ML sub-system development. Activities include format specification, selection, and application tools for data annotation and cleanup.

The extent and format required for the data is driven by the performance metrics previously obtained; therefore, the training and ground-truth data must be in a form that will enable the desired measurement to be made. For example, for the task of image analysis object localization, the performance metric is specified as overlap agreement between the accepted standard and predicted outcome, which is measured by image segmentation. For a classification task, simple labeling of an image as containing a particular feature may be sufficient.

See Chapter 3 of *ISPE GAMP Guide: Records and Data Integrity* [35] for additional details on data governance.

31.3.5.3 Change Management

Change management is where data controls are integrated into system responses and the process used to maintain control of the system. Defined change management, data management, and quality risk management processes allow iterative change and retraining to protect patients and maintain compliance.

User stories and system controls along with human intervention are a great place to start. Drafting a visual of the process with possible scenarios and mitigation can help identify weak points.

Consider using inputs and outputs:

- Input: Internal procedures may need to be adjusted or managed differently with the use of ML to meet regulatory needs
- Output: Necessary regulatory deliverables required for change management such as if it is SaMD or a specific GxP software (i.e., a clinical system and a laboratory system can have very different approaches and documents needed for changes)
- Input: Details about how a change is initiated and how it impacts the data

See Section 4.2.5.2 of the Main Body and Appendix M8 for additional details on change management.

The motivation for changes to the AI/ML system may be driven by external factors, including:

- Performance Requirements – changes made to improve performance may result in retraining or tuning of the models on existing datasets
- Additional Inputs – changes to case data and/or algorithms to expand compatibility and increase generalization
- Intended Use – changes made to significance of information, healthcare situation, condition

32 Appendix O – Introduction to Operation Appendices

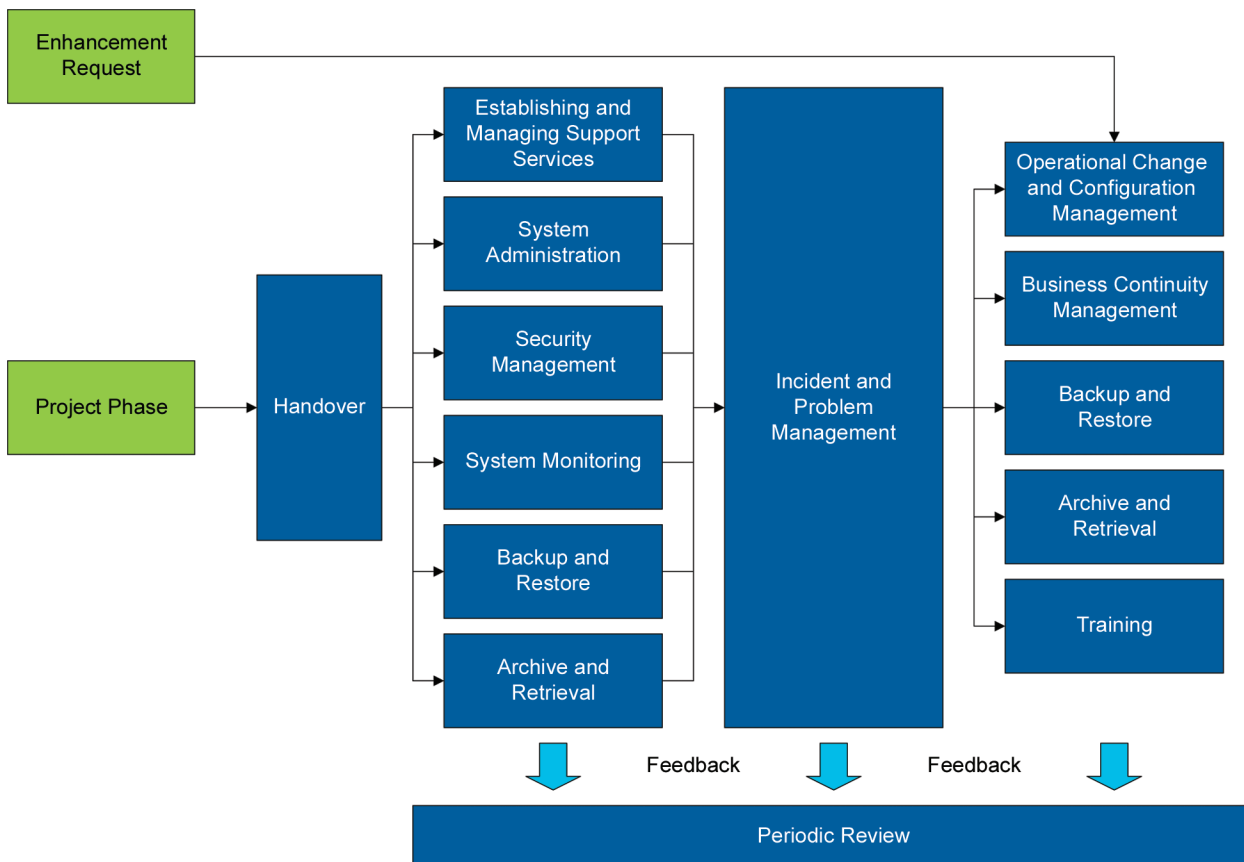
The Operational Appendices provide guidance on how to maintain GxP computerized systems in a compliant state. These appendices ensure that GxP computerized systems continue to maintain patient safety, product quality, and ensure the integrity of GxP data and records.

The Operational Appendices may be used as a foundation for implementing policies, procedures, and/or work instructions within the QMS. The application of good IT practices such as described by ITIL [54], supported by IT service management tools and automation is encouraged.

Scalability should be considered, that is, the controls can be implemented at a level of formality and complexity appropriate to the individual organization and across a wide range of systems. Organizations may have straightforward controls for simple systems and more sophisticated tools and procedures for systems with increased impact, size, and complexity.

In view of the diversity of computerized systems and organizations, it is not considered appropriate to be prescriptive regarding how operational controls are implemented. To ensure compliance with regulatory expectations regulated companies should be able to demonstrate that they have considered and reviewed the maintenance and support needs for each system and decided what procedures, processes, and records should be established and maintained. Figure 32.1 shows the relationships between operational processes.

Figure 32.1: Operational Process Relationships



33 Appendix O1 – Handover

33.1 Introduction

Handover is the process of moving the computerized system from the project phase to the operational phase. This includes the transfer of the responsibilities for maintaining the computerized system in a validated state and maintaining data integrity to the business process owner, data owner, and system owner.

33.1.1 Changes from GAMP 5 First Edition

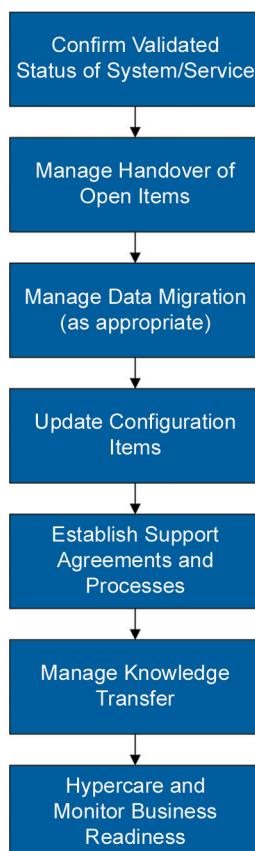
- Provide an updated process flow and expanded definition of handover activities
- Recognize use of support tools
- Include hypercare and business readiness

33.2 Key Requirements

Regulated companies should be able to demonstrate the formal acceptance of computerized systems following the completion of validation activities and verification of the controlled transfer the system into operational use.

33.3 Process

Figure 33.1



33.4 Guidance

33.4.1 General Approach

The project management approach should include a process for handover of a system from the project phase into the operational phase.

Handover activities should ensure that the business and support organization are ready to receive the new or upgraded solution or service following confirmation of successful validation (e.g., approval of the validation report) and should cover:

Confirmation of the Validated State of the Computerized System or Service: The validated state of the computerized system or service should be confirmed (e.g., closure of change controls, approval of validation reports). This should confirm that system specifications are in an “as built” state reflecting the configuration baseline of the validated system or service. Availability of system use, and support procedures should be confirmed as well as business readiness (e.g., organizational transformations completed where appropriate, SOPs updated to reflect changed roles, and training provided in new or modified business processes, data flows, services, etc.). Application components and configuration should be archived to support disaster recovery. Confirmation of inclusion of application components, configuration, and data (including metadata) into backup schedules.

Managed Handover of Open Items: There may be open items remaining at the point of handover including defects, known issues, incomplete training, and documents awaiting final approvals. It is important that such issues are transferred to the business and/or support organizations in a controlled manner and in accordance with risk. Incident logs, change records, CAPAs, deviations, or otherwise appropriate records may be raised to ensure that the open items are appropriately assigned to the new accountable organization. Particular attention should be given to incomplete issues that could negatively impact GxP or compromise the compliance status of the system.

Completion of Data Migration (as appropriate): Data migration/load has been successfully completed, where appropriate, in accordance with a documented data migration plan and data migration report.

Update of Configuration Items: Updates to the configuration management database/configuration items list for the system are complete. Updates to system inventory and/or configuration management database to register new system, modules, and/or infrastructure components have been completed.

Establishing Support Agreements and Support Controls: Establishment of appropriate SLAs, quality agreements, operational level agreements, contracts, or otherwise defining the support requirements is in place. Creation and/or update of operational support procedures in accordance with the O Appendices has been finished. Service management and other tools used in the provision of support services have been established/updated.

Knowledge Transfer: Knowledge transfer to support organizations to ensure they understand the new or upgraded solution or service has been conducted. This should include the transfer of any documentation/records to be used and maintained throughout the support phase (e.g., system life cycle documentation, business process and data flows, risk assessments). System specifications should be up-to-date and reflect the validated baseline of the system.

Knowledge transfer is not only the transfer of documentation and records; it also includes the transfer of knowledge and experiences of the project team, including implicit knowledge that cannot be easily articulated in documentation and records. Lessons learned by the project team should be shared as well with support teams and future project teams to improve the effectiveness and efficiency of ongoing support and future projects. For further guidance on knowledge management see *ISPE Good Practice Guide: Knowledge Management in the Pharmaceutical Industry* [10].

Hypercare and Monitoring of Business Readiness: Depending on the system complexity, a period of hypercare may be required where there is a phased transition from the project team to the support team following handover. During hypercare, support activities are transitioned from the project team to the support team in accordance with defined criteria (e.g., demonstration that post go-live incidents are diminishing in quantity and severity, demonstration that the support organization has the knowledge to support the system). The period of hypercare should be defined prior to implementation of the system into operational use. The duration of continued support by the project team may also be determined by the level of business readiness to operate the solution. For new, complex and/or highly integrated computerized systems, project team support may continue until business users have demonstrated that they are able to operate the solution/service successfully. This may include a measure of business process utilization and levels of business support inquiries during in the early days of operation.

Handover may involve the transfer of the system or service and associated support processes from one QMS to another. This transfer should be carefully controlled, documented, and communicated.

Handover activities may be included in project plans, handover plans, service transition plans, or similar artifacts. IT tools are also often used to plan system cutover and handover activities.

Consideration should be given to defining a period for monitoring the system after handover and to defining a rollback strategy in the event of a significant problem during the monitoring period.

33.4.2 Responsibilities

Project Manager:

- Prepares the system for handover

Process Owner, Data Owner, and System Owner:

- Accepts the system into operational use

The responsibility for completion of any outstanding actions at the point of handover should be agreed between the parties.

34 Appendix O2 – Establishing and Managing Support Services

34.1 Introduction

The process for establishing and managing support services ensures that support services (whether internal or external) are appropriately specified and managed. This is often governed by Service Level Agreements (SLA).

34.1.1 Changes from GAMP 5 First Edition

- Expand considerations for SLAs
- Recognize other aligned agreements such as quality agreements
- Include considerations for contract exit

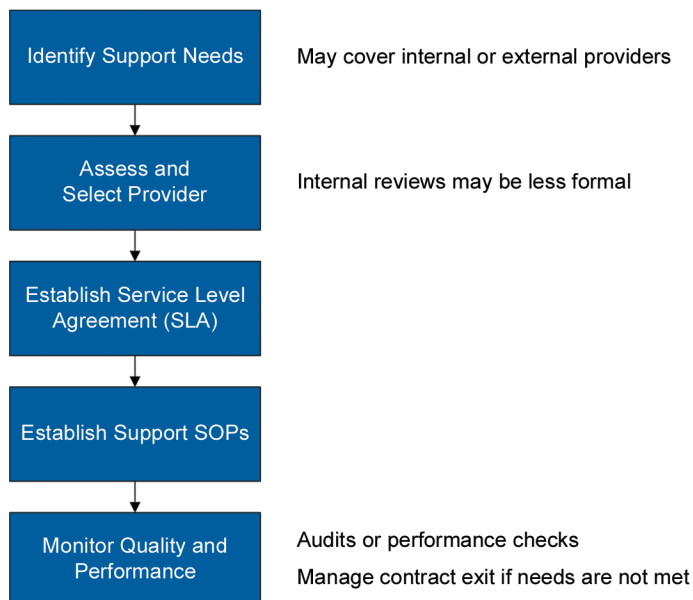
34.2 Key Requirements

When external service providers are used, there should be a formal agreement that clearly states the services to be provided and the responsibilities of the service provider and the regulated company.

In this context *service provider* is interpreted as meaning both external third parties and internal support functions.

34.3 Process

Figure 34.1



34.4 Guidance

34.4.1 General Approach

SLAs describe the services to be provided and service level targets, and specifies the responsibilities of the service provider and the regulated company.

SLAs may be established separately for individual systems or may cover groups of similar systems (e.g., instruments in a single laboratory).

SLAs are typically created and maintained by the service provider organization. Service providers may have a standard SLA that is applied to all customers or may need to create a specific SLA to address non-standard services. Business process owners and system owners should ensure the provisions of the SLA adequately meet business needs.

Some service providers offer different tiers of support that may be selected by the regulated company depending on their specific needs.

The SLA should be agreed, understood, and approved by both the regulated company and the service provider.

The capability of the service provider should be assured and monitored using appropriate service provider assessment processes.

While this appendix focuses on the use of SLAs there may be additional agreements established, which may include the following:

- **Operating Level Agreement (OLA):** An agreement between a service provider and another part of the same organization. An OLA defines the goods or services to be provided and the responsibilities of both parties. An example is an agreement between a primary support organization and a storage management group regarding the time required to restore a file or application.
- **Underpinning Contract (UC):** A contract between a service provider and a third party. The third party provides goods or services that are integral to the delivery of overall support services. An example is a contract between a regulated company and a telecommunications provider for maintenance and troubleshooting of the company's Wide Area Network (WAN).

Contractual and legal implications associated with SLAs, OLAs, and UCs are outside the of scope of this appendix.

Additionally, the regulated company may require the service provider to fulfill the requirements of a quality agreement. The quality agreement defines the quality expectations of the regulated company to be achieved by the service provider.

34.4.2 Responsibilities

System Owner:

- Works with the service provider to establish an SLA that addresses the required services.
- Ensures that supplier assessments and periodic monitoring are conducted in accordance with the risk of the support services provided.

Service Provider:

- Ensures that the support organization understands, follows, and reports against the requirements of the SLA.

34.4.3 General Guidelines

The SLA should define the system to be supported and/or the services to be provided. It should define how the service is to be provided and define responsibilities of the service provider organization and regulated company organization.

Periodic meetings should be established to monitor the performance of the support services against the SLA performance criteria.

The service provider should maintain adequate processes to deliver the services in accordance with the SLA. Quality agreements may also be established to define the regulated company's quality expectations of the service provider's QMS. These should be documented in the service provider's internal quality system and support tools. The regulated company assurance processes should ensure that appropriate support-service controls are in place.

34.4.4 Service Level Agreements

SLAs should address:

- Scope of computerized systems/support services to be provided
- Service descriptions as appropriate:
 - Software releases patch, maintenance, enhancement releases (maintained versions, available upgrades/annum)
 - IT infrastructure and platform services
 - IT security
 - Backup and restoration
 - Business continuity and disaster recovery
 - Information security and data privacy controls
 - Data archiving and retention
 - Record and asset disposal
 - System maintenance, administration, repair, housekeeping
 - Expectations for maintaining different environments for development, testing, and operations
 - Routine testing and calibration
 - Training
- System support tools (e.g., service desk, security monitoring, backup, etc.)
- Roles and responsibilities of the service provider and regulated company organizations
- Reporting, prioritization, and processing times for support requests and faults

- Key contact details (may be maintained in a separate contact list)
- Escalation processes

34.4.5 Measurement, Reporting, and Review

The method of monitoring and reporting service performance should be defined in terms of:

- Key performance metrics (e.g., system and data availability, system downtime, unplanned outages, response times, system incidents, etc.)
- Reporting frequency
- Periodic meetings to review performance

34.4.6 Managing Contract Exit

Support services may be terminated or transitioned to alternative service providers. Contract exit should be set forth in advance to minimize support-service disruption. Contractual agreements should consider expectations for:

- Knowledge transfer
- Documentation/records transfer
- Software code transfer
- Data transfer

35 Appendix O3 – System Monitoring

35.1 Introduction

System monitoring can be used to monitor and report system failures, availability, performance, configuration baseline, and information security issues.

Feedback from monitoring processes and tools can be used to anticipate and respond to potential computerized system incidents and to improve the overall controls environment.

35.1.1 Changes from GAMP 5 First Edition

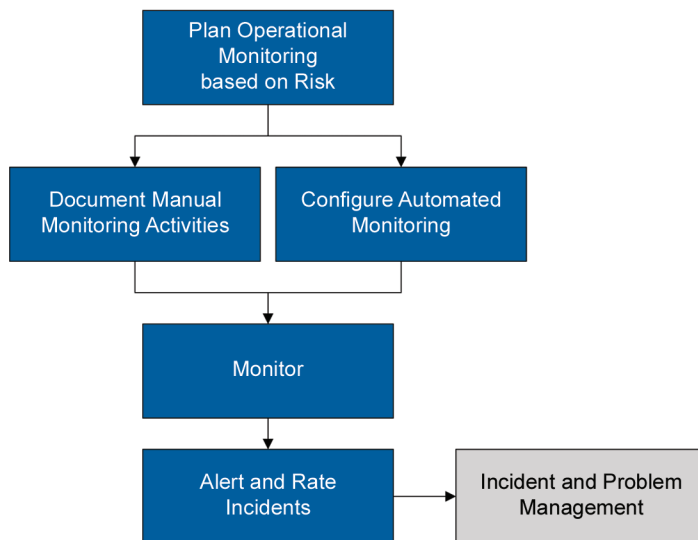
- Include the use of modern monitoring technologies
- Establish link to incident and problem management

35.2 Key Requirements

The need for, and extent of, monitoring activities should be based on business risk and risk to patient safety, product quality, and data integrity. The extent of monitoring shall also be relative to the type of system and its external exposure.

35.3 Process

Figure 35.1



35.4 Guidance

35.4.1 General Approach

System monitoring may cover a specific system, group of systems, or an IT infrastructure environment. The extent of monitoring is dependent on the system complexity, business impact, patient safety, product quality, and data integrity impact, and external threats such as cyber security threats.

35.4.2 Responsibilities

System Owner:

- Ensures system monitoring is in place based on risk
- Works with system administrators, technical functions (e.g., IT) to ensure system monitoring is implemented
- Participates in the incident and problem management process for detected incidents

Business Process Owner/Data Owner:

- Agrees on scope of system monitoring
- Participates in incident and problem management process for detected incidents

35.4.3 System Monitoring Considerations

System monitoring should consider the following:

- System and data availability
- System performance and resources (e.g., CPU utilization, storage capacities)
- Network routing, load, and failures
- Configuration status against configuration templates
- Batch job failures including backups
- System interface failures
- Instrumentation communication failures
- Software, operating system, and hardware failures
- Printer queue failures
- System and process alarms and events
- Response times
- Security vulnerabilities
- Intrusion detection

Note: the above are examples of system monitoring tasks.

35.4.4 System Monitoring Tools

System monitoring should leverage automated tools wherever possible. Such tools can detect and report potential system incidents and automatically alert support organizations who are able to respond in accordance with SLAs.

Some monitoring tools include self-rectification capabilities. For example, resources such as processing, and storage capacity can automatically adjust to address performance issues. Monitoring tools can also automatically reset configuration that does not meet the current configuration template. Such tools can be utilized to minimize the risk of system failure.

35.4.5 Manual Monitoring

Manual monitoring activities can be defined in and scheduled from service management tools or defined in system administration plans or procedures. Monitoring tasks, roles, frequency, and reporting requirements should be clearly defined.

36 Appendix O4 – Incident Management and Problem Management

36.1 Introduction

An incident relates to the effect of an unplanned interruption to a service, or reduction in service quality, typically linked to a breach of the SLA, user observation, or feedback from automated monitoring tools.

A problem relates to the root cause of one or more incidents. Problems can be raised in response to a single significant incident or multiple related incidents.

Problems are the cause and incidents are the effect.

36.1.1 Changes from GAMP 5 First Edition

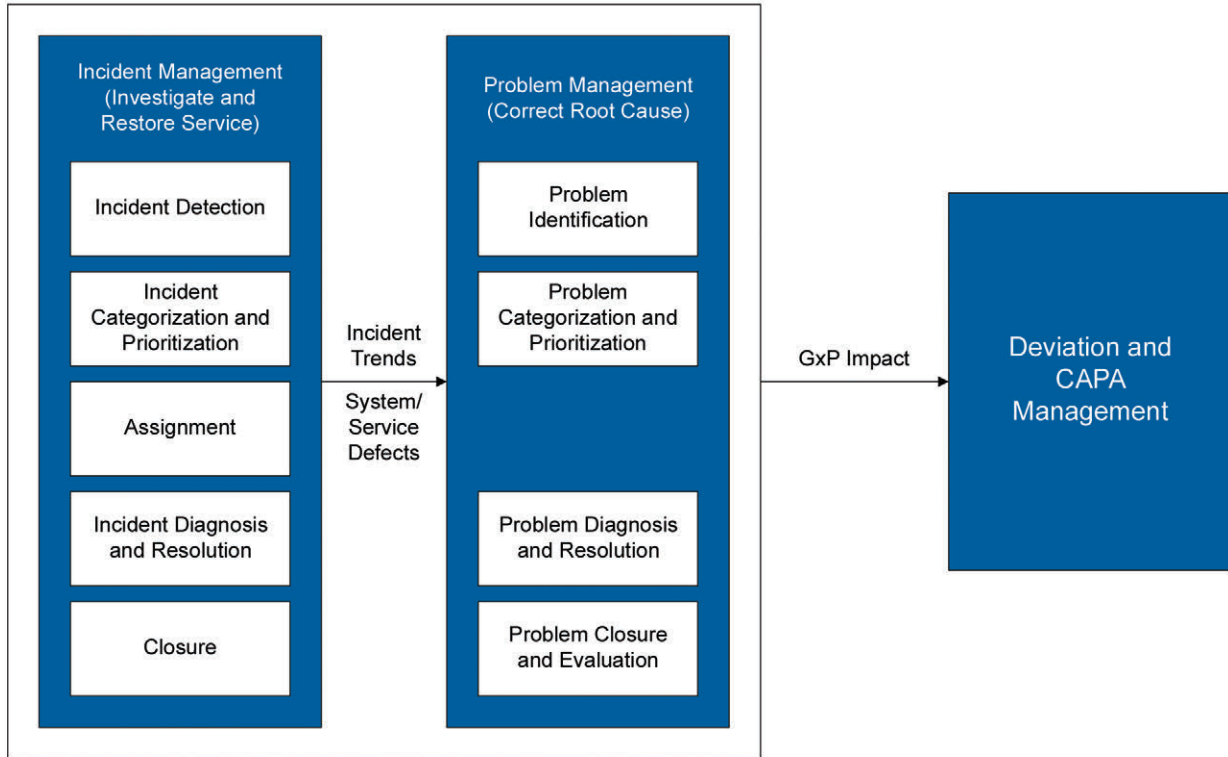
- Further clarification on the approach to incident management
- Describe the relationship between incident management and problem management
- Describe the relationship between incident management and deviation management
- Highlight the use of IT service management tools in incident and problem management

36.2 Key Requirements

Unplanned computerized system interruptions or reductions in service quality should be reported, investigated, and resolved by support teams in order to ensure incidents are effectively managed.

36.3 Process

Figure 36.1



36.4 Guidance

36.4.1 Incident Management

Incidents are reported in a variety of ways including telephone, email, service desk, and automated reporting through monitoring and service management tools.

Reported incidents are categorized and prioritized to determine the response times for investigation and resolution in accordance with defined SLAs. Incidents are captured in an incident management tool. (See Section 36.4.4).

Level 1 support is usually provided by IT SMEs, often those staffing the help desk. Level 1 support teams gather information describing the incident. Level 1 support conducts basic investigations and troubleshooting, and solves common problems such as access issues, solution understanding, setup issues, service restoration, etc.

Level 2 support conducts a more in-depth assessment of the incident and identifies available solutions to address the issue. Level 2 support is generally provided by IT SMEs outside of the help desk staff. If there are no available solutions, a problem is raised and handed to Level 3 support for further investigation and technical resolution.

Level 3 support is provided by SMEs such as developers, Development/Operations (DevOps), architects, engineers, or an external software or IT service supplier. Level 3 support investigates potential design and implementation issues. Technical solutions may be required to resolve the issue.

Major incidents relate to serious disruption to business operations and should be resolved with greater urgency. A major incident team may be assembled comprising appropriate business, technical, and quality expertise to manage the incident. Business continuity and disaster recovery plans may need to be invoked.

36.4.2 Problem Management

Problem management is the process by which the root cause of one or more incidents is determined and solutions are implemented to minimize the risk of similar incidents occurring in the future.

Problem management processes are integrated with other processes such as change management, configuration management, training, disaster recovery, etc., to resolve the problem and minimize the risk of recurrence. Resolution of a problem may also involve GxP CAPA activities (see Section 36.4.3).

Communication is an essential part of the incident and problem management process to ensure that business users and the quality unit are aware of any disruptions or restrictions of service.

36.4.3 Determining GxP Impact

Incidents are generally reported directly to technical functions who are not necessarily SMEs with respect to the impact of incidents and problems on patient safety, product quality, and data integrity.

Configuration Management Databases (CMDB) or other tools/mechanisms should identify configuration items (hardware components, application modules, business processes, records, etc.) that impact GxP. Processes for GxP determination should be agreed by business process owners, data owners, and the quality unit.

The quality unit should be informed of incidents and problems potentially impacting product quality and patient safety. They may raise a quality deviation and initiate a CAPA process to address quality -related actions. IT, system owners, process owners, and the quality unit should cooperate in ensuring effective resolution

36.4.4 Incident and Problem Records

Incident and problem records are typically managed in IT tools and capture information relating to the incident/problem, impact assessments, root cause analyses, and definition of actions taken to resolve the problem, including any workarounds. Such tools may be single purpose or are commonly part of a suite of tools used by the IT help desk.

If changes are required to system life cycle documentation or configuration records, they should be updated and tested in line with any changes implemented to address the problem. Such changes should be implemented in accordance with approved change management procedures.

37 Appendix O5 – Corrective and Preventive Action

37.1 Introduction

CAPA is a process for investigating, understanding, and correcting deviations and nonconformities to address the immediate impact of the issue and to minimize the risk of recurrence.

The scope and depth of CAPA plans should be developed relative to the risk of the associated deviation or nonconformity to patient safety, product quality, and data integrity.

37.1.1 *Changes from GAMP 5 First Edition*

Minor changes were made to improve process flow and understanding.

37.2 Key Requirements

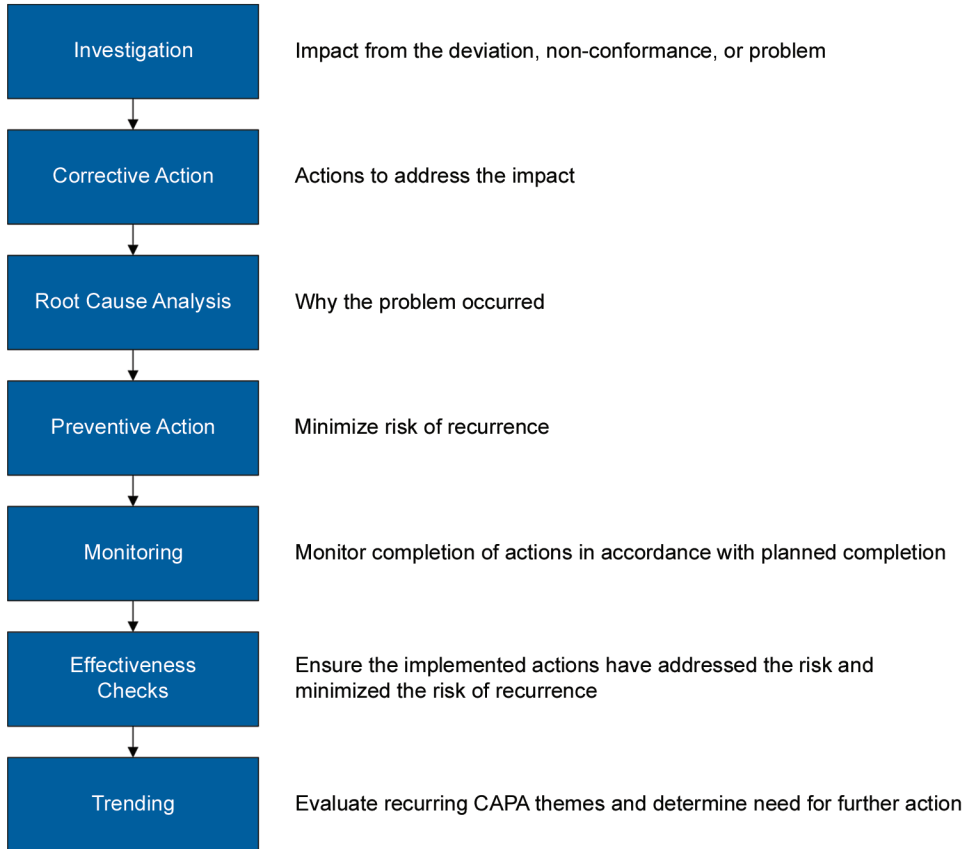
The CAPA process covers:

- Assessment of the impact of the identified deviation or problem
- Corrective action planning to address the impact of the deviation or problem
- Root cause analysis to determine the cause of the deviation or problem
- Preventive action to minimize the risk of recurrence
- Monitoring of action completion
- Effectiveness checks and trending analysis

Computerized system failures and issues are typically reported through the IT incident and problem management process. Incidents related to GxP configuration items should be reported as a deviation and investigated for any impact to patient safety, product quality, or data integrity. CAPAs are created in response to the investigation.

37.3 Process

Figure 37.1



37.4 Guidance

37.4.1 General Approach

The CAPA process is part of the quality system and is used to manage corrective and preventive actions associated with GxP deviations and nonconformities. CAPAs are typically managed within an electronic QMS but may be managed as paper records in the absence of such a system.

Actions arising from CAPAs should be assigned to appropriate SMEs, including technical functions such as IT and engineering.

Planned actions should be proportionate to the identified risk.

37.4.2 Responsibilities

Quality Unit:

- Establishes the CAPA process within the quality system
- Provides oversight of the CAPA process

Business Process Owners and System Owners:

- Review and approve CAPAs to ensure suitability
- Monitor the status of CAPAs relating to computerized systems and services within their remit

Subject Matter Experts:

- Support CAPA planning
- Implement assigned actions in accordance with planned completion dates

37.4.3 CAPA Process

An effective CAPA process investigates and solves problems, identifies causes, takes corrective action, and prevents recurrence of the root causes. The key steps in the CAPA process include:

- **Investigation:** determines the impact from the deviation, nonconformance, or problem. The investigation should determine the impact on patient safety, product quality, and data integrity.
- **Corrective Action Planning:** defines the actions to address the impact. This could include invoking disaster recovery or restoration processes, resolving data issues, reinstating services, etc. Corrective action planning shall also consider actions to isolate the issue and prevent further impact.
- **Root Cause Analysis:** determines why the problem occurred. Root cause analysis is conducted by relevant SMEs.
- **Preventive Action:** defines the actions to minimize the risk of recurrence. Such actions may involve solution enhancements, bug fixes, training, SOP updates, improved system security, improved system monitoring, etc.
- **Monitoring:** ensures CAPAs are addressed within defined targets. Automated or manual monitoring of CAPA target dates ensures reminders and escalations are issued to make certain progress is made against plans. CAPA target date extensions should be reviewed and approved by the business process owner and system owner as appropriate following an assessment of the impact of the extension. Quality Unit oversight should ensure CAPA extensions are being appropriately managed. A review of CAPA status should be included in periodic reviews and relevant audits.
- **Effectiveness Checks:** ensure that the implemented actions are effective in addressing the root cause of the problem being addressed.
- **Trending:** ensures that recurring CAPA themes are investigated and further consideration is given to the CAPAs to minimize the risk of recurrence. Trends may be identified at an organizational, process, or system level. Learning from deviation, nonconformity, and CAPA trends may be used to inform updates to computerized system risk assessments.

CAPA action planning should be risk based and proportionate to the impact on patient safety, product quality, and data integrity. CAPA plans should be reviewed and approved by the business process owners, systems owners, and action owners as appropriate.

38 Appendix O6 – Operational Change and Configuration Management

38.1 Introduction

Change management is the process by which changes to configuration items (e.g., business processes, application software, application configuration, data, IT infrastructure, services, etc.), are managed from their inception to completion. The primary objective of change management is to enable beneficial changes to be made without compromising regulated business processes or records and with minimum disruption to services.

Configuration management comprises the activities necessary to define a computerized system or service at any point during its life cycle, from the initial steps of development through to retirement.

Configuration management and change management are closely related. When changes are proposed, both activities need to be considered in parallel, particularly when evaluating the impact of changes.

Change processes are often supported by electronic systems such as electronic QMS and IT service management tools. Such systems support change workflows, review and approval, record capture, and support integration with other processes such as IT incident and problem management (see Appendix O4), deviation management and CAPA (see Appendix O5).

This appendix covers operational change and configuration management. Project change and configuration management during project development phases is covered by Appendix M8.

The point of transfer from project to operational change and configuration management should be clearly defined before handover to operational use (see Appendix O1).

38.1.1 Changes from GAMP 5 First Edition

- Enhanced process description
- Introduce concept of low-risk standard/routine changes
- Enhance description of relationship between regulated company, IT, and external service provider's change processes
- Describe change relating to cloud services

38.2 Key Requirements

Operational change management should start following system handover to operations and continue until system retirement. Change management should be applied to GxP records following system retirement when such records are still with their defined retention period.

Changes should be specified, authorized, impact and risk assessed, implemented, tested, and approved before release to operations. Impacted documentation and/or records should be updated as appropriate to maintain the configuration baseline of the system.

Configuration records should be maintained to enable the system to be effectively and efficiently restored to service following a disaster scenario.

Note: Configuration records may be maintained electronically, including within the system itself, provided backup and restoration provision is in place.

Risk assessments should be maintained in line with changes.

Testing of changes should be relative to the risk to patient safety, product quality, and data integrity. Testing should demonstrate that:

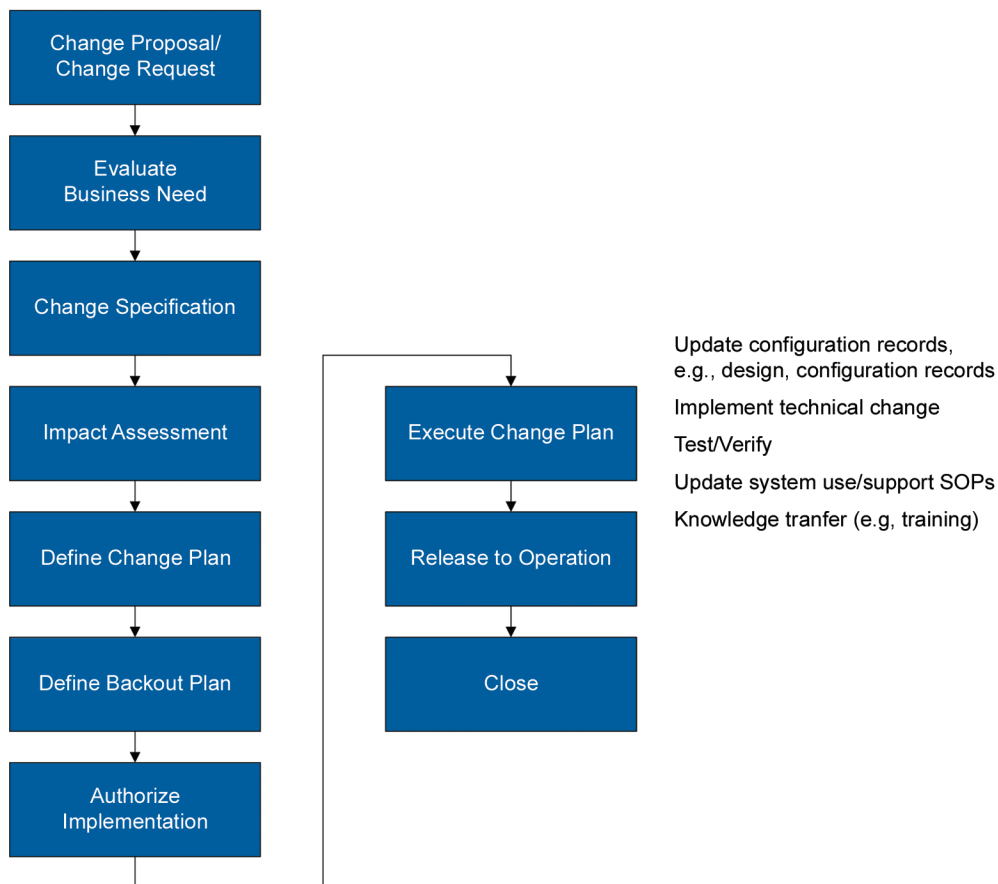
- The change functions as specified
- The change does not introduce unforeseen impacts, i.e., regression testing

Original system risk assessments provide a good basis for assessing the impact of the change and determining the scope of testing. Risk assessments may require updates as a result of the change.

Training needs should be assessed based on the change impact. System use procedures may require revisions when business processes are impacted.

38.3 Process

Figure 38.1



38.4 Guidance

38.4.1 Responsibilities

Process Owner

- Ensures that change control and configuration management process and procedures are in place

System Owner:

- Coordinates changes and evaluates the technical impact of the change

Quality Unit

- Ensures that the process and procedures are followed

Change Plan Task and Action Owners

- Ensure completion in accordance with defined timelines

38.4.2 General Approach

The point of transfer from project change management (see Appendix M8) to operational change management should be clearly defined and documented, e.g., in the computerized system validation report, system release documentation, handover reports, or other means. Operational change management should start no later than handover for operational use.

38.4.2.1 Change Request/Change Proposal and Evaluation

The change request or change proposal is the initial definition of the change outlining the business and/or technical reason for the change and the general scope of the change. The change may be accepted or rejected based on the business value of the change. The outcome of the decision should be communicated to key stakeholders.

38.4.2.2 Change Specification

The change specification defines the change in terms of impacted configuration items (e.g., business process, data, functionality, configuration, service, infrastructure). The change specification may be defined in the change record and/or supported by system life cycle records.

38.4.2.3 Impact and Risk Assessment

Risk assessments should determine whether the change impacts patient safety, product quality, or data integrity. Existing risk assessment records should be updated to reflect the change. The output of the risk assessment influences the change plan and the change verification approach.

Impact assessments should consider the potential impact of the change. This may include:

- Solution regression
- Data
- Business process and data flows (SOPs)
- System life cycle documentation and records

- Training
- SLAs

38.4.2.4 Change Plan

The change plan should define the actions and tasks required to implement the change. The change plan should address the change specification and the output from the impact and risk assessment. Responsibilities for actions and tasks should be defined.

The change plan should clearly define the responsibilities of all parties involved in the change, including the responsibilities of the regulated company and external service provider.

38.4.2.5 Backout Plan

Backout plans should be considered for complex and/or high-risk changes. Backout plans should define how the solution/service is reverted to a known state if the change has an unexpected impact.

38.4.2.6 Authorize Implementation

The change should be authorized prior to implementation to confirm the adequacy of change specifications, impact and risk assessments, change plans, and backout plans as appropriate. Once authorized, the change plan can be implemented. Authorization may take the form of review and approval of the change record.

38.4.2.7 Execute Change Plan

Actions and tasks defined in the change plan are executed. Such tasks include but are not limited to:

- Update system documentation and/or records
- Update system manuals/SOPs/work instructions/SLAs
- Implement functional, configuration, and technical changes (where possible within a validation/test environment)
- Creation/update and execution of tests/verification tasks to confirm correct outcome of the change
- User and support organization training

38.4.2.8 Release to Operations

Once the change has been verified, the change and associated change records are reviewed to confirm all actions in the change plan have been fully executed in accordance with the plan.

Transition of the change to the operations is approved to confirm the change has been implemented and verified. Installation of the change into the operational environment should follow documented installation procedures.

Knowledge transfer to the support organizations should be completed before the transition to operations.

38.4.3 Types of Change

- **Standard/Routine Changes:** Such changes may be considered low risk to patient safety, product quality, and data integrity. Work instructions, system administration plans, support plans, service requests, or similar may be used to define the change implementation tasks and responsibilities. A record of the change should be maintained, including any verification tasks required to confirm successful implementation of the change. Internal audit processes should identify any misuse of the standard/routine change process.
- **Like-for-Like Replacements/Repairs:** These changes may be controlled by maintenance, service management, or system administration procedures designed to control materials usage and record system history. For computerized systems, like-for-like changes may be extended to like-for-kind (similar) changes providing use of such similar components has been prior approved, e.g., replacement of a disk with an alternative higher capacity disk.

Repairs and replacements are typically low-risk changes. Repair and replacement processes may be defined in several ways, including pre-approved or standard/routine changes, system administration plans, or service requests. Replacement components and devices are pre-approved for use as replacements. Approved replacements are usually determined during the initial validation project. Replacements include like-for-like (identical components) and like-for-kind (similar pre-approved components with the same functional characteristics). Repairs and replacements are usually triggered by the incident and problem management process following the detection of a failure. Verification of successful repair or replacement should also be recorded. Where devices hold data or configuration, procedures should define how data and/or configuration is restored to the replacement component, including any verification activities.

- **System Administration Changes:** Some system administration activities may involve changes to system components. Any such changes and associated responsibilities should be defined as part of system administration procedures; see Appendix O12.
- **Emergency Changes:** Emergency changes may be implemented when a delay in the implementation of the change may result in a greater impact, e.g., data loss or data integrity impact, impact on system availability due to a cyber security attack. The criteria for determining an emergency change should be defined in change procedures. Emergency changes may require the retrospective application of the change process to document the change.
- **Temporary Changes:** Temporary changes are planned to be in place for a limited period. Any such changes may introduce new or increased risk that should be assessed and managed. Particular attention should be paid to the reversal of temporary changes to ensure that they are “rolled back” and properly reviewed through the formal change management process before being made permanent. Due to their temporary nature, temporary changes may not require updates to system life cycle documentation and records. The specification and assessment of the change may be contained within the change records. Temporary changes should be monitored to ensure that they do not remain beyond the plan duration.
- **Global Changes:** Changes to global systems and services may require additional governance to minimize the impact of locally identified changes on other functions and regions. For additional information related to managing change in a globally implemented computerized system, see the *ISPE GAMP® Good Practice Guide: Global Information Systems Control and Compliance (Second Edition)* [80].

38.4.4 Business, IT, Engineering, and Service Provider Change Processes

Changes often involve interdependencies between business organizations, internal IT organizations, and external service providers.

A change to business process may be initiated by a business function and encompass all business and technical aspects of the change. The technical aspects relating to the change to the computerized system may be managed by an internal technical function such as an IT or engineering organization. The technical function may follow their own change management process to implement the technical changes.

Similarly, where an external service provider is responsible for managing aspects of the change, e.g., software products, services, infrastructure, etc., the external service provider's change processes should be followed.

Responsibilities for the different aspects of the change should be clearly defined and agreed, e.g., through a quality agreement, operational agreement, or SLA.

Where IaaS, PaaS, and/or SaaS is used, risk-based supplier assessments and monitoring should be considered to ensure that the service provider has appropriate change and configuration management processes in place, and that they are adhered to.

Where a SaaS service provider controls the release of new software versions, the regulated company should ensure that internal change processes evaluate the impact of new releases on business processes and ensure that actions are taken to accept the new release, e.g., SOP updates, training, verification (see *ISPE GAMP Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management* [20]).

38.4.5 Configuration Management

Operational configuration management should start with the baseline configuration and associated configuration management records that are part of system handover.

Configuration management consists of the following activities:

- Configuration Identification (WHAT to keep under control)
- Configuration Control (HOW to perform the control)
- Configuration Status Accounting (HOW to document the control)
- Configuration Evaluation (HOW to verify that control)

Configuration records may take the form of documentation, e.g., specifications, or may be recorded in tools such as application life cycle management tools and configuration management tools/databases. Automated tools, including discovery tools, may be used to maintain an “as built” record of application and IT infrastructure status.

Backup processes (see Appendix O9) should ensure that the current configuration baseline is backed up in support of disaster recovery and system availability (see Appendix O10).

38.4.5.1 Configuration Identification

To support effective configuration management, the system should be broken down into configuration items. These may include business processes, application modules, application components, interfaces, data objects, IT infrastructure components, etc.

Configuration items with GxP impact should be identified to support identification of GxP-impacting changes.

The list of configuration items and their status (e.g., version) is called the configuration baseline and serves as reference for the validated state of a system.

38.4.5.2 Configuration Control

Changes to business processes, software, hardware, data, infrastructure components should be managed in accordance with risk and should only be made by authorized personnel. Configuration records should reflect the implemented changes.

Configuration items should be subject to version control to provide traceability to any implemented changes. Configuration records should be securely controlled and stored to minimize the risk of unauthorized or inadvertent change. Release management should govern the release of changes to operations.

38.4.5.3 Configuration Status Accounting

Configuration records should be traceable to the relevant configuration item and version. As a minimum, configuration records should be traceable to the overall system release.

38.4.5.4 Configuration Evaluation

Documentation and records defining the configuration baseline should be subject to appropriate document management and/or records management controls. Configuration records should be maintained in an as built state. Configuration records are subject to records retention as defined by company records retention policies.

Periodic review of operational systems should ensure that the configuration status of defined configuration items is maintained in line with changes.

38.4.6 Cloud Service Provider Changes

Changes may be scheduled, for example periodic updates by an IaaS, PaaS, and/or SaaS provider, with limited or no opportunity to evaluate the impact of the change. Supplier assessment and monitoring processes should ensure that the service provider has a robust change and release management process that ensures such changes are not released without appropriate change management and verification.

The regulated company should be aware of the service providers release schedule and should evaluate the impact of releases on business processes and regulated data. System use procedures should be updated to reflect any impact on business processes. User training should be provided as appropriate for major upgrades. Where appropriate, the regulated company should conduct testing to confirm business processes have not been adversely impacted by the changes. This may include leveraging supplier testing through risk-based supplier evaluation.

Infrastructure support and/or DevOps teams manage changes to IT infrastructure and platforms using standard configuration templates (e.g., VM templates or IaC). Tools for managing such changes often follow a standard (configurable) life cycle that ensures changes are assessed and approved prior to deployment to controlled environments. Automated monitoring may be used to deployed configuration changes and monitor unauthorized changes to the IT infrastructure, including reverting back to approved configurations.

38.4.7 Software Development Change Management

Software development processes commonly follow an iterative or Agile development approach. Such processes are followed to manage the implementation of software enhancements and bug fixes. System development and support processes should ensure that all software changes are adequately assessed, specified, built, and tested to ensure they are fit for purpose. As such, suppliers and services providers may not differentiate change management from normal software development processes.

39 Appendix O7 (Retired)

Repair activities are typically managed as standard changes, service requests, and/or system administration tasks. As such, the requirements for repair activities are defined in Appendix O6. Appendix O7 has been removed from this Guide.

40 Appendix O8 – Periodic Review

40.1 Introduction

Periodic reviews are conducted throughout the operational life of a computerized system to verify that it remains in a validated state, complies with current regulatory requirements, is fit for intended use, and satisfies company policies and procedures. The review should confirm that operational controls are in place and are being effectively applied.

40.1.1 Changes from GAMP 5 First Edition

- Risk-based approach to periodic reviews
- Utilization of metrics and trends to determine fitness for intended use
- Sample-based review of records

40.2 Key Requirements

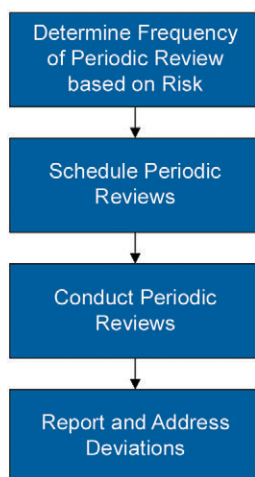
A documented risk assessment should determine the frequency of periodic reviews. The risk assessments should evaluate the impact on patient safety, product quality, and data integrity. Risk assessments should also consider system complexity and system stability (for example, mature systems not undergoing regular change).

Identified deficiencies should be recorded with a prioritized action plan to mitigate the risk. The outcome of the periodic review and associated actions should be reviewed by business process owners, data owners, system owners, and the quality unit.

Execution of actions should be monitored to ensure timely completion.

40.3 Process

Figure 40.1: Periodic Review Process



40.4 Guidance

40.4.1 General Approach

The periodic review may be conducted as a dedicated review or may leverage the outcome of routine reviews conducted against different operational activities, e.g., user-access reviews and problem management reviews.

The periodic review process should provide guidance on what to examine during the review. Where possible, the periodic review should use metrics to determine the stability and fitness for intended use of the system.

40.4.2 Responsibilities

Process Owner:

- Ensures that periodic reviews are conducted and actions arising from the periodic review are implemented

Quality Unit:

- Ensures that periodic reviews are scheduled, performed, documented, and that actions arising from the review are completed in accordance with plans

The review is conducted by a multi-disciplined team including business process owners, data owners, system owner, SME, users, IT, engineering, and the quality unit.

40.4.3 Timing and Scheduling

Periodic review frequency should be based on a documented risk assessment that considers the impact on patient safety, product quality, and data integrity. The risk assessment should also consider system complexity.

The frequency of periodic reviews may be adjusted based on current and historical information, for example:

- Increasing the frequency of periodic reviews when significant incidents occur that suggest operational controls are not effective
- Decreasing the frequency of periodic reviews based on system maturity, e.g., systems that are not subject to significant change or incidents

The periodic review frequency should be documented, e.g., in validation reports, system inventories, or configuration management database.

The responsibility for managing the timing and scheduling process, and allocating resources to reviews, should also be clearly defined.

Where a SaaS solution is provided, the service provider should demonstrate that periodic reviews are conducted for all aspects of the solution under their control. Supplier assessments, including monitoring, should be conducted to confirm that reviews are carried out. Supplier periodic review participants may differ in the supplier organization, e.g., product owners, DevOps, support, infrastructure management, information security, data privacy, and quality unit. Regulated company periodic reviews ensure that SLAs and quality agreements are in place and that supplier assessment and monitoring is being conducted in accordance with planned schedules. Actions from supplier assessments should be addressed.

40.4.4 Review of a System

40.4.4.1 Preparation

The scope of the periodic review should cover the period since the last periodic review or since the issue of the last validation report where a previous periodic review has not been completed.

Records should be reviewed on a sample basis, selected by SMEs applying critical thinking and taking into account system risk, complexity, size and novelty, and including information on incidents and changes.

Wherever possible, the periodic review should leverage existing ongoing reviews such as user-access reviews, problem management reviews, etc., to avoid duplication of effort.

The following information should be collated to support the periodic review:

- Policies, SOPs, and work instructions relating to the business processes and computerized system validation and compliance
- Risk assessment records
- Last periodic review report
- Current validation plan and report
- Internal audit observations
- System life cycle documents/records (including traceability)
- Change and configuration management records
- Incident, problem, and service management records/fault and error logs
- User-access management records
- Periodic user-access review records
- Backup and restoration records
- Deviation and CAPA records

Personnel supporting the periodic review should be notified, including expectations for providing the sample of records to be reviewed.

40.4.4.2 Conducting the Review

Each team member should clearly understand their role in the review and the level of assessment to be conducted. The periodic review is intended to provide assurance that operational controls are effective and is not intended to re-execute work that has previously been reviewed and approved.

The periodic review should assess:

- Action Closure
 - Open actions from previous validation reports, periodic reviews, audit reports, CAPAs, etc., addressed
- Operational SOPs
 - Operational SOPs in place per O Appendices
- System Inventory and CMDB
 - Up-to-date for current release of computerized system
- Change Management
 - Validated state of system changes
 - Maintenance of system life cycle documentation following change (e.g., business process, user requirements, data life cycle documents, design, etc.)
 - Re-evaluation of risks following business process and functional changes
 - Appropriate use of change processes (e.g., emergency changes, routine/standard changes, like-for-like changes)
 - Change trends (for example do trends indicate design or validation weaknesses)
 - Change authorization
- Incident and Problem Management
 - Incident trends (for example do trends indicate design or validation weaknesses)
- User-Access Management
 - Appropriate user role assignments based on business role
 - Appropriate segregation of duties
 - Management of system administration/superuser/service accounts is appropriate
 - User training prior to granting system access
- Information Security Controls
 - Adequacy of controls to assure data integrity
 - Patching
 - Virus protection
 - Vulnerability monitoring

- Backup and Restoration
 - Scheduled and operating
 - Investigation and mitigation of backup failures
 - Periodically tested
- Record Archiving
 - Tested to ensure recoverable in readable form throughout retention period
- Continuity and Disaster Recovery
 - Approved Business Continuity Plans (BCPs) and Disaster Recovery (DR) plans in place (as required)
 - BC plan rehearsals in accordance with schedule
 - DR plan testing in accordance with schedule
- Support and Maintenance
 - SLAs and contracts in place
 - Routine support tasks conducted in accordance with schedule
 - See Appendix O4
- Audit Trails
 - Verification that audit trails remain enabled for GxP records
- Training
 - Training records maintained for new users

40.4.4.3 Output from the Review

The periodic review should be documented, and prioritized action plans created, with appropriate accountability, to address identified deficiencies. A statement of acceptability for continued use of the system should be provided. The timing of the next periodic review should be considered based on the deficiencies and risks identified.

41 Appendix O9 – Backup and Restore

41.1 Introduction

Backup is the process of copying data, records, configuration, and software to protect against loss of integrity or availability of the original. Restore is the subsequent restoration of data, records, configuration, or software when required.

Backup and restore should not be confused with archiving and retrieval processes, which are covered by Appendix O13. Backup supports DR whereas archive supports the ability to access records in a readable and meaningful form for the duration of the records retention period. **Backup should not be used as a means of archive.**

41.1.1 Changes from GAMP 5 First Edition

- Consideration of new technologies, approaches, and cloud services
- Provide a life cycle description of backup processes

41.2 Key Requirements

Backup strategies may include cloud backup, use of backup appliances, snapshots, disk to disk, disk to tape, image backups, and replication to an alternative data center or cloud environment.

Procedures should define the backup and restoration approach including the management of backup failures.

Backup technologies and/or services should be based on business need. Backup scheduling should be consistent with disaster Recovery Point Objective (RPO).

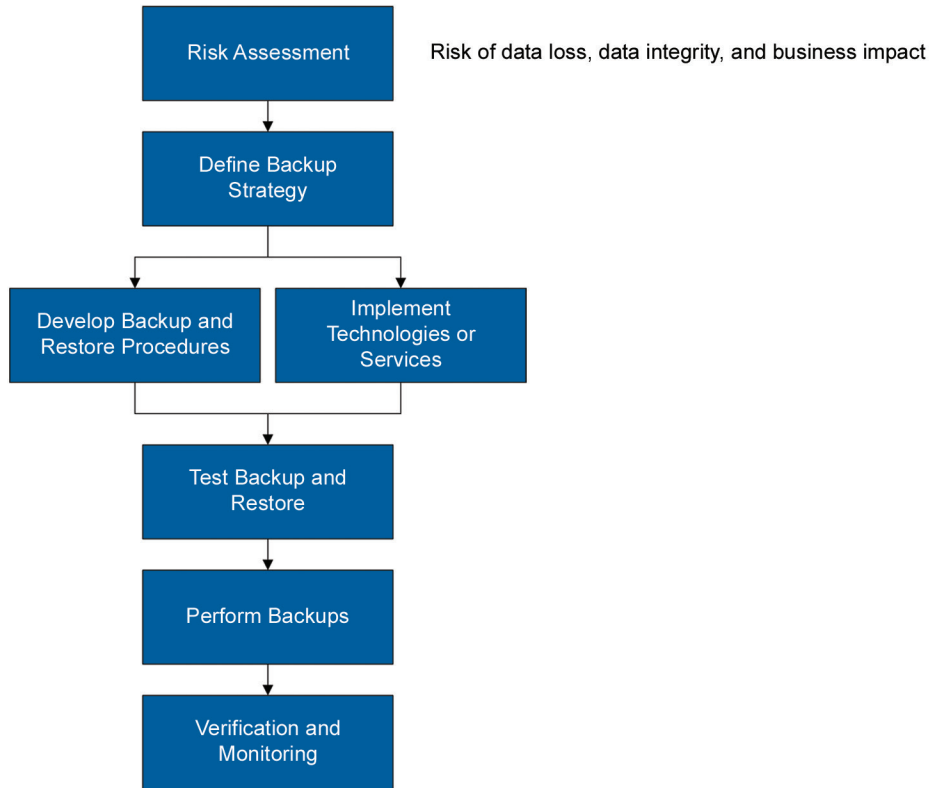
Backup procedures and technologies should be verified to ensure correct operation.

Backup storage locations should be separated from the primary location. Geographical separation should be based on risk and consider natural hazards, e.g., earthquakes, hurricanes (see Section 41.4.3.4).

Contracts and/or SLAs should be in place with service providers. Service providers should be subject to supplier assessment in accordance with a documented risk assessment.

41.3 Process

Figure 41.1



41.4 Guidance

41.4.1 Responsibilities

Process Owner and/or Data Owner:

- Define when external service providers should be used; ensuring assessment of service providers in accordance with a documented risk assessment
- Establish and monitor compliance against SLAs, quality agreements, and/or contracts relating to backup and restoration services
- Define RPO and Recovery Time Objective (RTO)
- Definition of backup scope (e.g., data and records, configuration, application components)
- Backup retention requirements based on business need

The System Owner or Service Owner:

- Ensures technologies and/or services are established to meet (as a minimum) business needs, RPO, and RTO.
Note: backup frequencies may exceed RPO and RTO when a centralized backup service is provided.

- Verifies the operation and performance of the backup solution
- Detects and manages backup failures

41.4.2 Risk Assessment

A documented risk assessment should inform the backup strategy. The risk assessment is usually conducted when planning overall disaster recovery requirements.

The business process owner and/or data owner should define the RTO and RPO.

The backup and restoration strategy should ensure that the data, records, configuration, and applications can be recovered in accordance with the defined RPO and RTO.

41.4.3 Backup Strategy

The backup and restoration strategy should address the defined RPO and RTO. Several backup approaches can be adopted including cloud backups and historical approaches using portable media.

41.4.3.1 Cloud Backup Services

Cloud backup services are increasingly used to support backup of data, records, configuration, and applications. Several cloud backup approaches may be used, including:

- Replication to the Cloud – The regulated company's internal backups are copied to the cloud for secondary storage
- Backup to the Cloud – The regulated company's data, records, configuration, and applications are directly backed up to the cloud
- Cloud Backup Services – The service provider backs up data, records, configuration, and applications hosted in a cloud environment
- Cloud-to-Cloud Backup – Data, records, configuration, and applications stored in the cloud (e.g., SaaS application and data) are replicated to another cloud environment

When hosting data and applications in the cloud in virtual environments, virtual servers and storage devices can simply be added to the backup schedule when deploying the device.

Service providers typically manage backup failures. Customers are alerted to significant backup incidents in accordance with incident and problem management processes.

41.4.3.2 Backup Appliances

Backup appliances may be used to manage backups. Backup appliances are commercially available solutions that include backup software and storage capability. The most recent backup is typically retained in the backup appliance in addition to being stored in a secondary location such as cloud.

41.4.3.3 Backup Using Portable Media

Backup should use suitable media in accordance with manufacturers recommendations. When choosing and using portable storage media, the following should be considered:

- Recommended service life

- Acceptable environmental conditions for storage
- Periodic verification of backup integrity

Guidance on the storage, transportation, and maintenance of various types of magnetic and optical media is available from national and international standards organizations.

41.4.3.4 Facilities and Storage

Backups should be stored in a separate secure location. The geographical separation of backups should be based on risk. Backups should be physically secured and protected from fire, water, and other hazards. The storage process, standards, and access should be defined and documented.

41.4.4 Backup and Restore Procedures

Backup and restoration procedures should address:

- Backup instructions
- Restoration instructions
- Approval responsibilities for restoration from backup
- Backup identification (e.g., backup naming, labeling, inventories as appropriate for identification of backup content and timing)
- Backup approach (e.g., full, differential, incremental)
- Technologies and/or services used as required to meet RPO and RTO
- Backup scope (e.g., data, records, configuration, applications)
- Backup frequency in response to RPO
- Backup separation requirements
- Backup monitoring and management of backup failures
- Periodic backup integrity verification

Many backup approaches are fully automated and therefore backup steps may simply include instructions on how new backup requirements are added to the backup schedule (for example configuration of backup tools/service). This may also be defined in service requests within a service management tool.

41.4.5 Implementation of Backup Technologies and Services

Backup technologies and/or services should be selected based on the overall backup strategy. The correct operation of backup technologies should be tested to ensure that backup and restore processes function correctly.

Contracts and/or SLAs define the backup services to be provided and responsibilities of the service provider and customer.

41.4.6 Verification and Monitoring

Backup and restoration should be tested during the validation of the solution and following any changes in technology and/or approach.

Scheduled backups should be monitored to ensure successful completion. Backup failures may be determined through review of backup logs/reports or through automated notifications via incident and problem management processes and tools.

Backup logs should be retained in accordance with company records retention policies as evidence of backup completion. Persistent backup failures should be investigated and resolved in accordance with incident and problem management procedures.

Backup integrity should be periodically verified. The frequency and approach to backup integrity verification should consider:

- Risk assessments
- Backup retention periods
- Backup approach, technologies, and storage media

It should be noted that backup integrity verification does not require restoration to production environments.

It is acceptable to combine testing of the backup process with testing of DR procedures (see Appendix O10).

41.4.7 Specific Considerations

41.4.7.1 Software and Configuration Backup

Backup of software components and system configuration may not be conducted at the same frequency as data and records backups. When determining the timing of software and configuration backups, the following should be considered:

- Backups that enable backout prior to change implementation
- Backups post implementation of changes to enable DR
- Periodic backups to ensure that routine changes such as configuration, security patches, middleware updates, application patches are covered by the backup

Software development environments should be subject to periodic backup to protect against the loss of source code, configuration, and executable code. Typically, this will include the backup of source code and integration management tools.

- Software backups should be identifiable to enable restoration of a specific version/build

41.4.7.2 Specific Considerations for Data Backup

Data backups should also include metadata, which provides the context and meaning of data. Specifically, GxP-audit trails should be backed up.

41.4.7.3 Backup Metadata

Backup solutions often maintain backup metadata. The backup metadata contains information about each backup, including the system on which the backup was created, restore points, and the relationship between restore points. This metadata must itself be backed up otherwise the ability to restore following a backup solution failure will be impaired.

42 Appendix O10 – Business Continuity Management

42.1 Introduction

The regulated company should perform BC planning to actively protect its ability to continue to supply the public, maintain safety, and to comply with the regulatory requirements. Where DR plans can restore operations within an acceptable time, a separate BC plan may not be necessary.

BC is the documented procedure that guides organizations to respond, recover, resume, and restore to a defined level of operation during and following disruption. BC focuses on keeping businesses operational during a disaster, while DR focuses on restoring computerized systems and data to a known point (RPO) within an agreed time (RTO). Disaster scenarios and BC provisions should not be limited to computerized system failures. Considerations include:

- Loss of application components
- Loss of IT infrastructure
- Loss of a service provider
- Loss of access to premises
- Network failures
- Cyber attacks
- Pandemics

For large integrated solutions (e.g., enterprise resource planning), BC planning may be challenging during a prolonged system outage. In such cases, DR often ensures BC through the provision of high-availability solutions and rapid recovery solutions. In such cases, BC is provided by high-availability solutions. This should be clearly documented in DR and/or BC plans.

DR is typically the responsibility of a support organization such as IT and is likely to include external service providers. DR is typically a shared responsibility between the regulated company and service providers. For IaaS, PaaS, and SaaS, the role of the service provider will increase.

The regulated company should ensure that SLAs and contracts include responsibilities for DR. Responsibilities should include communication of major incidents requiring DR. Shared roles and responsibilities should be clearly defined. Supplier assessment and monitoring should ensure that service providers fulfill their responsibilities relating to DR planning and periodic testing.

The need for and extent of the BC plan and DR plan should be based on a documented risk assessment that considers the risk to patient safety, product quality, and data integrity.

BC and DR may be planned at several levels. For example, BC may be addressed at a corporate, departmental, and process level. DR may be planned at the IT infrastructure, computerized system application, and data level.

42.1.1 Changes from GAMP 5 First Edition

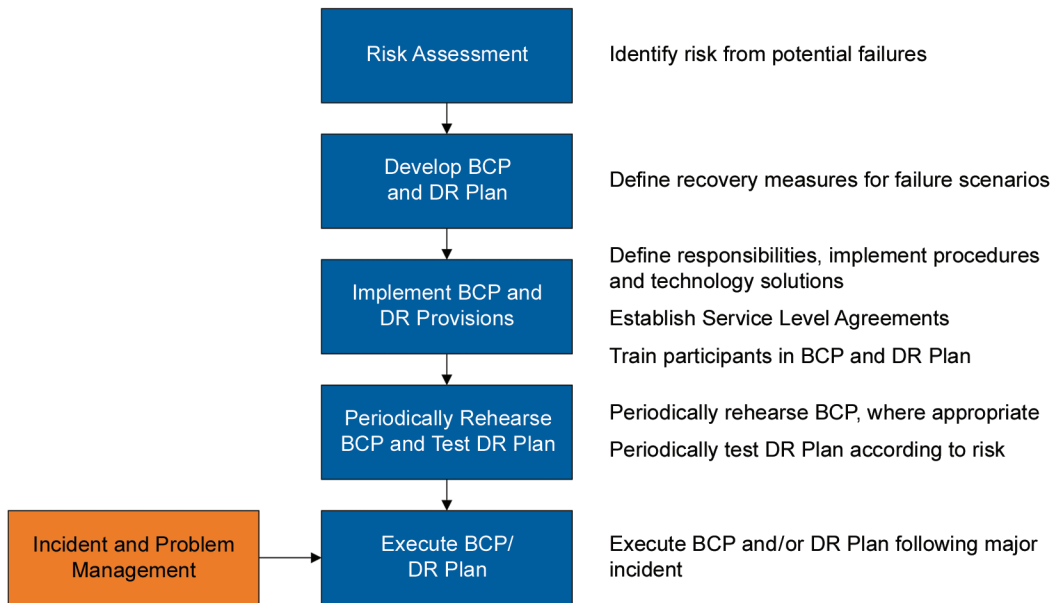
- Further clarified BC and disaster recovery processes
- Established link to incident and problem management
- Considerations for cloud services (XaaS)

42.2 Key Requirements

Patient safety, product quality, and data integrity should not be compromised by disaster scenarios.

42.3 Process

Figure 42.1



42.4 Guidance

42.4.1 Business Continuity Planning

BCPs should define alternative processes to be followed during a prolonged disruption that allow the safe continuance of business during the disruption. Such processes may be manual and/or may rely on alternative computerized systems, environments, and/or facilities.

BCPs should be rehearsed. Alternative processes and materials required by the BCP should be suitably documented and available during a disaster scenario. Roles of all parties (internal and external) should be clearly defined, and personnel should be adequately trained.

Regulated companies should be able to demonstrate that critical services and processes can continue, and that there is a timely resumption of essential business functions.

Regulated companies should define a process for BCP creation that should include a documented risk assessment that identifies failure scenarios that require contingency arrangements.

BCPs may cover multiple facilities, systems, and organizations (internal and external) supporting the business process. For some complex highly integrated processes and systems, it may not be possible to operate the business processes without the supporting computerized systems. In such cases, high-availability DR solutions may be required to minimize the risk of an outage and the duration of any disruption.

BCPs should consider measures for minimizing the threats and vulnerabilities that could lead to a disaster scenario. Such measures include:

- Review the DR strategy (including solutions to maximize availability)
- Review appropriateness of backup and recovery procedures
- Review of security controls
- Establish failure and backup systems, e.g., design in redundancy, uninterrupted power supplies, high-availability solutions
- Establish and ensure availability of any resources necessary for alternative business solutions needed until the disruption is resolved
- Avoid single-supplier agreements
- Failure detection systems, e.g., monitoring of software, hardware failures, intrusion detection and monitoring, automated monitoring of the configuration baseline
- Ability to work from alternative locations

BCPs should include provisions for re-establishing the status quo once the disruption has been rectified, for example:

- Manual records established during disruption being synchronized with the computerized system following recovery
- Reprocessing of operations and data that were not included in backup at the time of failure
- Retention of manual records established during the outage period

Where BC includes alternative arrangements such as facilities, alternative IT provisions, etc., relevant contracts should be established. Where BC is dependent on third-party organizations, BC rehearsals should consider their inclusion.

BCPs can be rehearsed or tested in several ways, such as through reviews, brainstorming, walkthroughs, testing of sections of the plan, reinstallation of servers, switching to hot sites, and full testing.

42.4.2 Disaster Recovery Planning

A documented risk assessment should identify the potential disaster scenarios. The risk assessment should include the company workforce, suppliers, facilities, IT infrastructure, applications, and data. The risk assessment should evaluate all threats to BC. Threat considerations may include:

- Natural events, e.g., fire, floods, weather

- Failure of power sources
- Hardware/infrastructure component failures
- Software defects
- Information security risks
- Service provider failures
- Human error
- Prolonged unavailability of personnel, for example due to a pandemic
- Cyber attack

The identified risks should be evaluated against the RPO and the RTO to define appropriate recovery strategies. Such strategies may include to:

- Availability of spare parts
- Backup restoration
- Rerouting networks
- Re-establishing connection to external services, e.g., IaaS, PaaS, SaaS
- Alternative systems
- Alternative premises
- High-availability solutions
- Repair

DR plans should address:

- Immediate steps to be taken to minimize further impact
- Actions to be taken to recover the situation including the order in which systems must be brought online if relevant
- Actions to be taken to implement alternative working arrangements (BCP)
- Roles and responsibilities for recovering the situation
- Communication requirements

A DR plan may involve multiple organizations responsible for facilities, IT infrastructure, and computerized systems. DR plans and supporting contractual arrangements should ensure that all parties understand their roles.

DR testing should be conducted periodically in accordance with risk. DR testing should ensure that IT infrastructure, applications, and data can be recovered in accordance with the RPO and RTO.

Where third-party organizations conduct DR testing as part of their service, supplier assessment and monitoring should confirm that DR testing is conducted in accordance with contractual agreements.

Incident management procedures should define how a disaster scenario is invoked and the governance of the incident. This may include establishing a major incident team comprising business, technical, and quality representatives to oversee the incident. Links between the incident management process of the regulated company and service providers should be defined in processes, plans, and/or agreements.

The communication process is an important aspect of BC Planning and DR Planning, key contacts (e.g., process owner, system owner, supplier(s), and quality unit) should be listed with their contact details.

The DR plan should include a clear process for prioritizing system restore as the disruption may involve failure or unavailability of multiple systems that may be within or outside the regulated company.

DR plans may need to phase the restoration of operations. For example, initial restoration activities may re-establish immediate operation. Later restoration processes may be required to recover historical data.

42.4.3 Responsibilities

Company Management (including Process Owners, System Owners and Quality Unit):

- Ensure that appropriate BCPs are established, are tested periodically, and once initiated, are followed, and reported upon.

Process and System Owners:

- Ensure that appropriate DR plans are in place for their systems to support the BCPs and are tested.

43 Appendix O11 – Security Management

43.1 Introduction

Security management is the organizational, process, and technical considerations that ensure the confidentiality, integrity, and availability of an organization’s regulated computerized systems, data, and records.

Effective security management minimizes the risks from internal and external information security threats.

43.1.1 Changes from GAMP 5 First Edition

This appendix has been updated to include considerations of current data IT security practices aligned with industry standards such as ISO 27001 [44] and the National Institute of Standards and Technology (NIST) [81].

43.2 Key Requirements

Security measures include:

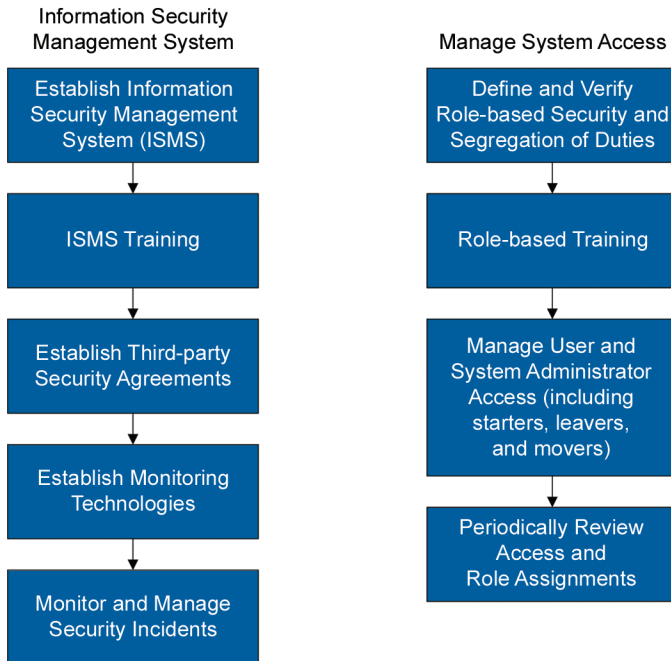
- Establishing and maintaining organizational, procedural, and technical controls to minimize the risk of unauthorized or inadvertent access to computerized systems, data, and records
- Managing role-based system access for user and system administrators, including segregation of duties
- Establishing manual and automated monitoring of computerized systems and environments to identify and respond to potential vulnerabilities and intrusions
- Following incident and problem management processes to evaluate and mitigate potential security risks

Security controls should be continuously assessed to address and respond to new threats to the computing environment.

The scope of security measures depends on several factors including the scope and criticality of regulated processes, records and data maintained by the system, and whether the computerized system is externally facing (i.e., connected to the internet).

43.3 Process

Figure 43.1



43.4 Guidance

43.4.1 Organizational Controls

Organizational measures ensure that personnel, including associates and third parties, understand their responsibilities and accountabilities with respect to regulated business processes and records.

Information security training should be provided periodically to ensure personnel understand the risks to information security. Training should ensure all levels of the organization behave appropriately when working on company premises and while working remotely, e.g., from home or when traveling.

Roles and responsibilities must be clearly defined and communicated throughout the organization.

Human-resource checks should be conducted at the time of engaging new employees and associates (in accordance with defined roles and responsibilities) to identify potential risks from human behaviors (for example criminal records checks).

43.4.2 Information Security Management System (ISMS)

An ISMS (as defined by ISO 27001 [44]) should be established to define the policies, procedures, and tools to be followed to protect computerized systems data and records. Such policies and procedures include:

- Information security
- Human-resource management

- Risk management (Chapter 5)
- Information classification and handling
- Identity and access management (including starters, movers, and leavers, and periodic review of access rights)
- Segregation of duties
- Change management (Appendix O6)
- Incident and problem management (Appendix O4)
- Virus and malicious code protection
- Document and records management (Appendix M9)
- Supplier management (Appendix M2)
- Backup and restore (Appendix O9)
- Archive and retrieval (Appendix O13)
- Business continuity and disaster recovery (Appendix O10)
- Network security management
- Patch management
- Vulnerability management
- Mobile computing and teleworking
- Control and synchronization of system clocks
- Software development and maintenance (D Appendices)
- Asset management
- Data classification
- Media handling and disposal
- Information sharing and transfer
- Regulatory and geographic considerations for data privacy
- Internal audit of ISMS

Many of these policies and procedures overlap with the expectations of GxP quality systems, in particular the IT quality system. The IT quality system and ISMS may be integrated into a cohesive IT framework defining all policies and procedures required by the IT organization.

43.4.3 Physical and Technical Controls

Physical and technical controls should be implemented to minimize the risk of unauthorized or inadvertent computerized system access. Such controls include but are not limited to:

- Role-based security
- Role-based access
- Identity and access management (including multifactor authentication)
- Password management controls
- Segmentation of environments
- Encryption of backups
- Restrictions on software installation
- Anti-phishing software
- Network security
- Virtual private networks
- Data encryption (at rest and in transit based on risk)
- Key management
- Vulnerability scanning
- Intrusion detection and prevention
- Penetration testing
- Firewall and perimeter detection and prevention
- Endpoint device protection
- Physical access controls to facilities (e.g., server rooms and data centers, server cabinets, controlled spaces such as manufacturing suite, laboratories)
- Logical segmentation
- Guest and Wi-Fi networks
- Clear screen and desk policy

43.4.4 Monitoring and Incident Management

Security threats should be monitored, especially when computerized systems are externally facing. Automated and manual monitoring processes should consider (as appropriate):

- Intrusion detection
- Security vulnerabilities
- Software and hardware failures
- Unauthorized configuration changes
- Resource utilization
- Network failures
- Periodic review of user access

Where possible, monitoring tools should automatically alert the information security organization and raise incident tickets to facilitate the investigation and mitigation of potential risks.

43.4.5 Cloud Environments

Cloud environments are continuously at risk from sophisticated cyber threats that could lead to data breaches, data loss, and / or service disruption. A robust cloud-security framework is a shared responsibility between infrastructure, platform, and application providers.

Information security considerations should cover:

- Human resources
- Data protection/location
- Application security
- Configuration baseline monitoring (changes to configuration and Infrastructure code)
- Operating system security
- Physical and virtual networks
- Servers and storage
- Monitoring of cloud suppliers

Cloud architectures should incorporate physical and logical segregation of computing devices, services, and data to minimize the risk of security threats impacting the whole environment and multiple user bases.

Centralized management of information security policies ensures consistent application of up-to-date controls across the cloud environment.

Application Programming Interfaces (API) provide a gateway to applications, data, and services. API design must include robust security provisions to minimize the risk of unauthorized access.

Security provisions should include protection against denial-of-service attacks.

Regulated companies should only utilize trusted service providers and should ensure appropriate assessment of security provisions.

Cloud service providers offer multiple SLA tiers. The selected tier should provide required security provisions including monitoring.

DR plans (see Appendix O10) should ensure that the responsibilities of infrastructure, platform, and application providers are clearly defined and coordinated during a major incident.

Backup and restore operations, as a component of DR, should be in place and verified.

43.4.6 Auditing

The internal audit program, based on risk, should include assessment of organizational, process, physical, and technical controls.

43.4.7 Management Review

Senior management should periodically review the effectiveness and information security controls, ensuring that appropriate investment and resources are made available to ensure GxP records are protected.

43.4.8 Security Patching and GxP Compliance

Security patches (see Appendix S4) and fixes are released by operating system and application suppliers on a regular basis. Security patches should be applied based on risk. Typically, security patches resolve security vulnerabilities or defects. The risk that such patches impact business process-related functionality is low and therefore the impact on the validated status of the computerized system is low. Certain approaches may be applied to evaluate security patches prior to release to the production environment. These include:

- Risk assessment of patch release
- Release of patches to development and test/validation environments prior to release to production environments
- Release of patches to non GxP environments before GxP environments

43.4.9 Security Certification Programs

Third party independently certified security certifications such as ISO27001 Information security management systems [44], ISO/IEC 27017 Information technology [46], ISO/IEC 27701 Security techniques [82], and AICPA SOC 1®, SOC 2®, and SOC 3® reports [40] can be a useful benchmark for service providers to evaluate the efficacy of their ISMS.

Similarly, when procuring cloud services such certifications can be leveraged by the regulated company to provide a level of assurance that their cloud service provider has a systemic approach to security management. It is critical however, that the boundaries of ISO certifications [7] be fully disclosed and understood, and in the case of the use of a SOC attestation report [40], that the regulated company is comfortable with, or has provisions for, any management assertions noted in the report.

44 Appendix O12 – System Administration

44.1 Introduction

System administration involves routine management and support of systems to ensure that they are running efficiently and effectively. This may include database administration activities.

System administration activities may be conducted by multiple roles including system owner, IT, and/or system administrator. Further, system administration may be the responsibility of a service provider e.g., IaaS, PaaS, and/or SaaS.

44.1.1 Changes from GAMP 5 First Edition

This appendix has been revised to expand the description of system administration activities and link to security and performance management.

44.2 Key Requirements

Support processes should be established, and appropriate resource made available before a computerized system becomes operational.

System administration tasks should be identified, documented, and supported by controlling procedures. System administrators should be trained to perform these tasks and evidence of their competency retained. System administration duties generally should be segregated from operational processing duties.

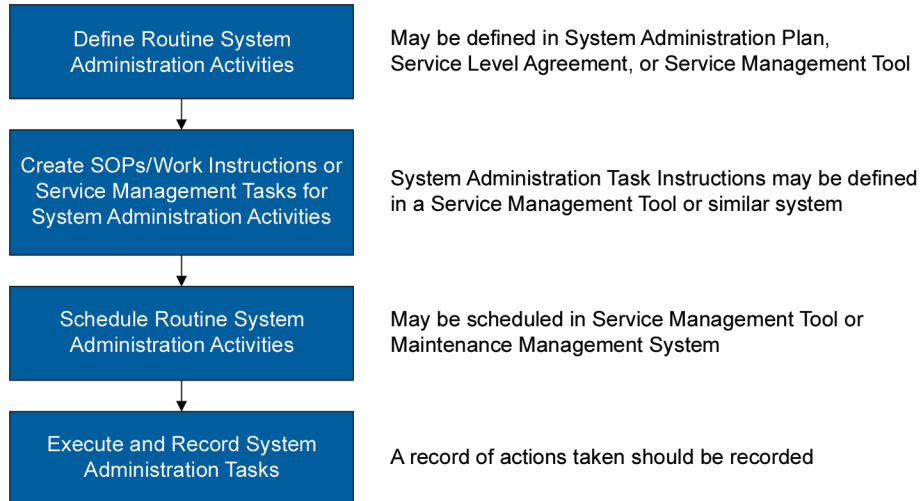
Any activities relating to the system that are not covered by standard administration procedures should be subject to operational change and configuration management.

System administration activities often require an elevated level of system access. Administrator privileges should be assigned using the principle of least privilege and should only be assigned to a limited number of people required to carry out system administration tasks. Where system administration activities are conducted by service providers, SLAs, contracts, or similar agreements should be in place to ensure that service providers also adopt these principles.

These activities should be documented.

44.3 Process

Figure 44.1



44.4 Guidance

44.4.1 General Approach

System administration tasks are more commonly managed in IT tools that define the instructions to execute the system administration tasks and the information records associated with the tasks. Similarly, such tasks may be included in SOPs, work instructions, SLAs, or system administration plans.

Typical system administration tasks include:

- User and administration account management and review
- System housekeeping (for example file cleanup)
- Backup and restoration
- System repair (like-for-like or qualified like-for-kind replacements – see Appendix O6)
- Nonfunctional static data management
- Monitoring system logs
- Database tuning and capacity management

Some security and performance monitoring (see Appendices O3 and O11) may be addressed by system administration.

44.4.2 Responsibilities

System Owner:

- Accountable overall for ensuring that processes and procedures are in place to ensure the system is used and maintained in a compliant manner.
- Ensures that any tasks delegated to the system administrator or other roles are clearly identified and documented.

System Administrator:

- Develops sufficiently detailed procedures/work instructions relating to system administration tasks and ensures that such tasks are executed in compliance with the procedure(s) and that the required records are established and maintained.

45 Appendix O13 – Archiving and Retrieval

45.1 Introduction

Archiving is the process of moving records/data from the computerized system to a different location or system, often protecting them against further changes. Archived records should be readily retrievable for business or regulatory purposes. Use of cloud storage solutions for archived records is acceptable.

Archiving and retrieval should not be confused with backup and restore processes, which are covered in Appendix O9.

45.1.1 Changes from GAMP 5 First Edition

- Enhanced process definition
- Address current archiving technologies and approaches

45.2 Key Requirements

GxP records and data should be secured by physical and/or logical means against unauthorized or inadvertent access, alteration, or deletion throughout the required retention period. Archiving approaches should be considered during the design of the computerized system.

The archiving strategy should consider how the GxP records might be accessed and used during the records retention period. This will determine how the records are stored, accessed, and retrieved during the retention period.

Archiving processes should ensure that record content and meaning are preserved, including the preservation of electronic signature and audit trail information and other metadata required to understand the record.

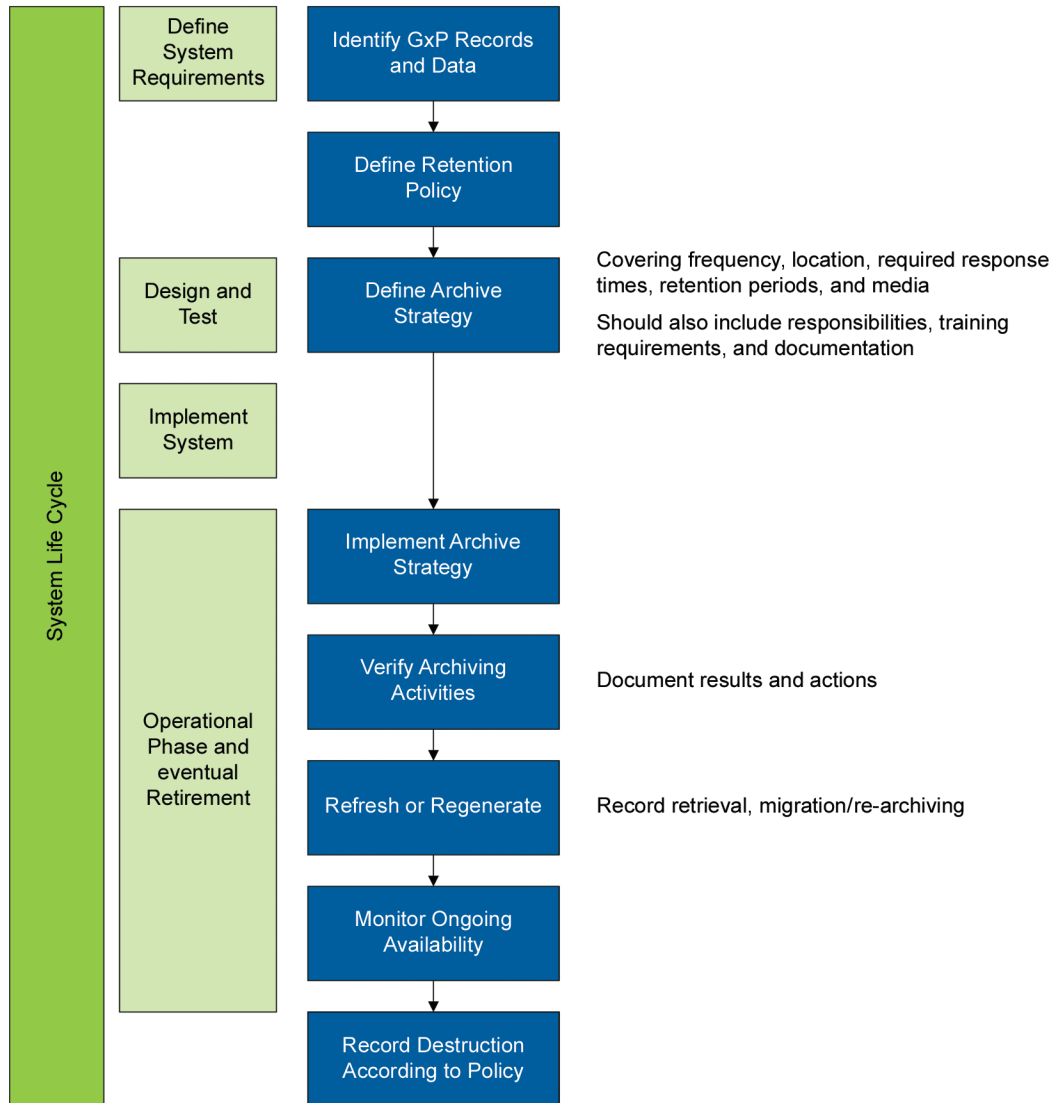
Roles, responsibilities, procedures, and specific regulatory requirements for archiving and retrieving should be defined. Certain regulations, e.g., GLP, require a defined archivist role.

Stored records and data should be initially and periodically checked for accessibility, durability, readability, and completeness.

Regulators should have reasonable access to archived GxP records during an inspection and within a reasonable time.

45.3 Process

Figure 45.1



45.4 Guidance

45.4.1 General Approach

Archiving processes should be documented and verified to ensure that record content and meaning, including electronic signature and audit trail information (where appropriate) and other metadata (where relevant), are preserved.

There should be a documented approach to managing archived records that ensures records can be accessed throughout the retention period. The approach should consider the impact of application, database, and technology upgrades on the ability to retrieve electronic records in an accessible, accurate, and complete way.

The frequency of the periodic checks for accessibility, durability, and completeness should be determined by risk assessment, taking into consideration file/record formats, archive media, and method of access.

The archiving process should pay particular attention to metadata related to electronic signatures, ensuring that the electronic signature authentication and attributability is maintained.

45.4.2 Responsibilities

Process Owner:

- Ensures that appropriate archiving procedures and technologies are in place

Quality Unit:

- Ensures that procedures are followed

System Owner:

- Maintains the systems needed to access the records

45.4.3 Policies and Strategy

The regulated company should have an established, documented policy on record retention that defines the types of records to be retained along with format and retention period for each record type.

An archive strategy is recommended that sets out the requirements and how they should be met. The strategy document can be applied to an organization, site, department, or an individual Electronic Data Archive (EDA).

45.4.4 Archival and Retention

A Data Archiving Plan (DAP) should be created to define:

- Roles and responsibilities including external organizations
- Triggers for data archiving (e.g., event driven (storage capacity) or frequency driven)
- Archiving process
- Scope of records to be archived
- Archiving tools
- Archive media and location
- Archived record searchability
- Verification approach (initial and periodic)
- Retrieval approach
- Synchronization of archived records with system upgrades

Where third parties support the archiving process, for example the storage of archived records, the need to conduct a supplier assessment and ongoing monitoring should be considered based on a documented risk assessment.

Facility controls and environmental conditions should be defined to minimize the risk of storage media degradation. The use of fire-resistant and off-site storage should be considered based on risk. Archives should be geographically separated to minimize the risk of a disaster scenario impacting all archive copies. Data privacy requirements should be considered when determining the geographic location of archived records.

Archived records should be searchable, both individually and collectively, to facilitate ready access and retrieval when required.

Records may be archived to alternative media, appliances, and/or the cloud. The archive solution should be evaluated to determine the need for archive management controls such as periodic archive refresh, duplicate copies, etc. Where records are archived to the cloud, SLAs, contracts, or other agreements should be in place to define the required retention period, restoration requirements, and data integrity and protection requirements.

The destruction of archived records at the end of the retention period should be authorized by the data owner, quality unit, and where appropriate, the legal department. Records should be destroyed at the end of the retention period using appropriate secure destruction methods unless a legal hold prevents destruction.

Where the archiving process is automated, the system should be specified and verified as being fit for intended use. Specifically, the automated process should:

- Ensure records are archived in accordance with their schedule or based on triggers (such as exceeding storage limits)
- Ensure the completeness and accuracy of archived records
- Ensure the system and its contents are secure
- Consider the ongoing availability of the devices and software needed to access the records

45.4.5 Retrieval

Retained records should be readily retrievable for business or regulatory purposes. The retrieval process should be documented, and consideration should be given to the following:

- Authorization to access controlled records
- Ability to search for archived records
- Ability to retrieve audit trail information associated with the archived record
- Verification to ensure correct record(s) has been retrieved

45.4.6 Copying/Migrating Records to an Alternative Format

The archiving process may require records to be copied or migrated (see Appendix D7) to an alternative file format. The copying/migrating process must be documented and verified to ensure the accessibility, accuracy, completeness, and searchability of the records.

45.4.7 Retaining Existing Systems

On occasions, historical records may be retained in an existing system to support records retention. For example, following the upgrade to a new solution, only current records may be migrated to the new solution and historical records may be retained in the existing solution. In such cases, the DAP must consider:

- Securing records from any further processing
- How the system is to be maintained to ensure accessibility for the retention period
- Licensing
- Maintaining information security controls
- Retaining knowledge of how to access the system and records
- Ensuring that hardware and software are available to support the legacy application

Maintaining aging and potentially unsupported systems for managing archive records can present significant risks. Current technologies enable the migration of physical systems to virtual environments, which will significantly reduce risks. Such strategies should also carefully be planned to ensure that virtualized environments can be maintained in the long term.

45.4.8 Return of Archived Records

Archived records managed by a third party may need to be returned to the record owner, for example the return of records to a sponsor. Contracts should clearly define the required controls to ensure all records are returned in a form that enables the record owner to access, search, and where required, process records once returned.

46 Appendix S1 – Alignment with ASTM E2500

46.1 Introduction

There is a requirement in several of the GxP regulations to validate those parts of computer and control systems that are critical for the health and protection of the patient. In some cases, regulated companies have chosen to perform an inefficient process of installing and commissioning equipment, and then validating it in a separate exercise. This often resulted in wasted effort and repeated activities. Some of the documentation generated during the process did not add value in contributing to fitness for intended use. The principles and practices described in all the versions of the GAMP Guides have been aimed at cost reduction, but separate and duplicated activities are inefficient and unnecessarily costly.

New standards and guidance documents¹⁸ are emerging that aim to make the implementation process for GMP manufacturing control systems more cost effective and value-added, focusing only on those aspects of systems critical to the protection of the patient. These new standards and guidelines are based on a science- and risk-based philosophy that focuses on the risk to the patient in combination with Good Engineering Practices (GEP)¹⁹. If used correctly, the combination of:

- GEP
- A science- and risk-based approach
- Understanding of the process
- The appropriate involvement of SMEs from relevant organizational areas

These can all be combined in a streamlined effort within an overall framework that meets all regulatory requirements.

These emerging standards have influenced GAMP 5, the associated Good Practice Guides, and the terminology used, especially in the context of GMP process systems and equipment.

46.2 Focusing on Patient Risk

While GAMP 4 and previous guides provided an overall life cycle framework for systems and controlled equipment, they recognized that the practicalities are different for different system types. As a result, a series of Good Practice Guides were produced to support the understanding of these differences and provide more practical detail.

Many pharmaceutical companies undertake complex, time consuming, and expensive qualification practices. There are aspects of qualification that can add value in terms of ensuring the equipment and systems are fit for intended use, but there are other aspects that often do not add this value. Some of the prescriptive and rigid conventions and practices that surround qualification as often practiced can detract from its overall value. GMP regulations provide the basis for the activities that are called qualification, but no specific requirements that relate to how qualification is practiced.

¹⁸ From ICH [6] and ASTM [83] (including the ASTM E2500 - Standard Guide for Specification, Design, and Verification of Pharmaceutical and Biopharmaceutical Manufacturing Systems and Equipment [8]).

¹⁹ GEP is defined as those established engineering methods and standards that are applied throughout the life cycle to deliver cost effective solutions that minimize risk to the patient.

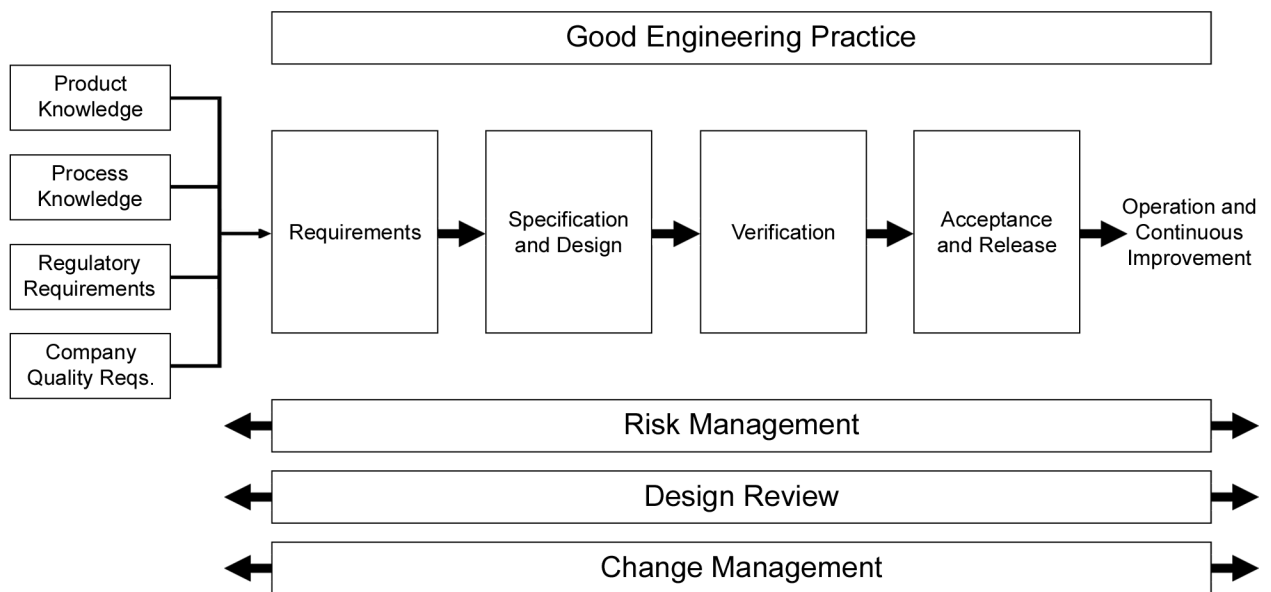
By focusing on the risk to the patient and leveraging the expertise of the supplier and SMEs based on GEP, verification is considered as a set of integrated activities that can replace the activities previously called IQ and OQ. Regulated company IQ and OQ activities may then be omitted or limited to an assessment of the supplier’s activities and documentation, and if necessary performing mitigation activities to close gaps. This eliminates much of the costly duplicated testing that does little or nothing to protect the patient. Finally, the overall performance and fitness for intended purpose can be ensured through Performance Qualification or Verification, which focuses on critical-to-quality attributes. Overall, this will demonstrate that the equipment or system is performing satisfactorily for its intended purpose, the process with which it is involved is controlled, and the risks to the patient have been effectively managed, thus meeting the regulatory requirement for validation.

It is important to select the right tool for a specific need, such as design review, inspection, or testing (e.g., Commissioning, Qualification, IQ, or OQ). The term verification is used in ASTM E2500 [8] and aims to promote flexibility in choosing the right approach. (See Figure 46.1.)

A science- and risk-based approach is inherent in the thinking behind verification, where the level and extent of verification is based on scientifically assessed risk to the patient from specific processes, equipment, and systems. This is directly in line with the principles described in ICH Q8 [84], Q9 [14] and Q10 [21] documents for the development, quality risk management, and quality management of pharmaceutical products throughout their life cycle.

Figure 46.1: The Specification, Design, and Verification Process [8]

Reprinted with permission from ASTM E2500-20 Standard Guide for Specification, Design, and Verification of Pharmaceutical and Biopharmaceutical Manufacturing Systems and Equipment, copyright ASTM International, 100 Barr Harbor Dr., West Conshohocken, PA 19428. A copy of the complete standard may be obtained from ASTM at www.astm.org.



It is, of course, still appropriate to create a plan describing and justifying the approach taken to ensure the equipment is fit for use in a GxP regulated environment, and to have a report available providing the necessary evidence to support this claim.

Performed in this way, the process for the specification, design, and verification of controlled process equipment meets all GxP regulatory expectations.

46.3 Different Types of Computerized Systems

For integrated manufacturing systems or equipment where a computer-based system is part of the overall functionality, a specific and separate computerized system validation may not be required.

For example, where the computer-controlled equipment can be regarded as one component of a wider manufacturing or process control system the verification can be an integrated part of the overall process validation effort. The verification of fitness for intended use may be adequately demonstrated by documented integrated engineering or project activities together with subsequent process validation, and the overall approach may be defined based on each regulated company's policies and preferences.

Validation is the common term used in regulations worldwide to describe a process that demonstrates that systems are fit for intended use. Some computerized systems are intimately involved in many regulated business activities outside the manufacturing area and are critical for the health and protection of the patient. Examples include the collection of clinical trials data, the management of donor details in blood collection, the recording of adverse events and complaints, the release of product for sale, and the recall of defective product.

Such IT systems have no direct correlation with the manufacturing and release of the product. Consequently, there is no direct parallel with the manufacturing process and associated process validation. Acceptance of the system is dependent on the satisfactory completion of a functional test, such as the traditional OQ or equivalent tests, prior to a controlled cut over into the live environment. (Some further testing, e.g., stress or performance testing, may be necessary which some organizations call PQ but it is not an activity parallel to the PQ testing of controlled process equipment.)

The principles described in ASTM E2500 [8] should be interpreted with attention to the special characteristics of particular systems, and suitable verification that the critical-to-quality requirements of the system have been met should be completed before the computer system can be approved for use in a GxP regulated environment.

The ideas that led to the development of ASTM E2500 [8] are applicable to all computerized systems. GAMP 5 has tried to describe a process that follows the same principles:

- The requirements of the system should be clearly defined
- Requirements critical to the health and protection of the patient (critical-to-quality requirements) have been identified and the risks identified and controlled
- The principles of GEP are applied throughout
- Testing carried out and documented by the supplier should be leveraged as much as possible
- The critical-to-quality requirements are appropriately verified and reported by the regulated organization in line with regulatory expectations

It is also recommended that a plan describing and justifying the approach taken, and a report supporting the claim that the system is fit for intended use, are created.

Performed in this way, the process described above for computerized systems meets all the GxP regulatory expectations for validation.

46.4 Terminology

Since GAMP 5 covers both systems involved in manufacturing of pharmaceuticals and systems for other critical types of IT applications, this Guide uses terminology that enables appropriate selection of the relevant life cycle activities, depending on the specific context.

Some organizations have already taken the decision to adopt the term “verification” and apply it to both computer and control systems. Others have indicated that they will stay with the word “qualification” but adopt the principles described in the ASTM E2500 [8]. Still others have changed to verification for controlled process equipment but retained “qualification” for computer systems.

The GAMP Community of Practice aims to strongly support and promote innovation. GAMP Guidance is neither mandatory, nor prescriptive, but aims at enabling innovation in a compliant and cost effective manner.

Descriptions of current industry practices should not be read as constraining in any way the development and adoption of other approaches. Individual companies should and will decide what terms and precise approach they will use.

GAMP 5, like previous versions of GAMP, is a guide that supports good quality management practices. The enhanced focus on science and the increased focus on risk to the patient are important to the future of the pharmaceutical industry. GAMP will continue to support evolving good practices for the pharmaceutical industry at large, its regulators, and suppliers.

47 Appendix S2 – Electronic Production Records

47.1 Introduction

This appendix describes concepts and high-level descriptions of approaches and technologies that reflect what has been observed to be best practice when determining requirements for Electronic Production Records (EPRs) related to manufacturing operations.

Principles are presented to facilitate the implementation and use of EPRs in GxP environments supporting quality processes required for:

- Review by Exception (RBE)
- Real-time disposition
- Cloud technology
- Blockchain technology

Software design principles and regulatory requirements provide a framework to help define activities and methodologies to ensure the integrity of data across integrated computer systems' functionality in a manufacturing domain. Formal data definitions including purpose, format, and usage are critical to the establishment of system designs, policies, and procedures ensuring the accuracy of data in the domain.

- Technologies, including cloud computing and blockchain, allow for efficiencies and security enhancements for life sciences manufacturing to help meet regulatory requirements and ensure patient safety.
- Generation of reports for RBE and other functionality from EPR can be performed in real time or after completion of operations for supervision and review of process execution.
- The same design and implementation principles may be applied to electronic records portions of hybrid paper/electronic systems.

ANSI-ISA-88.00.01-2010 [85] provides a basis for establishing electronic systems terminology equivalent to paper-based batch record systems. This and additional terminology supporting the implementation of EPR is defined further in Appendix G2.

This appendix will utilize ISA [86] terminology as the model to represent batch, assembly, and continuous manufacturing paradigms such as Master Recipe and Control Recipe versus Master and Production Batch Records. Various industry guidance and/or regulatory documentation may reference different terminology, such as batch production and control records (FDA [87]), batch processing record (EU [88]), master production records (FDA [89]), and manufacturing formula and processing instructions (EU [88]). These are represented by equivalents in the ISA model independent of region or origin.

A clear distinction is made between EPRs and reports generated from such data for review, disposition, investigation, and process performance or improvement. Table 47.1 shows paper and ANSI-ISA 88 [85] terminology equivalents.

Table 47.1: Paper and Electronic Record System Equivalents [90]

Paper System	Electronic System Equivalent Structure	Description
Master Batch Record (MBR)	Master Recipe	Contains product name or designation, recipe designation or version, formulas, equipment requirements or classes, sequence of activities, procedures, normalized bill of materials (quantity per unit volume to produce)
	Work Instructions (optional)	Additional detailed instructions – may include electronic SOPs or SOP references
	Critical Process Parameters (optional)	Required Process Parameters that are to be checked or monitored or are to be downloaded to other systems such as automation
Production Batch Record	Control Recipe	<ul style="list-style-type: none"> • A Master Recipe dispatched or otherwise made available in manufacturing-related areas for execution • Includes Master Recipe information with the addition of schedule, specific quantity to make, actual target bill of materials quantities, and other data for the batch and production instance
	Electronic Production Record	<ul style="list-style-type: none"> • A store of data and information created by systems or entered by personnel during execution of Control Recipes • May be located in one or more systems or databases • Data may or may not be stored in human-readable format
	Batch (Production) Report	Data and information in human-readable format, presented either in electronic or paper format for activities, such as review, disposition, investigation, audit, and analysis

47.1.1 Changes from GAMP 5 First Edition

Technologies have significantly evolved since GAMP 5 was first published and allow for efficiencies and security enhancements within life sciences manufacturing. They are addressed as appropriate in this significantly revised appendix:

- Greater adoption of cloud computing technologies
- Usage of blockchain technology
- Real-time generation of reports for RBE and other functionality from EPR
- Clarification on data audit trails and data audit trail review

47.2 Considerations for Electronic Production Records

47.2.1 GxP-Relevant Data

Activities such as a strategic assessment of the current and desired future state of operations, and standard design and review cycles should determine what data is relevant for the intended use of each system based on identification of CPPs and CQAs of materials, products, assemblies, and intermediate stage production output. Data impacting GxP production and decision-making may be generated by non-process systems with at least some functionality supplying or processing GxP-relevant data. Such GxP-supporting systems functionality should be verified.

Systems must be capable of reconstruction of GxP-relevant data in appropriate formats for evaluation. The following are examples of information to be defined:

- Production record data requirements (GxP and non-GxP)
- Data Audit trails
- Methods to manage a multiple language user base across regions
- External documents that are linked or otherwise made available to systems
- Security, user rights, and roles
- Master data, metadata, and specifications
- Methods for time zone, time stamp recording, and reconciliation

47.2.2 Audit Trail Terminology

This Guide uses the same terminology as the *ISPE GAMP Guide: Records and Data Integrity* [35] when referring to data audit trails as clearly distinguished from system technical logs and other event logs.

Data audit trails, as required by various regulations [91, 32, 75, 92], record operator actions that create, modify, or delete GMP records during normal operation, and should be clearly distinguished from other system and technical logs.

In a traditional paper-based system, if a user recognizes that a specific data entry is wrong, they strike out the wrong data in a way that it is still readable and put the correct value next to it with their initials, date, and in some cases, the reason.

In an electronic system, the data audit trail provides equivalent traceability when users create, modify, or delete GMP records and data as part of normal operations by recording the original and new values, the user, the time stamp, and in some cases, the reason.

In a paper-based manufacturing system, modifications made to the production batch record during normal operation are traced manually while the master batch record and SOPs are likely controlled via a separate (document management) system. Note that changes to approval status are traceable within the record, and that destruction of the approved production batch record at the end of the retention period is also traceable through entries in an indexing system.

In the equivalent electronic manufacturing system, the data audit trail captures modifications that are part of normal operations. It also tracks activities that change the status of a record (e.g., applying an electronic approval signature) and to deletion of the production record at the end of the retention period. [93]

47.2.3 Data Audit Trail Review

Audit trail review should be part of the routine data review and approval process usually performed by the operational area which has generated the data, such as manufacturing or the laboratory. There may be also specific GxP requirements for quality unit review. Data audit trail review is described and explained in regulatory guidance from FDA [94] and MHRA [37]:

“Who should review audit trails?”

Audit trail review is similar to assessing cross-outs on paper when reviewing data. Personnel responsible for record review under CGMP should review the audit trails that capture changes to data associated with the record

as they review the rest of the record (e.g., §§ 211.22(a), 211.101(c) and (d), 211.103, 211.182, 211.186(a), 211.192, 211.194(a)(8), and 212.20(d)). For example, all production and control records, which includes audit trails, must be reviewed and approved by the quality unit (§ 211.192). The regulations provide flexibility to have some activities reviewed by a person directly supervising or checking information (e.g., § 211.188). FDA recommends a quality system approach to implementing oversight and review of CGMP records.”

“How often should audit trails be reviewed?”

If the review frequency for the data is specified in CGMP regulations, adhere to that frequency for the audit trail review. For example, § 211.188(b) requires review after each significant step in manufacture, processing, packing, or holding, and § 211.22 requires data review before batch release. In these cases, you would apply the same review frequency for the audit trail.”

“If the review frequency for the data is not specified in CGMP regulations, you should determine the review frequency for the audit trail using knowledge of your processes and risk assessment tools. The risk assessment should include evaluation of data criticality, control mechanisms, and impact on product quality.”

“The relevance of data retained in audit trails should be considered by the organisation to permit robust data review/verification. It is not necessary for audit trail review to include every system activity (e.g. user log on/off, keystrokes etc.).”

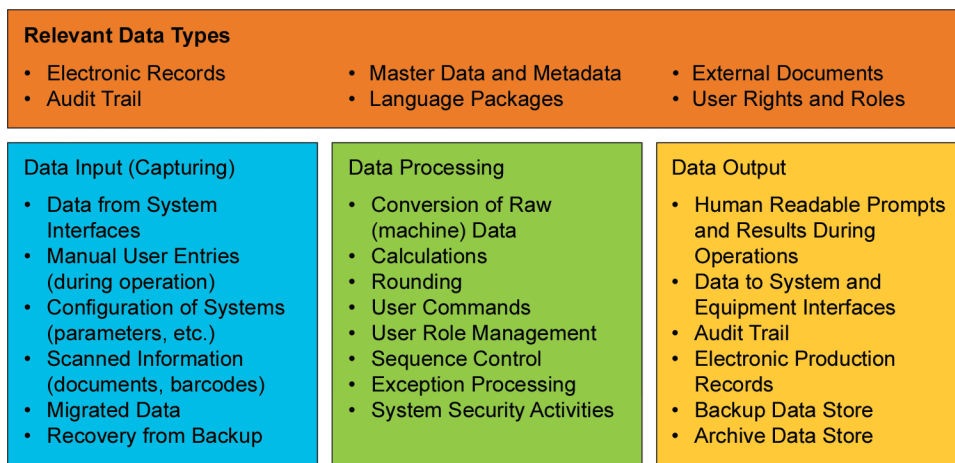
“Routine data review should include a documented audit trail review where this is determined by a risk assessment. When designing a system for review of audit trails, this may be limited to those with GXP relevance. Audit trails may be reviewed as a list of relevant data, or by an ‘exception reporting’ process. An exception report is a validated search tool that identifies and documents predetermined ‘abnormal’ data or actions, that require further attention or investigation by the data reviewer.

Reviewers should have sufficient knowledge and system access to review relevant audit trails, raw data and metadata...”

47.2.4 EPR Data Types

Design and implementation of computer systems requires an understanding of the various types of data to be created and managed. Exercises such as data mapping help define these types. Figure 47.1 provides example types and related data processing actions.

Figure 47.1: Scope of EPR Data Types



ALCOA+ principles should be applied to EPR data.

Systems design and/or procedural controls should ensure that the versions of all master data and system configurations are known and controlled.

Within an EPR, only the records and data required by applicable regulations are considered GxP electronic records.

Other information such as trends and warnings recorded as an output of manufacturing are often used by production and quality personnel to determine long-term effects of operational tolerances and variances, but are not part of GxP production records unless directly related to GxP decision-making.

The life cycle of manufacturing and process data and records, and data integrity considerations for such records are described in detail in *ISPE GAMP RDI Good Practice Guide: Data Integrity – Manufacturing Records* [93].

Along the manufacturing process numerous data relevant for an EPR is input to and output from all electronic systems with functionality in the manufacturing domain as described in the *ISPE GAMP Good Practice Guide: Manufacturing Execution Systems – A Strategic and Program Management Approach*, Chapter 7 [90]. Systems with functionality within the manufacturing domain may include:

- Document management system
- Supply chain management system
- Manufacturing execution system
- Data historian
- Enterprise resource planning
- Process control system (supervisory and automation/controls)
- Laboratory information management system

Computerized systems across the manufacturing domain may generate or transfer source data for master and control recipes, as well as serve as a repository for part or all of an EPR as required by designs and technology in a given implementation. Data can be made available by means of interfaces for further human or machine review of in-process or post-production activities and results.

Operational processing may have master data such as material specifications, process parameters, alert and alarm limits, or process step sequences controlled by several systems with functionality in the manufacturing domain (see Figure 47.1). Recipes may combine master data from one or more sources either by direct entry or by links to systems for the production environment for execution. Systems design and/or procedural controls should ensure that the version of all master data is known and controlled and can be demonstrated for any specific master recipe.

Particularly within the manufacturing execution layer, data flows can be complex and require a detailed data-flow mapping exercise to identify what data is required for each business process in addition to critical parameters and aspects. To enable fit-for-purpose process maps, a business analysis team uses appropriate techniques to interview operational staff. (See *ISPE GAMP Good Practice Guide: Manufacturing Execution Systems – A Strategic and Program Management Approach*, Chapter 6 [90] for strategic assessment methods.) Designed process steps and system workflow mapping sessions typically include data specialists to ensure comprehension of data challenges and where the data will be exchanged through multiple platforms within future system architecture. Where data flows result in duplication of data sets, definition of where the primary records reside is required.

47.2.5 Data Capturing/Data Input

Execution of manufacturing-related processes generate data and records that may reside in, or be linked to, multiple systems.

Creation of a master recipe based on product processes and environmental requirements typically requires input from multiple data sources associated with manufacturing systems functionality. These would be regarded as regulated records. The design and the transformation from paper to an EPR requires verification of processes used to perform the transformation and confirm equivalency of the electronic form.

Master recipes are transferred by appropriate means to production systems creating control recipes that when executed produce EPRs with content in various systems. ALCOA+ principles should be applied.

47.2.6 Data Processing

Based on established CPPs and CQAs, key factors in processing data include verified rounding rules and other mathematical standards, calculation definitions, alerts, alarms, and specific events that may automatically create data or initiate other actions or further processing. Data audit trails and procedures for data review, (including audit trail reviews where relevant), is essential for process management, review and improvement, and investigations. Appropriate and effective security features, user management, and privilege management is essential.

47.2.7 Data Output

At any desired point, reports may be generated for operator or quality approval from live data or the EPR as a check point to facilitate operator or system intervention, continuation of processing, or production reports. Systems such as ERP or LIMS may set the status of materials (output) and provide the results (output) to other systems.

Verified methods must be in place to generate reports from EPRs to present all required information accurately in timely, human-readable form to facilitate disposition, investigation, and other GxP activities.

EPRs often contain data for efficiency and other process indicators that are non-GxP. Such reports should be verified for engineering or other non-GxP purposes.

An important aspect of data output are interfaces to adjacent systems. Interfaces require specific attention to design, development, and verification. Early involvement of interface partners is recommended in order to move from simulators to actual interfaces before entering formal verification stages.

47.3 Understanding Manufacturing, Production Processes, and Data Flow

The processes and data flows within the manufacturing and production areas within a life-sciences organization should be understood. It may be necessary to periodically review (or document and develop if not in place) in order to provide a detailed understanding of the process/data flows and associated controls through the processes. Critical thinking should be used to determine critical/important records/data. Associated systems should be formally assessed for data integrity gaps. Any gaps found should subsequently be assessed for risk and remediation actions determined by a panel of business, data, and system SMEs and owners.

Gemba walks are a way of gaining knowledge of the processes and linked critical records data. Gemba is the action of going to see the actual process:

- Understand the work being conducted, ask questions, show respect to the SMEs, and learn the business
- Do the same with the data structures

- Do the same with the data along the data life cycle
- Do the same with the actual system usage on the shop floor

47.4 Completion of Production Activities

EPRs residing in various data repositories are processed by verified means combining data from all sources to create reports (output) for automated or human review/disposition of assemblies, intermediates, and products as necessary. These reports are the equivalent of what was formerly a paper batch record; however, reports may be generated for presentation in either electronic or physical media form. Reports may be divided as desired into focused sections for review by various departmental personnel responsible for each part of process execution/output. An overall review/disposition of the complete report is performed to verify completion of each subsection review.

Two methodologies may be used to facilitate review and approval/disposition activities by automated functions, human review, or a combination thereof: RBE and real-time disposition.

47.4.1 Review by Exception

RBE may also be known by other terminology such as focused review or limited review. Data from manufacturing-related operations is screened by verified systems functionality to create reports for review that include all critical process exceptions and information, and reduce or eliminate the need for reviewing in-tolerance data, trends, and information.

The RBE method:

- Does not eliminate or change data in EPRs
- Is a verified extension of existing systems report-generation functionality
- Filters EPR data presented to personnel
 - Includes human process/system interaction such as disposition and alarm processing
 - Includes any critical exceptions or deviations to the process whether mechanized, computer related, or human instigated such as exceptions to CPPs or CQAs
 - Reduces or eliminates reporting in-tolerance operational data, events, or alerts not required to support critical exceptions
- Is applied to well-understood processes or portions of processes
- May be implemented in electronic or hybrid systems

The concepts underlying RBE have been part of the regulatory environment for several decades. However, regulatory authorities will review and determine the acceptability of a company's RBE strategy.

The goal of RBE is to provide risk-based efficient manufacturing and quality review processes by dividing data review operations between human and systems functionality, leveraging the comparative strengths of each.

Modern automation systems generate very large volumes of data, and a practical human review can only be based on statistical methods such as data samples, averages, or other processed summaries such as trend graphs.

Detailed review by personnel of all data from validated computerized systems is considered a redundant operation that diverts human-review efforts from more critical disposition activities.

47.4.2 Implementation of RBE

Implementation of RBE is based upon the following requirements:

- System functionality, accuracy, and reliability are clearly defined
- Processes or process steps subject to RBE are well-defined
- Operational experience with processes and process steps is established
- Novel processes or process steps may require parallel generation and comparison of more complete reports with exception reports to verify accurate filtering of data and information
- A risk-based analysis determines the complexity and length of exception report verification efforts
- Data that would be manually reviewed in non-RBE type reports is retained and can be presented in human-readable form for the appropriate retention period
- The computerized means of review is at least as comprehensive and accurate as a manual review
- RBE functionality is appropriately specified, verified, and periodically reviewed as part of configuration management
- Communications or other systems errors that could prevent the report of a critical exception are alerted and noted in an RBE report or otherwise made available to reviewers
- When no critical exceptions occur during operations, the associated exception report indicates that operations were completed without error. A report is always generated and examined by appropriate approvers even when no exceptions occur.

RBE may be implemented in a phased approach until complete process steps or processes use the method.

- This approach provides RBE reports for segments of processes or equipment for which there is no current automated exception analysis and reporting capability
- Individual RBE reports may be used in conjunction with other reports where appropriate procedural and technical controls define methods to manage the overall review process

RBE can be applied to the execution of product recipes generating logical or physical inventory such as intermediate, product, sub-assembly or device, and non-product recipes such as asset preparation, which may include equipment selection, setup, cleaning, and sterilizing.

Exception instances in reports should include sufficient contextual information to allow reviewers to procedurally or automatically retrieve data associated with each exception to support investigations and other activities. This may include actual data, links, or references to data sources.

RBE is enabled by the GAMP approach, where systems are appropriately specified and verified to ensure CPPs and overall systems operations are implemented correctly, and are appropriate to each process, process step, or system function. Following the GAMP approach should ensure the following:

- Processes are maintained within predefined tolerances
- Data and events are accurately recorded contemporaneously

- Process data is monitored and checked at appropriate rates for processes
- All defined process or system alerts and alarms are generated when tolerances or other operating constraints are exceeded
- Electronic records are accurate, trustworthy, secure, and available in a timely fashion
- Production reports for process/product review are demonstrated to be accurate

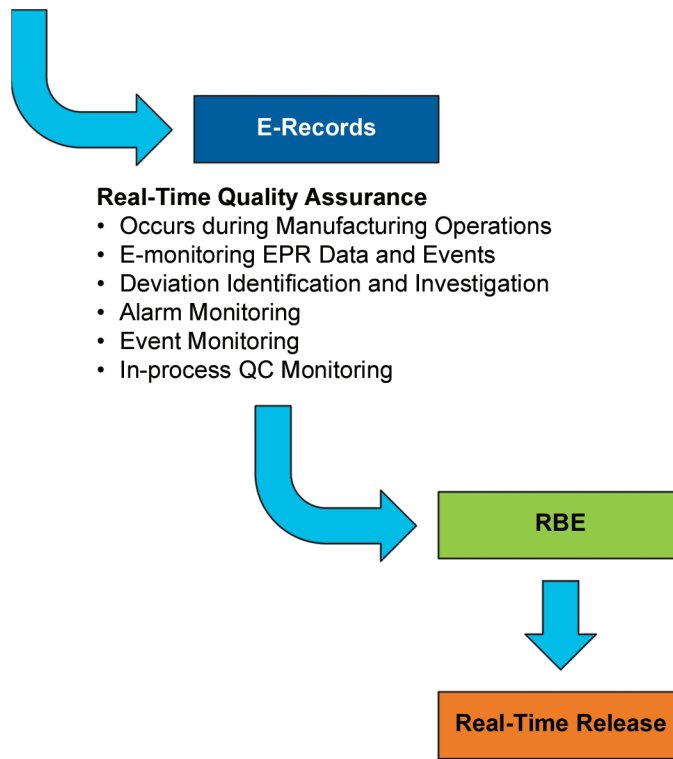
47.5 Migration to Real-Time Disposition

With the implementation of EPRs and RBE, quality assurance and quality control activities can be performed at time intervals close to actual process execution. These activities can be performed either at the physical process location or remotely via verified interfaces and approved activities as appropriate. An example of an implementation migration is shown in Figure 47.2.

Figure 47.2: Example of Migrating to Electronic Records and RBE

Electronic Control Recipe(s)

- Raw Materials (incoming inspection)
- Weigh and Dispense
- In-Process Inspection
- Operator Work Instructions
- Environmental Monitoring and Control
- Process Monitoring and Control
- Product Inspection



Automated systems may be designed with programmed control logic to determine if results from operations meet criteria for advancing to next steps either with or without human approval as defined in process definitions. Where human intervention is required, necessary real-time production record contents are made available to required personnel for disposition or approval to facilitate efficient continuation of manufacturing processes. For example, there is a regulatory requirement for human intervention in EU GMP for a Qualified Person to perform batch certification as described in EU GMP Annex 16 [95].

47.6 Technology and Architecture Considerations

47.6.1 Cloud Computing

Migrating process control technologies to utilize cloud services requires additional analysis to determine where the risk of interrupting or delaying real-time process execution may be encountered with remotely hosted applications and/or databases. Mapping of data flows is a prerequisite to understanding the potential threats to data integrity, the output of which will be used both in defining user requirements and in data integrity risk management.

Selection of cloud service providers includes the determination of adequate protection against unintended data/application access, exporting, or sharing.

Two cloud paradigms are possible for cloud implementation: central cloud and edge cloud. Edge cloud indicates the service provider is physically closer with shorter data connections to the customer/user. When utilizing cloud services for hosting systems critical to real-time process automation, using edge cloud may mitigate risks of data or software operation interruptions, while improving response times for data transfer and user interface activities.

Regardless of the cloud configuration, the end user organization is responsible for procedural and electronic controls to ensure that its cloud-based systems are reliable and minimize risk. To maintain the validated state, the end user's transfer and updates of information among local and remote systems must be accurate, monitored, and appropriately documented. The end user is always ultimately responsible for determining and maintaining the validated state of the systems and services they utilize, regardless of whether or not they are implemented in a cloud.

Cyber security implications and vulnerabilities should also be considered. Global organizations typically have cybersecurity measures in place, yet occasional large data breaches still occur. Denial of service attacks may not alter data but can cause delays or interruptions in access and communications between local and remotely hosted systems. Cloud computing may have the potential to increase security risks, although properly applied network security methods can mitigate most of them.

47.6.2 Blockchain

Blockchain technology uses a distributed computer network to create a digital ledger acting as a virtual database storing uniquely identified transaction records. By definition, every network server or node assigned to a blockchain verifies each data entry and archives all transactions that have been recorded. Transaction data stored in a blockchain is not kept in a central repository but distributed across many designated nodes that may be local to the production environment or widely dispersed geographically. The distributed nature of blockchain ensures records are immutable. This effectively defines a write once and read only data storage implementation.

Candidates for blockchain implementations include audit trails as these records must provide accurate confirmation of human activities for company and regulatory agency review for all computer systems with functionality in the manufacturing domain.

As part of risk-based data integrity design and assessments, it may be appropriate to utilize blockchain technology for some subsets of EPR data associated with QbD methodologies based on risk-mitigation efforts.

48 Appendix S3 – End User Applications Including Spreadsheets

48.1 Introduction

This appendix gives guidance on the use of end user applications such as spreadsheets, small databases, statistical applications, and other small applications in a GxP environment.

Application tools are available for creating a wide range of end user applications, including customized statistical analyses, the creation of local databases, data mining, and multivariate analysis. These may be used for GxP regulated activities, and they present particular compliance challenges.

Among end user applications spreadsheets tend to be the most under-documented systems used in GxP environments, for the following reasons:

- Users regard them as part of the desktop
- Users may not understand or be aware of quality procedures
- The ease with which applications can be built without much training
- The data processing power that they can have

The flexibility and power of the spreadsheet allows users to create tools that range from performing simple calculations to sophisticated analysis of a major clinical study. Special emphasis is placed on spreadsheets in this appendix because users may have the opportunity and ability to create a spreadsheet application, and may use them to process regulated data. Consideration should be given to the standard requirements to build a template. This includes standard headers and footers, font size, etc., in the creation of all spreadsheets regardless of type for use in support of GxP operations.

The level and rigor of specification and verification applied to end user applications should be based on risk, complexity, novelty, and intended use. This appendix provides guidance to help users determine the appropriate approach. While the examples given in this appendix are mainly spreadsheets, the same principles can be applied to other end user applications.

48.1.1 Changes from GAMP 5 First Edition

Additional data integrity considerations and detailed risk-based recommendations have been included.

48.2 Application Types

This section considers typical examples of end user applications found in the GxP environment.

48.2.1 Disposable Spreadsheets

Spreadsheets may be used in the same way as a hand calculator. For example, 10 output values from a laboratory test are input for the purpose of calculating a mean and standard deviation. In this scenario, the electronic copy is not retained.

This should be documented in the same way the use of a non-printing calculator would be documented, i.e., the values and result are recorded and signed.

The results can be printed, labeled, and signed. Alternatively, they may be saved to a static format and signed via an external electronic signature tool. In either case, these are now documents, and the guidance in Section 48.2.2 applies. It should be clear on the page exactly what arithmetic manipulation was done. This can be facilitated in most spreadsheet tools by printing a copy of the spreadsheet displaying the cell formulas.

Calculations used to process GxP data should be verified. This does not mean that algorithms used by native functions of the spreadsheet need to be checked for accuracy every time the sheet is run, but rather to demonstrate that they are the correct calculations during the verification stage. For example, $(a+b)^c$ is a very different expression from $a+(b^c)$, and errors like this are easily made. Verification of the calculations can be accomplished by printing the cell formulas, or by a third-party review. Such calculation verification is appropriate for any GxP spreadsheet.

48.2.2 Spreadsheets Retained as Documents

In many cases, the way in which spreadsheets are used is more like a word processing document than a traditional application. The main difference is that the spreadsheet can be used to both record GxP data and to manipulate it. The flexibility of manipulation that makes spreadsheets useful makes it advisable to manage them as documents rather than applications. It is likely to be extremely difficult to establish that all subsequent saved copies are the same as the original. Calculations should, therefore, be verified and fully explained, as they would be in a text document. This should include proof that the intended formulas have been used, as described in Section 48.2.1.

The level of risk related to data integrity should be a consideration when choosing a control strategy. There are a variety of options for achieving adequate control, such as:

- Using the spreadsheet tool's internal security options, such as password protecting cells or sheets
- Storing the spreadsheet in a secure directory
- Managing the spreadsheet in an EDMS

If the spreadsheet cannot be adequately controlled through these or other means, it may be advisable to consider a static version, for example, a secure PDF or even hardcopy, as the primary record.

Spreadsheets that are effectively documents should be managed in compliance with the applicable regulations. For example, a common use of spreadsheets is to manipulate and maintain GxP laboratory data, where compliance with electronic record and signature regulations is a particular concern.

48.2.3 Spreadsheets as Databases

Another popular use of spreadsheets is as a simple database, i.e., to manage or store GxP data electronically. In a GxP environment, this presents a risk because data may be frequently updated on spreadsheets that lack the intrinsic controls necessary to ensure data integrity. For example:

- Spreadsheets generally have limited or no capability to limit a user's ability to edit data
- Spreadsheets do not support audit trails where needed
- Every time data is added or removed from the spreadsheet and saved, and entirely new database is also saved; this means that someone adding data may inadvertently or intentionally change the database "code", and it could go unnoticed.

If a compliant solution is to be developed using a spreadsheet, external controls should be developed to overcome these shortcomings. While there are commercially available products intended to provide audit trail capability to spreadsheets, as a general rule the use of spreadsheets where audit trails are required is inadvisable.

Users should be fully aware of the limitations and weaknesses of spreadsheets as databases when proposed. A far better alternative is to use an actual database application.

48.2.4 Template Applications

A very common use of spreadsheets is the development of template solutions, where data can be subjected to a standard manipulation and the result saved as a unique document or even exported to an application. Statistical analysis or data mining applications may also fit this subtype. Templates may be used, e.g., in tabulating and processing data from a clinical study, or similarly, for calculations based on QC test results prior to product release.

When developing such templates, users and developers should fully understand and document the required manipulation. This allows clear confirmation of design intentions against standard package features to be established and confirmed. The following should also be considered:

- Calculations should be verified to be correct.
- Will the template run on a single workstation, or will it be available for download from a single location? If not, how is it ensured that everyone is using the correct version? Version control should be established, supported by an effective change management process.
- How will access to the application and data fields by users and developers be controlled? Ideally, all cells other than data entry should be locked and inaccessible to users.
- How will functionality be configured? Is there a custom script requirement when using application wizards? A macro is custom software. Even when created by keystroke capture, there is a program in a language such as Visual Basic® for Applications (VBA) behind each macro.
- Will there be more than one module? Integration testing is appropriate in such circumstances. For spreadsheets this may involve direct cell links to other worksheets. These links can be affected by changes and should be addressed as part of the change control process.
- Will data input be only via keyboard? External data feeds need configuration, and a spreadsheet may not be sophisticated enough to deal with unusual input (e.g., a character string that is too long).
- Will output be saved to a file or only printed? Electronic record controls may be necessary if the document is retained electronically.
- Will the master template be locked and stored under version control in a separate location from daily use?
- How is the result secured to protect data integrity?
 - Should the forms be automatically numbered when downloaded?
 - Is the completed form protected from further editing?
 - Is there protection to ensure that the form cannot be deleted and replaced?

48.2.5 True Desktop Databases

Both proprietary and open-source desktop databases offer superior solutions to managing large volumes of data compared to spreadsheets, but they still are often significantly less secure than more sophisticated database management systems developed to run in IT-managed server-based environments (e.g., Oracle®). This may present significant issues if the information in the database is GxP regulated. External controls may be required.

48.2.6 Other End User Applications

There are myriad tools that can be developed by end users that may impact decision-making, for example a multivariate statistical analysis using a tool such as SAS®. The level of GxP impact may vary from negligible in the case where the user is trying to optimize manufacturing yield by manipulating setpoints within already validated ranges, to critical when analyzing data related to clinical safety. Therefore, it is important to understand how the data from any end user application will be used.

48.3 Risk-Based Approach

End user applications can vary significantly in risk and complexity. The following are, however, required for all applications:

- Risk assessment and appropriate risk control measures to manage identified risks
- Appropriate specification and verification to demonstrate that the application performs as intended.

The strategy for specification and verification of the application being built should be based on:

- System impact on patient safety, product quality, and data integrity (risk assessment)
- System complexity and novelty (architecture and categorization of system components)
- Appropriate security to mitigate the risk of unauthorized changes to data or the application
- Management of the application under change control

Company policies and procedures should define their specific approach to achieving and maintaining compliance and fitness for intended use of end user applications.

This section of the appendix:

- Describes how the use of GAMP categories assists with understanding novelty and complexity
- Provides advice on appropriate risk-based controls
- Provides examples of typical approaches for different applications

48.3.1 Use of GAMP Categories

The tool on which the application is built, such as the spreadsheet package, should be considered as Category 1. Categories for spreadsheets and other end user applications should be viewed as a continuum that spans Categories 3, 4, and 5 (see Figure 48.1). Assignment of a category is a function of the complexity and novelty of the spreadsheet or application. Note, however, that a spreadsheet that merely makes use of the tabular editing power and does no calculations should be considered a document. The intended use of the analyzed data should be considered when determining the rigor of verification and the controls to be put in place. A simple spreadsheet (Category 3) may pose a high GxP risk depending upon the use of the data.

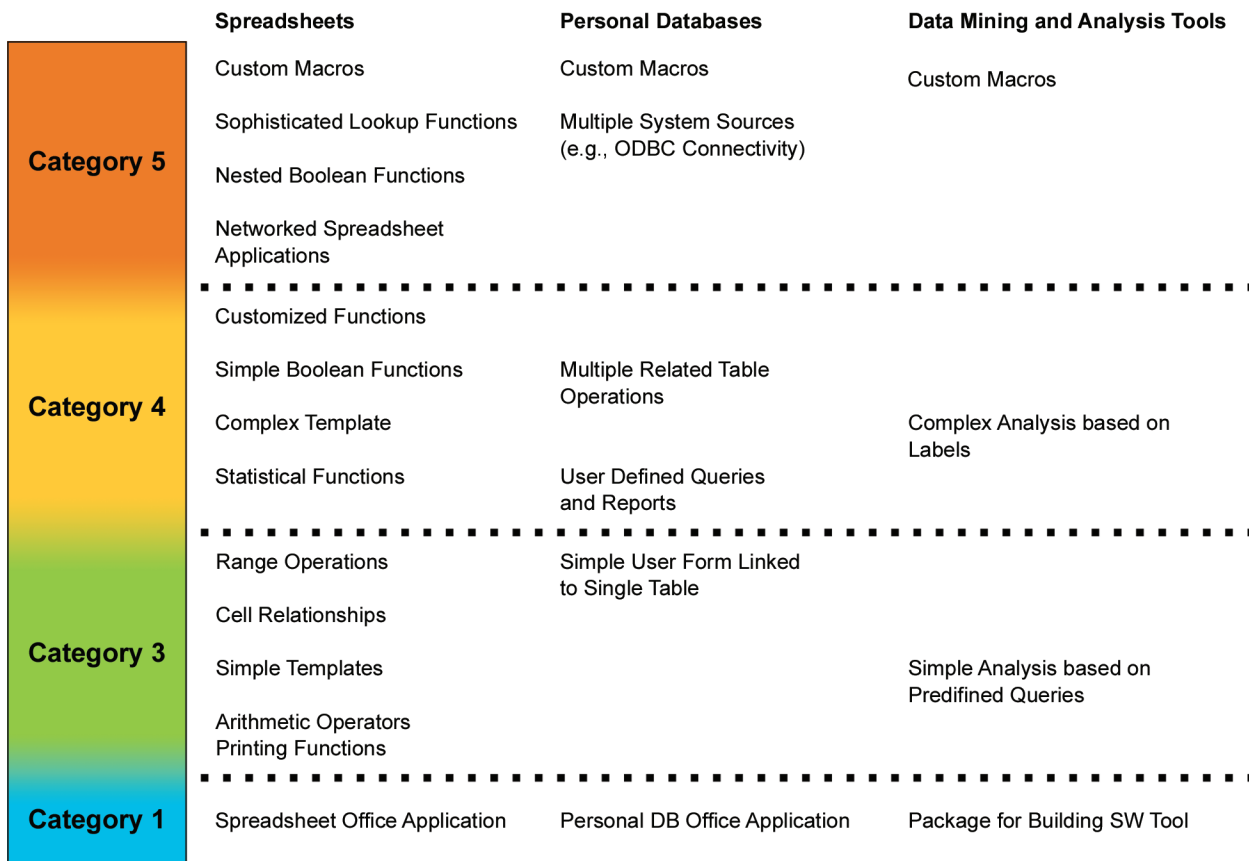
A spreadsheet that simply uses native functions to make calculations in place of a hand calculator is typically Category 3. For example, a laboratory analyst might create a unique spreadsheet to do a calculation related to an out of specification investigation. When the spreadsheet’s arithmetic functions are used, the calculations should be fully explained, as they would be in a text document. This should include verification that the intended formulas have been used properly, and that the data being analyzed is the right data. Such verification could easily be documented by having another analyst or a supervisor examine the spreadsheet and approve it. No further verification is required, since there is no need to challenge the accuracy of the calculations.

When developing spreadsheets as templates, such templates could be Category 3 to 5, depending on complexity, see Section 48.2.4. For the purpose of end user applications, the definition of Category 4 is altered: it focuses on complexity as opposed to configurability (see Figure 48.1).

For example:

- A template is used by analysts in a laboratory to do a routine calculation of averages and standard deviations of experimental results. This is a straightforward arithmetic operation with no configuration, so the template is Category 3.
- A spreadsheet template requires the user to input tablet strength, so that the application automatically branches to different cells to use strength-specific calculations based on this initial input. Such a simple operation would make the sheet Category 4, as it has some simple Boolean operations based on user input.
- A spreadsheet application that employs custom macros or sophisticated or nested logic or lookup functions should be treated as Category 5

Figure 48.1: Continuum of Categories for End User Applications



48.4 Risk-Based Controls

GxP risk²⁰ should be assessed. The following aspects should be considered:

- Data integrity related to the control of data files, as most end user applications process data
- The complexity of the application, based on the assumption that undetected systemic errors are more likely in software not developed under a rigorous development method, and more complex applications have more opportunities for errors
- Intended use of the data in support of GxP operations
- Potential impact on patient safety, product quality, or data integrity

Based on these risk assessments, controls should be established that focus on:

- Degree of verification (for example boundary condition testing might be more rigorous where risk is high. However, in a spreadsheet, ready access to the actual calculations to prove boundaries are correctly handled could alleviate the need for six-point testing.)
- Security control (for both the application code and any GxP records that are in the application or files created by or from it)
- Control of changes
- Version control of the spreadsheet in alignment with the history of changes
- Control of the infrastructure on which the end user application is built

48.4.1 Degree of Verification

The extent and rigor of verification should be based on risk, complexity, novelty, and intended use.

One level of testing may be appropriate for simple and low risk systems, or several levels may be required.

Complex and higher risk applications require more rigorous testing. The amount of logical branching in the application is a good gauge for complexity; if many logic functions (IF, AND, OR, etc.) or lookup tables are used, complexity is higher. Although they are native functions, these introduce more potential pathways through the application, and such branching requires a more sophisticated or thorough test strategy.

Macros also increase complexity because these are effectively embedded secondary applications. Even when created by keystroke capture, there is a program in a high-level language like VBA behind it. Macros that simply automate a string of actions are less of a concern than ones that contain logical branches, although they still introduce risk. Macros should be challenged in documented functional testing. Macros that include logical branches should be subject to greater rigor, with attention paid to multiple logic paths.

See Appendix D5 for further details on testing.

²⁰ Note that while this Guide focuses on GxP risks, the process is also readily applicable to business process risks.

48.4.2 Security Control

Security considerations for end user applications are similar to those for server or web-based applications, such as access to the application, access to data through the application, and access at the operating system level to data or the application code. Security within the environment should be adequate for the type of information stored or processed. As with server-based applications, security is a fundamental part of ensuring data integrity.

For many end user applications, a combination of infrastructure controls (e.g., restricted access to directories) and controls available through the application (e.g., password protection of spreadsheet cells) can provide some security against unintentional change. These controls may, however, be ineffective in keeping the application author from making changes outside of a change control process, especially if the application resides on an individual workstation. In some cases, it may be possible to improve security by running the end user application on a network drive on which the user's rights are limited, and which includes a regular scheduled back-up process.

Data is often saved within the application itself, especially in spreadsheets. Ensuring adequate data integrity held in spreadsheets requires the use of strict controls, including any required electronic record controls. Where spreadsheets are subject to edit, it is difficult to establish whether original data in subsequently saved copies has been edited. In such cases adequate control can be provided through the use of an EDMS. Alternatively, it may be necessary to maintain controlled copies in an unalterable format, e.g., PDF or hardcopy. In general, GxP data should be saved to a secure, backed-up local disk drive.

If the output of an end user application is a separately saved file, data integrity controls should be in place. Solutions could include:

- Forcing the file save path to go to a protected directory that disallows overwriting or deletion of files and prohibits unauthorized access
- Maintaining the files in an EDMS

If the degree of security that can be provided is not adequate for the data being managed, consideration should be given to the use of applications that operate in a more robust environment.

48.4.3 Change Control

End user applications that process GxP data should be subject to change control. Version management is difficult for such applications. In some cases, especially spreadsheets, management of the application within an EDMS may be an appropriate solution, as an audit trail of application versions will be retained. Another solution is to use library tools that are often used by developers to manage code. These can be used to manage any type of file, can be effective and reasonably easy to implement, and are less expensive than an EDMS. The use of either approach may also control risks related to security of the environment.

As with any change control process, changes to end user applications should have a change record that includes a description of the change and an assessment of the impact. Where appropriate, associated testing should be documented.

48.4.4 Control of the Infrastructure

End user application environments and products are Software Category 1 (see Appendix M4). These tools provide an application environment for the spreadsheets, databases, programs, or scripts that are developed by users.

The installation of the environment should be verified and the environment should to be managed under change and configuration management, considering:

- Standard records created by IT are adequate for such infrastructure, and are a standard part of the desktop build.

- In most companies, the IT department will also control the addition of non-standard desktop software. The processes they use to verify compatibility with the standard build are typically adequate to ensure proper control.
- If the desktop infrastructure software is added directly by the user without IT involvement, release notes should be perused for any known incompatibilities, and any irregularities in the installation process should be noted and investigated.

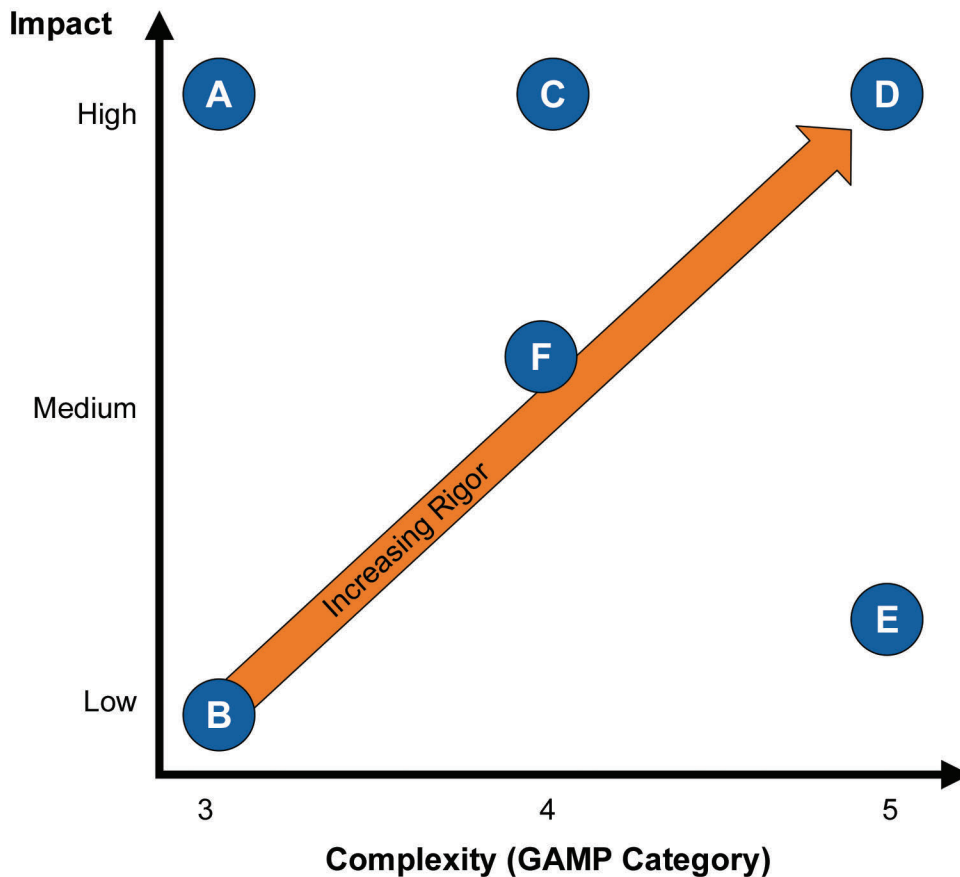
48.5 Examples of Typical Approaches

Figure 48.2 illustrates six different end user applications and a brief summary of potential approaches based on consideration of GxP impact and the complexity of the application. These examples are intended to be illustrative only, and not definitive. As in all risk-based approaches, applying critical thinking principles is imperative when considering the approach to end user application compliance, and should include the overall risk of the business process and the place of the end user application within it.

The analysis is based on an assumption of a constant level of risk. If the risk for a particular application is high, then the rigor should be increased.

Note that in all cases where “standard” documentation is recommended, for end user application these documents will typically be far smaller and easier to produce based on the fact that the applications themselves are smaller and have less functionality. For example, an RS will probably be a page or less in length. In all cases information may be combined into fewer deliverables than for larger, more complex applications.

Figure 48.2: Examples of Typical Approach Based on Impact and Complexity



A. Simple spreadsheet template for arithmetic calculation for content uniformity test:

- High impact, low complexity
- Recommended approach:
 - RS, documented verification by a third party that the calculations are the right ones
 - Security to ensure the sheet is protected against unauthorized change
 - Security to ensure the users can access only the approved version
 - Secure storage of electronic document
 - Change control

B. Spreadsheet record of training attendance

- Low impact, low complexity
- Recommended approach
 - No specific functionality requiring specification and verification
 - Standard controls for electronic documents containing evidence for GxP compliance

C. Desktop database for analyzing toxicology study

- High impact, medium complexity
- Recommended approach:
 - Full Category 4 approach:
 - > Validation plan
 - > RS
 - > Functional/design specification (may be combined)
 - > Traceability
 - > Documented testing against acceptance criteria
 - > Validation report
 - Security to limit access to authorized users
 - Change control

D. Spreadsheet for statistical analysis of a clinical study, with VBA macros

- High impact, high complexity

- Recommended approach:
 - Full Category 5 approach:
 - > Validation plan
 - > RS
 - > Functional/design specification (may be combined)
 - > Traceability
 - > Documented testing against acceptance criteria
 - > Validation report
 - Security to limit access to authorized users
 - Change control
- E. Spreadsheet for statistical analysis of manufacturing data for purpose of statistical process control of parameters within validated ranges (includes complex logic and lookup functions)
 - Low impact, high complexity
 - Recommended approach:
 - Documented verification by a third party that the calculations are the right ones
 - Change control
 - Security to ensure the sheet is protected against unauthorized change
 - Security to ensure the users can access only the approved version
- F. Desktop database tracking disposition of printed labels
 - Medium impact, medium complexity
 - Recommended approach:
 - Abbreviated Category 4 approach:
 - > Validation plan
 - > Combined RS/functional/design specification
 - > Documented testing against acceptance criteria
 - > Validation report
 - Change control
 - Security to limit access to authorized users

49 Appendix S4 – Patch and Update Management

49.1 Introduction

This appendix describes the compliance aspects to consider when planning security patches, hot fixes, or service pack upgrades.

There may be a frequent need for such patches and updates, for reasons including:

- Widely integrated and connected applications may be vulnerable on several levels to abuse and exploitation with malicious intent. Examples include:
 - Theft of personal data or IP
 - Theft of computing power and bandwidth for malicious software agents
- Complex software may be released with defects. Such defects may affect critical processes and require fixes to be issued by the supplier.
- Periodic software updates from suppliers

Appropriate management of patches and upgrades is particularly important to maintain compliance and fitness for intended use of GxP regulated computerized systems.

49.1.1 Changes from GAMP 5 First Edition

Clarification added regarding the role of the quality unit in patch management decisions.

49.2 Approach to Patch and Update Management

Regulated companies should develop an approach to patch and upgrade management that:

- Provides criteria for determining enterprise threat levels, and thus the urgency for applying patches
- Allows for flexibility in patch application that considers risks to both the enterprise and risks to the compliant status of regulated systems
- Generates configuration records that show the version and patch level for a system at any point in its life cycle
- Generates change records that describe what level of testing was done. This may include general testing at the enterprise level and/or application-specific tests. It may be determined by analysis of release notes that an application is unaffected, and no testing is required.

There is a need to determine what effect applying (or electing not to apply) a patch or upgrade will have on the compliance status of GxP regulated computerized systems. Many patches are released by suppliers to address urgent security vulnerabilities, and exploits may already be known or may be imminent. The time required to evaluate and test all affected GxP regulated computerized systems prior to implementation of the patch may therefore increase the risk to the integrity of these systems and their data.

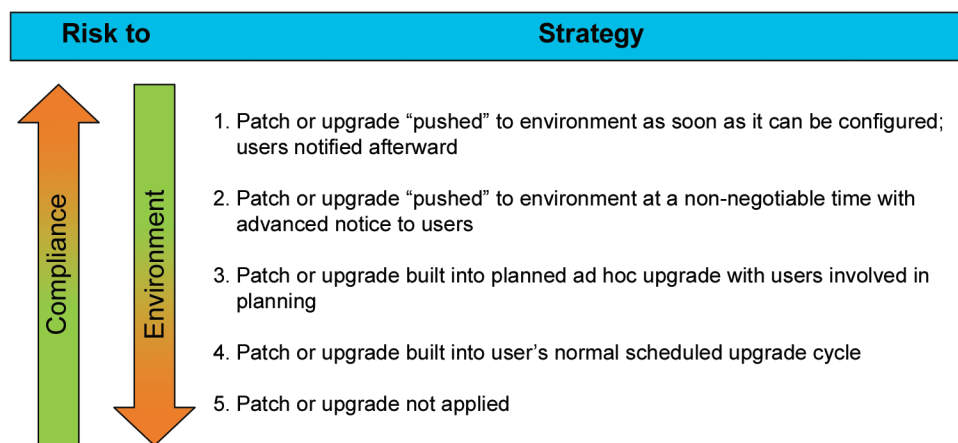
Regulated companies should develop a risk-management approach for patch and upgrade management that considers both regulatory compliance and the level of threat to the system and the wider computing environment. The approach should ensure there is a requirement for clear communication at the appropriate times.

Application-specific patches should be planned as part of normal change management procedures. Patches that must be applied enterprise wide are far more difficult to plan. Figure 49.1 illustrates some of the strategies that may be considered for such patches. The more aggressive options tend to minimize risk to the enterprise as a whole by expediting the fix, thus reducing exposure to the problem addressed by the patch. This has the side effect, however, of increasing the risk to the compliance status of individual applications because the impact of the change has not been considered fully in the context of each application.

The quality unit should approve the process for risk evaluation of all patches, but generally is not involved in the assessments. The evaluation should be performed by appropriate SMEs. If the quality unit is involved in an assessment, they should not have final say in how the patch is managed. Enterprise risk typically has to be given precedence over GxP risk. If a patch must be applied that is considered to have undesirable GxP risk, business process owners and the quality unit should work together to mitigate the risk. This could involve verification activities, a temporary business process change, or other approaches.

The process should define the criteria and required approvals for selecting and following each defined strategy, such as those shown in Figure 49.1. Roles and responsibilities should be defined. The process should also include notification to affected parties of the patch application, including the help desk in case unanticipated effects are raised as incidents.

Figure 49.1: Patch Strategies and Related Risk Levels



The selection of the strategy for applying the patch should consider the degree of risk reduction.

For example, for a fix to an operating system level security problem that threatens a wide range of GxP regulated and unregulated systems, it may be appropriate to follow Strategy 1 since this patch reduces risk to all of the applications.

Alternatively, the infrastructure group or other SMEs may be able to assess a risk level as being very low, e.g., when a patch disables a software port that is typically unused by applications. In this example the risk evaluation may conclude that there is no risk to the applications and therefore any of the patch strategies in Figure 49.1 may be selected. Furthermore, testing at the application level would not be necessary in this example based on knowledge of the application and the nature of the patch.

49.2.1 Configuration Management

Accurate and complete configuration management records support patch and update management in several ways, including:

- **Planning future patching activities:** Sometimes patches must be applied sequentially, so it is important to know the current patch level.
- **Interoperability:** If a system is running at multiple locations, there may be compatibility issues if different sites are at different patch levels. External applications may also require an interfaced system to be at a particular version and patch level.
- **Troubleshooting:** Thorough knowledge of a system's configuration is often critical to understanding what went wrong.
- **Data Integrity:** Application of a security patch may be crucial to data integrity, especially if exploits are widely published. It is important to be able to demonstrate that security gaps have been closed, and when this was achieved.

Configuration records for systems and infrastructure should be sufficient to show the current as-built state and when patches or upgrades have been applied.

See Appendix O6 for further details on these topics.

50 Appendix S5 (Retired)

Appendix S5 Managing Quality within an Outsourced IS/IT Environment has been withdrawn and the revised material renamed as Appendix M11.

51 Appendix S6 – Organizational Change

51.1 Introduction

This appendix provides guidance on how to deal with organizational change.

For many years there has been a global trend within the life sciences sector toward organizational consolidation and outsourcing of services, and this trend is likely to continue. However, additional challenges have been raised as a result of the COVID-19 pandemic and its resulting digital transformation that have stimulated and enabled more agile ways of working and thinking in the sector.

This appendix considers the impact of such changes on GxP regulated computerized systems and provides guidance in terms of areas to be considered for ensuring continued compliance and system availability.

51.1.1 Changes from GAMP 5 First Edition

While the approach to managing organizational change is essentially unchanged, minor revisions to some terminology have been made where appropriate.

51.2 Initiators for Change

Reasons for organizational change may include:

- Internal reorganization
- Being acquired by another company
- Divesting part of an organization to a third party (including activities such as offshoring and outsourcing)
- Supplier or service provider ceasing to trade

51.3 Scope and Impact of Change

Organizational change may apply to any aspect of the system supply chain, including regulated companies, system integrators, and base product and infrastructure suppliers.

Organizational change can occur during any stage of a computerized system's life cycle, and the impact of the change will depend on the stage. Aspects to consider include:

- Changes in the business process that a computerized system supports to enable optimization of collaborative efforts and strategies (for example, parallel phases of clinical development)
- Changes in how an existing system is used
- Moving systems from one location to another
- New or different regulatory and compliance requirements
- Impact on regulated records and any associated signatures

- Changes in how security (both physical and logical access control) is handled
- Clarification of where the primary data is located
- Clarification of who the process and system owners are or will be
- Timing and need to perform an audit of a reorganized supplier
- Business relationships with the supplier
- Impact of the change on any service level agreements
- Impact on company strategies with regard to preferred solutions
- Interim measures/solutions
- Maintaining expertise on systems (both with regard to the supplier and the regulated company)
- Validity of any support contracts with suppliers
- Postponement or cancellation of existing system implementation projects
- Acceleration or advancement of existing system implementation projects
- Changes in personnel and/or individual responsibilities.

51.4 Organizational Factors

Organizational challenges include:

- Maintaining multiple/parallel systems for the same business process
- Developing interfaces between these multiple/parallel systems
- Migrating data or subsets of data from one system to another
- Maintaining data sets for third parties for (possibly significant) periods of time
- Will all data be treated in the same manner?
- Will the data remain in the same format?
- Will some data be converted to paper/fiche records?
- Will some data be discarded?
- Systems likely to be retired
- the Location of life cycle records and information (paper and electronic) and inspection support on an ongoing basis
- Maintaining multiple/parallel compliance practices and documentation

- Harmonizing compliance practices, documenting the rationale for change (and justifying with regulator), and training in new practices
- Changes in regulatory expectations and subsequent impact life cycle activities and information
- How change management and configuration management will be handled across the changed organizations
- Ensuring that operation and maintenance activities are clearly identified and transitioned across to the revised organizations
- Increased focus/profile of incident monitoring during the transition period

51.5 Outsourcing

Where the organizational change is associated with outsourcing, then the following additional aspects should be considered:

- Whether the regulated company continues to own the equipment or whether this transfers to the outsource company
- Whether the outsource organization Quality Management System (QMS) is used or the regulated company QMS
- Need to both initially and periodically audit the outsource organization (the audit scope should be both compliance and financial)

See Appendix M11 for further details.

51.6 Loss of a Supplier

Where the change is associated with a supplier ceasing to trade consideration should be given to:

- Ensuring business continuity plans are established and accurate for the related systems
- Invoking any escrow agreements to gain access to application source code (if relevant)
- Record and system migration options
- Retrieval of any regulated company owned components, including hardware, software, records, and associated documentation retained from the supplier

51.7 Risk Assessment of Organizational Change

A risk assessment process to identify and rank risks should be executed in order to develop a plan. As part of the risk assessment process, projects should be considered as well as operational systems, the likely approach will depend on the nature of the change. Table 51.1 illustrates some possible scenarios:

Table 51.1: Possible Scenarios Resulting from Organizational Change

Project Status:	Company Acquisition	Company Merger	Supplier Insolvency
Not Started	Put on hold pending management review	Put on hold pending management review	Cancelled
In Progress Early in Life Cycle	Put on hold pending management review	Put on hold pending management review	Cancelled
In Progress Late in Life Cycle	Put on hold pending management review	Continued	Put on hold pending management review
Approaching Go-Live (Decisions on these projects will be a priority)	Put on hold pending management review	Continued	Put on hold pending management review

51.8 Affected Stakeholders

Whether dealing with internal or external organizations, agreement has to be reached between all affected organizations on any decisions concerning the system(s), the data, information, and documentation.

All affected stakeholders (from all organizations) should be involved, and where required should approve the strategy and decisions made.

Representatives from the following business areas typically would be involved:

- Business process owners
- Compliance/quality/regulatory
- Legal
- IT and Engineering
- Purchasing groups and in some cases finance

One of the key tasks for the business area stakeholders is to review and update as appropriate information and documentation affected by the organizational change. Some key areas to be considered are listed below:

- Business process documentation
- Policies, procedures, work instructions, test methods (if applicable)
- Training materials
- User manuals

- Batch records (if applicable)
- Archived records/data
- Contracts (if applicable)
- System interfaces
- Validation/qualification
- Record retention schedules

Another important task for stakeholders is to ensure that training requirements are evaluated. This should be performed against any changes made to the QMS, ways of working or documentation, and should be considered at both an organizational and a system level.

52 Appendix G1 – References

1. ISO 9000 Quality Management series, International Organization for Standardization (ISO), www.iso.org.
2. ISO 14971:2019 Medical devices -- Application of risk management to medical devices, International Organization for Standardization (ISO), www.iso.org.
3. ISACA® Capability Maturity Model Integration® (CMMI), <https://cmmiinstitute.com>.
4. ISO 12207:2017 Systems and software engineering — Software life cycle processes, International Organization for Standardization (ISO), www.iso.org.
5. ITIL® Foundation, ITIL 4 Edition, London, UK: Axelos, 2019, www.axelos.com.
6. International Council for Harmonisation (ICH), www.ich.org.
7. International Organization for Standardization (ISO), www.iso.org.
8. ASTM Standard E2500-20, “Standard Guide for Specification, Design, and Verification of Pharmaceutical and Biopharmaceutical Manufacturing Systems and Equipment,” ASTM International, West Conshohocken, PA, www.astm.org.
9. US FDA Center for Devices and Radiological Health (CDRH), Case for Quality, Food and Drug Administration (FDA), www.fda.gov.
10. *ISPE Good Practice Guide: Knowledge Management in the Pharmaceutical Industry*, International Society for Pharmaceutical Engineering (ISPE), First Edition, May 2021, www.ispe.org.
11. ISPE Initiative Advancing Pharmaceutical Quality (APQ), International Society for Pharmaceutical Engineering (ISPE), www.ispe.org/initiatives/quality-metrics.
12. ISPE iSpeak Blog, “ISPE Accelerating Digital Transformation with Pharma 4.0 Initiative,” International Society for Pharmaceutical Engineering (ISPE), 21 October 2021, www.ispe.org/pharmaceutical-engineering/ispeak/ispe-accelerating-digital-transformation-pharma-40-initiative.
13. ISPE Initiative Pharma 4.0™, International Society for Pharmaceutical Engineering (ISPE), accessed 3 June 2022, <https://ispe.org/initiatives/pharma-4.0>.
14. International Council for Harmonisation (ICH), ICH Harmonised Tripartite Guideline, Quality Risk Management – Q9, Step 4, 9 November 2005, www.ich.org.
15. ISPE GAMP® Good Practice Guide Series, International Society for Pharmaceutical Engineering (ISPE), www.ispe.org.
16. ISO 13485:2016 Medical devices — Quality management systems — Requirements for regulatory purposes, International Organization for Standardization (ISO), www.iso.org.
17. ISO 14971:2019 Medical devices — Application of risk management to medical devices, International Organization for Standardization (ISO), www.iso.org.
18. IEC 62304, *Medical Device Software – Software Life Cycle Processes*, Edition 1.1, 2015, International Electrotechnical Commission (IEC), www.iec.ch.

19. Sarbanes-Oxley Act of 2002, US Securities and Exchange Commission (SEC), www.sec.gov/about/laws/soa2002.pdf.
20. *ISPE GAMP® Good Practice Guide: Enabling Innovation – Critical Thinking, Agile, IT Service Management*, International Society for Pharmaceutical Engineering (ISPE), First Edition, September 2021, www.ispe.org.
21. International Council for Harmonisation (ICH), ICH Harmonised Tripartite Guideline, *Pharmaceutical Quality System – Q10*, Step 4, 4 June 2008, www.ich.org.
22. PIC/S Guidance: PI 011-3 Good Practices for Computerised Systems in Regulated “GXP” Environments, 25 September 2007, Pharmaceutical Inspection Co-operation Scheme (PIC/S), www.picscheme.org.
23. Federal Food, Drug, and Cosmetic Act (FD&C Act), US Food and Drug Administration (FDA), www.fda.gov.
24. Public Health Service Act, Title 42 of the United States Code (The Public Health and Welfare), Chapter 6A (Public Health Service), www.ecfr.gov.
25. United States Food & Drug Administration (FDA), www.fda.gov.
26. European Medicines Agency (EMA), www.ema.europa.eu/en.
27. United Kingdom Medicines & Healthcare products Regulatory Agency (MHRA), www.gov.uk/government/organisations/medicines-and-healthcare-products-regulatory-agency.
28. Japan Pharmaceuticals and Medical Devices Agency (PMDA), www.pmda.go.jp/english.
29. Prescription Drug Marketing Act of 1987; Prescription Drug Amendments of 1992; Policies, Requirements, and Administrative Procedures, A Rule by the Health and Human Services Department, and the Food and Drug Administration, Publication date: 3 December 1999, Effective date: 4 December 2000, www.ecfr.gov.
30. *ISPE Baseline® Pharmaceutical Engineering Guide, Volume 5 – Commissioning and Qualification*, International Society for Pharmaceutical Engineering (ISPE), Second Edition, June 2019, www.ispe.org.
31. PIC/S Guidance: PI 041-1 Good Practices for Data Management and Integrity in Regulated GMP/GDP Environments, 1 July 2021, Pharmaceutical Inspection Co-operation Scheme (PIC/S), www.picscheme.org.
32. EudraLex Volume 4 – Guidelines for Good Manufacturing Practices for Medicinal Products for Human and Veterinary Use, Annex 11: Computerized Systems, June 2011, http://ec.europa.eu/health/documents/eudralex/vol-4/index_en.htm.
33. General Principles of Software Validation; Final Guidance for Industry and FDA Staff, US Food and Drug Administration (FDA), 11 January 2002, www.fda.gov.
34. *ISPE GAMP® Good Practice Guide: A Risk-Based Approach to Operation of GxP Computerized Systems*, International Society for Pharmaceutical Engineering (ISPE), First Edition, January 2010, www.ispe.org.
35. *ISPE GAMP® Guide: Records and Data Integrity*, International Society for Pharmaceutical Engineering (ISPE), First Edition, March 2017, www.ispe.org.
36. *ISPE GAMP® RDI Good Practice Guide: Data Integrity by Design*, International Society for Pharmaceutical Engineering (ISPE), First Edition, October 2020, www.ispe.org.
37. MHRA Guidance: ‘GXP’ Data Integrity Guidance and Definitions, Revision 1, March 2018, Medicines & Healthcare products Regulatory Agency (MHRA), www.gov.uk/government/organisations/medicines-and-healthcare-products-regulatory-agency.

38. ISPE GAMP® Series, International Society for Pharmaceutical Engineering (ISPE), www.ispe.org.
39. ISO 9001:2015 Quality management systems — Requirements, International Organization for Standardization (ISO), www.iso.org.
40. American Institute of CPAs (AICPA), www.aicpa.org.
41. International Society for Pharmaceutical Engineering (ISPE), www.ispe.org.
42. Perez, A.D., Canterbury, J., Hansen, E., Samardelis, J.S., Longden, H., Rambo, R.L., "Application of the SOC 2+ Process to Assessment of GxP Suppliers of IT Services," *Pharmaceutical Engineering*, July/August 2019, Vol. 39, No. 4, pp. 14-20, www.ispe.org.
43. ISO 19011:2018 Guidelines for auditing management systems, International Organization for Standardization (ISO), www.iso.org.
44. ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements, International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.
45. ISO/IEC DIS 27002, Information security, cybersecurity and privacy protection — Information security controls, International Organization for Standardization (ISO), www.iso.org.
46. ISO/IEC 27017:2015, Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services, International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.
47. ISO/IEC 20000-1:2018 Information technology — Service management — Part 1: Service management system requirements, ISO/IEC JTC1, International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.
48. Simmon, E., NIST, Special Publication 500-322, "Evaluation of Cloud Computing Services Based on NIST SP 800-145," NIST Cloud Computing Cloud Services Working Group, NIST Cloud Computing Program, Information Technology Laboratory, National Institute of Standards and Technology (NIST), February 2018, <https://csrc.nist.gov>.
49. *ISPE GAMP® Good Practice Guide: IT Infrastructure Control and Compliance*, International Society for Pharmaceutical Engineering (ISPE), Second Edition, August 2017, www.ispe.org.
50. *ISPE GAMP® Good Practice Guide: A Risk-Based Approach to GxP Compliant Laboratory Computerized Systems*, International Society for Pharmaceutical Engineering (ISPE), Second Edition, October 2012, www.ispe.org.
51. ISPE Project Management Community of Practice, International Society for Pharmaceutical Engineering (ISPE), www.ispe.org.
52. Reid, C. and Wyn, S., "IT Services: Applying Good IT Practice & Automation," *Pharmaceutical Engineering*, May/June 2021, Vol. 41, No. 3, pp. 14-17, www.ispe.org.
53. Title 21-Food and Drugs, Chapter I, Food and Drug Administration Department of Health, Education, and Welfare, Subchapter C-Drugs: General [Docket No. 75n-0339] Human and Veterinary Drugs, Current Good Manufacturing Practice in Manufacture, Processing, Packing, or Holding, Preamble, VII. Definitions, paragraph 78, September 1978, www.fda.gov/files/drugs/published/Federal-Register-43-FR-45077.pdf.
54. "IT Infrastructure Library (ITIL)," ibm.com, 22 May 2019, www.ibm.com/cloud/learn/it-infrastructure-library.

55. AICPA, "2017 Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy," American Institute of CPAs (AICPA), March 2020, www.aicpa.org.
56. HITRUST®, <https://hitrustalliance.net/certifications/corporate-certifications>.
57. *ISPE GAMP® RDI Good Practice Guide: Data Integrity – Key Concepts*, International Society for Pharmaceutical Engineering (ISPE), First Edition, October 2018, www.ispe.org.
58. "Definition of WHISTLEBLOWER," www.merriam-webster.com, 28 June 2022, www.merriam-webster.com/dictionary.
59. EudraLex Volume 4 – Guidelines for Good Manufacturing Practices for Medicinal Products for Human and Veterinary Use, Chapter 7 – Outsourced Activities, January 2013, http://ec.europa.eu/health/documents/eudralex/vol-4/index_en.htm.
60. EMA, "Q&A: Good clinical practice (GCP)," European Medicines Agency (EMA), Accessed 28 June 2022, www.ema.europa.eu/en/human-regulatory/research-development/compliance/good-clinical-practice/qa-good-clinical-practice-gcp.
61. *ISPE GAMP® Good Practice Guide: A Risk-Based Approach to GxP Process Control Systems*, International Society for Pharmaceutical Engineering (ISPE), Second Edition, February 2011, www.ispe.org.
62. G.J. Myers, C. Sandler, T. Badgett, *The Art of Software Testing*, Third Edition, ISBN: 978-1-119-20248-6, Wiley, September 2015.
63. "Types of Static Analysis Methods," *GeeksforGeeks*, 22 April 2020, accessed 01 July 2022, www.geeksforgeeks.org/types-of-static-analysis-methods.
64. *ISPE GAMP® Good Practice Guide Series*, International Society for Pharmaceutical Engineering (ISPE), www.ispe.org.
65. *ISPE GAMP® Good Practice Guide: A Risk-Based Approach to Calibration Management*, International Society for Pharmaceutical Engineering (ISPE), Second Edition, November 2010, www.ispe.org.
66. ISO/IEC/IEEE 90003:2018 Software engineering — Guidelines for the application of ISO 9001:2015 to computer software, International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.
67. IEEE Standards Association, <https://standards.ieee.org>.
68. ISTQB® (International Software Testing Qualifications Board), www.istqb.org.
69. FDA's Technology Modernization Action Plan (TMAP), September 2019, Food and Drug Administration (FDA), www.fda.gov.
70. "Manifesto for Agile Software Development," 2001, <https://agilemanifesto.org>.
71. The Scrum Framework Poster, Scrum.org, accessed 20 August 2021, www.scrum.org.
72. A. McDonagh, S. Dubovik, M. R. Cherry, D. O'Brien, S. Jones, "Agile Software Development in GxP Regulated Environments GAMP® Special Interest Group," iSpeak Blog, 3 August 2020, International Society for Pharmaceutical Engineering (ISPE), www.ispe.org.
73. 21 CFR Part 11 – Electronic Records; Electronic Signatures, Code of Federal Regulations, US Food and Drug Administration (FDA), www.fda.gov.

74. Wake, B., "INVEST in Good Stories, and SMART Tasks," XP 123 Exploring Extreme Programming, 17 August 2003, <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks>.
75. Modernization in Action 2022, Technology Modernization Action Plan (TMAP) and Data Modernization Action Plan (DMAP) Anniversary Report, Food and Drug Administration (FDA), www.fda.gov.
76. Speer, J., "FDA Case for Quality: 2018 Comprehensive Review," *FDA Regulations and True Quality and Regulatory Affairs and Quality Management System (QMS) and Manufacturing and Computer System Validation*, 1 January 2019, www.greenlight.guru/blog/fda-case-for-quality-2018-comprehensive-review.
77. ISPE GAMP® Community of Practice, International Society for Pharmaceutical Engineering (ISPE), www.ispe.org.
78. Good Machine Learning Practice for Medical Device Development: Guiding Principles, October 2021, US Food and Drug Administration (FDA), www.fda.gov, Health Canada, www.canada.ca/en/health-canada.html, United Kingdom Medicines & Healthcare products Regulatory Agency (MHRA), www.gov.uk/government/organisations/medicines-and-healthcare-products-regulatory-agency.
79. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), GDPR Regulation (EU) 2016/679, (General Data Protection Regulation), <https://gdpr-info.eu>.
80. *ISPE GAMP® Good Practice Guide: Global Information Systems Control and Compliance*, International Society for Pharmaceutical Engineering (ISPE), Second Edition, February 2017, www.ispe.org.
81. National Institute of Standards and Technology (NIST), www.nist.gov.
82. ISO/IEC 27701:2019 Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines, International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.
83. ASTM International, West Conshohocken, PA, www.astm.org.
84. International Council for Harmonisation (ICH), ICH Harmonised Tripartite Guideline, *Pharmaceutical Development – Q8(R2)*, Step 5, August 2009, www.ich.org.
85. ANSI/ISA-88.00.01-2010 Batch Control Part 1: Models and Terminology, American National Standards Institute (ANSI), www.ansi.org.
86. International Society of Automation (ISA), www.isa.org.
87. 21 CFR Part 211.188 – Current Good Manufacturing Practice for Finished Pharmaceuticals, Batch production and control records, Code of Federal Regulations, US Food and Drug Administration (FDA), www.fda.gov.
88. EudraLex Volume 4 – Guidelines for Good Manufacturing Practice for Medicinal Products for Human and Veterinary Use, Chapter 4: Documentation, January 2011, http://ec.europa.eu/health/documents/eudralex/vol-4/index_en.htm.
89. 21 CFR Part 211.186 – Current Good Manufacturing Practice for Finished Pharmaceuticals, Master production and control records, Code of Federal Regulations, US Food and Drug Administration (FDA), www.fda.gov.
90. *ISPE GAMP® Good Practice Guide: Manufacturing Execution Systems – A Strategic and Program Management Approach*, International Society for Pharmaceutical Engineering (ISPE), First Edition, February 2010, www.ispe.org.

91. WHO Technical Report Series No. 996, WHO Expert Committee on Specifications for Pharmaceutical Preparations, Annex 5: Guidance on good data and record management practices, World Health Organisation (WHO), 2016, <https://apps.who.int/medicinedocs/documents/s22402en/s22402en.pdf>.
92. PIC/S Guide: PE 009-16 (Annexes), Guide to Good Manufacturing Practice for Medicinal Products Annexes, February 2022, Pharmaceutical Inspection Co-operation Scheme (PIC/S), www.picscheme.org.
93. *ISPE GAMP® RDI Good Practice Guide: Data Integrity – Manufacturing Records*, International Society for Pharmaceutical Engineering (ISPE), First Edition, May 2019, www.ispe.org.
94. FDA Guidance for Industry: Data Integrity and Compliance with CGMP, Questions and Answers, December 2018, US Food and Drug Administration (FDA), www.fda.gov.
95. EudraLex Volume 4 – Guidelines for Good Manufacturing Practices for Medicinal Products for Human and Veterinary Use, Annex 16: Certification by a Qualified Person and Batch Release, October 2015, http://ec.europa.eu/health/documents/eudralex/vol-4/index_en.htm.
96. *IEEE Standards Collection, Software Engineering*, Institute of Electrical and Electronics Engineers, 1994.
97. ISO/IEC TR 13066-3:2012 Information technology — Interoperability with assistive technology (AT) — Part 3: IAccessible2 accessibility application programming interface (API), International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.
98. "Definition of BLOCKCHAIN," www.merriam-webster.com, 15 July 2022, www.merriam-webster.com/dictionary.
99. ISO/IEC 27031:2011 Information technology — Security techniques — Guidelines for information and communication technology readiness for business continuity, International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.
100. International Council for Harmonisation (ICH), ICH Harmonised Tripartite Guideline, *Good Manufacturing Practice Guide for Active Pharmaceutical Ingredients – Q7/Q7A*, Step 4, 10 November 2000, www.ich.org.
101. Office of Regulatory Affairs, "Glossary of Computer System Software Development Terminology (8/95)," FDA, November 2014, www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/inspection-guides/glossary-computer-system-software-development-terminology-895.
102. ISO/IEC/IEEE 24765:2017 Systems and software engineering — Vocabulary, International Organization for Standardization (ISO), www.iso.org.
103. EMA, Guideline on good pharmacovigilance practices (GVP), Annex I - Definitions (Rev 4), European Medicines Agency (EMA), EMA/876333/2011 Rev 4, www.ema.europa.eu/en/human-regulatory/post-authorisation/pharmacovigilance/good-pharmacovigilance-practices#final-gvp-annex-i---definitions-section.
104. FDA Guide for Industry: Process Validation: General Principles and Practices, January 2011, Food and Drug Administration (FDA), www.fda.gov.
105. ISO 22966:2009 Execution of concrete structures, International Organization for Standardization (ISO), www.iso.org.
106. ISO/IEC TR 25051:2015 Software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing, International Organization for Standardization (ISO), www.iso.org, and International Electrotechnical Commission (IEC), www.iec.ch.

53 Appendix G2 – Glossary

53.1 Acronyms and Abbreviations

AI	Artificial Intelligence
ALCOA+	Attributable, Legible, Contemporaneous, Original, Accurate, Complete, Consistent, Enduring, Available
BaaS	Blockchain as a Service
BC	Business Continuity
BCP	Business Continuity Planning
BYOD	Bring Your Own Device
CAPA	Corrective and Preventive Action
CD	Continuous Deployment
CDRH	Center for Devices and Radiological Health
CDS	Chromatography Data System
CI	Continuous Integration
CMDB	Configuration Management Database
COTS	Commercial off the Shelf
CPP	Critical Process Parameter
CPU	Central Processing Unit
CQA	Critical Quality Attribute
CRO	Clinical Research Organization
CS	Configuration Specification
CSA	Computer Software Assurance
DAP	Data Archiving Plan
DIP	Dual In-line Package
DMS	Document Management System
DoD	Definition of Done
DoR	Definition of Ready
DQ	Design Qualification
DR	Disaster Recovery
DS	Design Specification
EDA	Electronic Data Archive

EDMS	Electronic Document Management System
EMA	European Medicines Agency (EU)
EPR	Electronic Production Record
ERP	Enterprise Resource Planning
ETL	Tool that extracts, transforms, and loads data
EU	European Union
FAT	Factory Acceptance Test
FDA	Food and Drug Administration (US)
FMEA	Failure Mode Effects Analysis
FS	Functional Specification
GDPR	General Data Protection Regulation (EU)
GEP	Good Engineering Practice
GLP	Good Laboratory Practice
GMP	Good Manufacturing Practice
GxP	Good “x” Practice
HSE	Health, Safety, and Environment
HVAC	Heating, Ventilation, and Air Conditioning
IaaS	Infrastructure as a Service
IaC	Infrastructure as Code
ICH	International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use
INVEST	Independent, Negotiable, Valuable, Estimable, Small, Testable
IP	Intellectual Property
ISA	International Society of Automation
ISMS	Information Security Management System
IT	Information Technology
ITQ	IT Quality
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
LIMS	Laboratory Information Management System
LMS	Learning Management System
MES	Manufacturing Execution System

MHRA	Medicines & Healthcare products Regulatory Agency
ML	Machine Learning
MSA	Master Service Agreement
MVP	Minimum Viable Product
NIST	National Institute of Standards and Technology
OLA	Operating Level Agreement
OQ	Operational Qualification
OSS	Open-Source Software
P&ID	Process and Instrumentation Diagram
PaaS	Platform as a Service
PAT	Process Analytical Technology
PI	Personal Information
PII	Personally Identifiable Information
PLC	Programmable Logic Controller
PPQ	Process Performance Qualification
PQ	Performance Qualification
QA	Quality Assurance
QbD	Quality by Design
QC	Quality Control
QMS	Quality Management System
QRM	Quality Risk Management
QTPP	Quality Target Product Profile
R&D	Research and Development
RAID	Risks, Actions, Issues, Decisions
RFP	Request for Proposal
RPO	Recovery Point Objective
RS	Requirements Specification
RTM	Requirements Trace Matrix
RTO	Recovery Time Objectives
SaaS	Software as a Service
SaMD	Software as a Medical Device

SBOM	Software Bill of Materials
SLA	Service Level Agreement
SME	Subject Matter Expert
SOP	Standard Operating Procedure
SOW	Statement Of Work
SOX	Sarbanes-Oxley
TMF	Trial Master File
UAT	User Acceptance Testing
UC	Underpinning Contract
UPS	Uninterruptable Power Supply
URS	User Requirement Specification
US	United States
UV	Ultraviolet
VBA	Visual Basic® for Applications
VM	Virtual Machine
VMP	Validation Master Plan
WAN	Wide Area Network
XaaS	Infrastructure/Platform/Software as a Service

53.2 Definitions

Acceptance Criteria (ASTM [8])

The criteria that a system, component²¹ must satisfy in order to be accepted by a user, customer or other authorized entity.

Acceptance Test (IEEE [96])

Testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.

Application Software (ISO/IEC 13066 [97])

Software or a program that is specific to the solution of an application problem.

Audit (ISO 9000 [1])

Systematic, independent, and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which agreed criteria are fulfilled.

²¹ In this Guide acceptance criteria may also be defined for functions, i.e., at a lower level than component or system.

Batch Record

The set of records of all relevant process information in any physical or electronic format.

Blockchain (Merriam-Webster [98])

A digital database containing information (such as records of financial transactions) that can be simultaneously used and shared within a large decentralized, publicly accessible network.

Business Continuity Planning (ISO/IEC 27031:2011 [99])

Documented procedures that guide organizations to respond, recover, resume, and restore to a pre-defined level of operation following disruption.

Calibration (ICH Q7 [100])

The demonstration that a particular instrument or device produces results within specified limits by comparison with those produced by a reference or traceable standard over an appropriate range of measurements.

Change Management (ICH Q10 [21])

A systematic approach to proposing, evaluating, approving, implementing and reviewing changes.

Coding Standards (FDA [101])

Written procedures describing coding [programming] style conventions specifying rules governing the use of individual constructs provided by the programming language, and naming, formatting, and documentation requirements which prevent programming errors, control complexity and promote understandability of the source code. Syn: development standards, programming standards.

Commercial Off-the-Shelf Software (IEEE [96])

Software defined by a market-driven need, commercially available, and whose fitness for use has been demonstrated by a broad spectrum of commercial users. Also known as COTS.

Computer System (IEEE [96])

A system containing one or more computers and associated software.

Computerized System

A broad range of systems including, but not limited to, automated manufacturing equipment, automated laboratory equipment, process control and process analytical, manufacturing execution, laboratory information management, manufacturing resource planning, clinical trials data management, vigilance and document management systems. The computerized system consists of the hardware, software, and network components, together with the controlled functions and associated documentation.

Computerized System Validation

Achieving and maintaining compliance with applicable GxP regulations and fitness for intended use by:

- The adoption of principles, approaches, and life cycle activities within the framework of validation plans and reports
- The application of appropriate operational controls throughout the life of the system

Critical Process Parameter (ICH Q8 [84])

A process parameter whose variability has an impact on a critical quality attribute and therefore should be monitored or controlled to ensure the process produces the desired quality.

Critical Quality Attribute (ICH Q8 [84])

A physical, chemical, biological or microbiological property or characteristic that should be within an appropriate limit, range, or distribution to ensure the desired product quality.

Design (IEEE [96])

The process of defining the architecture, components, interfaces, and other characteristics of a system or component.

Design Review ((ISO/IEC/IEEE 24765 [102]))

(1) Formal, documented, comprehensive, and systematic examination of a design to determine if the design meets the applicable requirements, to identify problems, and to propose solutions. (2) A process or meeting during which a system, hardware, or software design is presented to project personnel, managers, users, customers, or other interested parties for comment or approval.

Detectability (ICH Q9 [14])

The ability to discover or determine the existence, presence, or fact of a hazard.

Dynamic Machine Learning Sub-System

Online learning may be deployed to continually update the model parameters during operation as additional data is acquired

Electronic Batch Record

A type of an EPR that is a store of data and information for a batch or continuous processes.

Electronic Production Record

A record that is a store of data and information from production-related activities created by, and/or manually entered into systems, typically during execution of control recipes. The EPR may be located in one or more systems or databases.

Factory Acceptance Test (FAT) (IEEE [96])

An Acceptance Test in the Supplier's factory, usually involving the Customer. See also Acceptance Test. Contrast to Site Acceptance Test.

GxP Compliance

Meeting all applicable pharmaceutical and associated life-science regulatory requirements.

GxP Regulated Computerized System

Computerized systems that are subject to GxP regulations. The regulated company must ensure that such systems comply with the appropriate regulations.

GxP Regulation

The underlying international pharmaceutical requirements, such as those set forth in the US FD&C Act, US PHS Act, FDA regulations, EU Directives, Japanese regulations, or other applicable national legislation or regulations under which a company operates. These include but are not limited to:

- Good Manufacturing Practice (GMP) (pharmaceutical, including Active Pharmaceutical Ingredient (API), veterinary, and blood)
- Good Clinical Practice (GCP)
- Good Laboratory Practice (GLP)
- Good Distribution Practice (GDP)
- Good Quality Practice (GQP)
- Good Pharmacovigilance Practice (GVP)
- Medical Device Regulations
- Prescription Drug Marketing Act (PDMA)

Harm (ICH Q9 [14])

Damage to health, including the damage that can occur from loss of product quality or availability.

Hazard (ICH Q9 [14])

The potential source of harm (ISO/IEC Guide 51).

Incident

Operational event which is not part of standard operation.

Middleware

Software that provides common services such as communication and data management to software applications beyond those available from the operating system,

Network (FDA [101])

A system [transmission channels and supporting hardware and software] that connects several remotely located computers via telecommunications.

Periodic Review

A documented assessment of the documentation, procedures, records, and performance of a computer system to determine whether it is still in a validated state and what actions, if any, are necessary to restore its validated state. The frequency of review is dependent upon the systems complexity, criticality, and rate of change.

Pharmacovigilance (EMA [103])

Science and activities relating to the detection, assessment, understanding and prevention of adverse effects or any other medicine-related problem.

Process (ISO 9000 [1])

Set of interrelated or interacting activities that use inputs to deliver an intended result.

Process Analytical Technology (ICH Q8 [84])

A system for designing, analyzing, and controlling manufacturing through timely measurements (i.e., during processing) of critical quality and performance attributes of raw and in-process materials and processes with the goal of ensuring final product quality.

Process Validation (FDA [104])

The collection and evaluation of data, from the process design stage through commercial production, which establishes scientific evidence that a process is capable of consistently delivering quality products.

Process Owner

This is the owner of the business process or processes being managed. The process owner is ultimately responsible for ensuring that the computerized system and its operation is in compliance and fit for intended use in accordance with applicable company Standard Operating Procedures (SOPs). The process owner may also be the system owner. The process owner may be the *de facto* owner of the data residing on the system (data owner) and therefore, ultimately responsible for the integrity of the data. Process owners are typically the head of the functional unit using the system.

(cf. System Owner)

Product Lifecycle (ICH Q9 [14])

All phases in the life of the product from the initial development through marketing until the product's discontinuation.

Production Report

Information in human readable form, presented via electronic, paper or hybrid format for activities such as review, disposition, investigation, audit and analysis.

Quality (ICH Q9 [14])

The degree to which a set of inherent properties of a product, system or process fulfills requirements (see ICH Q6A definition specifically for "quality" of drug substance and drug (medicinal) products.)

Quality [product] (ICH Q8 [84])

The suitability of either a drug substance or drug product for its intended use. This term includes such attributes as the identity, strength, and purity (from *ICH Q6A Specifications: Test Procedures and Acceptance Criteria for New Drug Substances and New Drug Products: Chemical Substances*).

Quality by Design (ICH Q8 [84])

A systematic approach to development that begins with predefined objectives and emphasizes product and process understanding and process control, based on sound science and quality risk management.

Quality Management System (ISO 9000 [1])

Part of a management system with regard to quality. (A **management system** is a set of interrelated or interacting elements of an organization to establish policies and objectives, and processes to achieve those objectives.)

(This is equivalent to Quality System as defined in ICH Q9 [14])

Quality Plan (ISO 22966 [105])

Document specifying which procedures and associated resources shall be applied by whom and when to meet the requirements of the specific project

Quality Risk Management (ICH Q9 [14])

A systematic process for the assessment, control, communication and review of risks to the quality of the drug (medicinal) product across the product lifecycle.

Quality System (ICH Q9 [14])

The sum of all aspects of a system that implements quality policy and ensures that quality objectives are met.

Recursive Hierarchy

Grouping together similar requirements relating to a specific area of functionality. This hierarchical approach, which may have multiple levels depending on the complexity of the process or system, may help simplify requirements management and risk assessment.

Requirement (ISO 9000 [1])

Need or expectation that is stated, generally implied or obligatory.

Risk (ICH Q9 [14])

The combination of the probability of occurrence of harm and the severity of that harm (ISO/IEC Guide 51).

Risk Analysis (ICH Q9 [14])

The estimation of the risk associated with the identified hazards.

Risk Assessment (ICH Q9 [14])

A systematic process of organizing information to support a risk decision to be made within a risk management process. It consists of the identification of hazards and the analysis and evaluation of risks associated with exposure to those hazards.

Risk Communication (ICH Q9 [14])

The sharing of information about risk and risk management between the decision maker and other stakeholders.

Risk Control (ICH Q9 [14])

Actions implementing risk management decisions (ISO Guide 73).

Risk Evaluation (ICH Q9 [14])

The comparison of the estimated risk to given risk criteria using a quantitative or qualitative scale to determine the significance of the risk.

Risk Identification (ICH Q9 [14])

The systematic use of information to identify potential sources of harm (hazards) referring to the risk question or problem description.

Risk Management (ICH Q9 [14])

The systematic application of quality management policies, procedures, and practices to the tasks of assessing, controlling, communicating and reviewing risk.

Risk Reduction (ICH Q9 [14])

Actions taken to lessen the probability of occurrence of harm and the severity of that harm.

Risk Review (ICH Q9 [14])

Review or monitoring of output/results of the risk management process considering (if appropriate) new knowledge and experience about the risk.

Severity (ICH Q9 [14])

A measure of the possible consequences of a hazard.

Site Acceptance Test (SAT) (IEEE [96])

An Acceptance Test at the Customer's site, usually involving the Customer. See also Acceptance Test, contrast to Factory Acceptance Test.

Source Code (FDA [101])

(1) (IEEE) Computer instructions and data definitions expressed in a form suitable for input to an assembler, compiler or other translator. (2) The human readable version of the list of instructions [program] that cause a computer to perform a task.

Software Life Cycle (NIST [81])

Period of time beginning when a software product is conceived and ending when the product is no longer available for use. The software life cycle is typically broken into phases denoting activities such as requirements, design, programming, testing, installation, and operation and maintenance.

Specification (IEEE [96])

A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or component, and often, the procedures for determining whether these provisions have been satisfied.

Static Machine Learning Sub-System

Offline where there are controlled and identifiable changes to the case data and algorithm.

Subject Matter Expert (ASTM E2500 [8])

Those individuals with specific expertise in a particular area or field. Subject Matter Experts should take the lead role in the verification of computerized systems. Subject Matter Expert responsibilities include planning and defining verification strategies, defining acceptance criteria, selection of appropriate test methods, execution of verification tests, and reviewing results.

Supplier

An organization or individual internal or external to the user associated with the supply and/or support of products or services at any phase throughout a systems life cycle.

System Owner

The system owner is responsible for the availability, and support and maintenance, of a system, and for the security of the data residing on that system. The system owner is responsible for ensuring that the computerized system is supported and maintained in accordance with applicable company SOPs. The system owner also may be the process owner (e.g., for IT infrastructure systems or systems not directly supporting GxP). For systems supporting regulated processes and maintaining regulated data and records, the ownership of the data resides with the GxP-process owner, not the system owner.

The system owner acts on behalf of the users. The system owner for larger systems will typically be from IT or engineering functions. Global IT systems may have a global system owner and a local system owner to manage local implementation.

(cf. Process Owner)

Testing, Functional (IEEE [96])

(1) Testing that ignores the internal mechanism or structure of a system or component and focuses on the outputs generated in response to selected inputs and execution conditions. (2) Testing conducted to evaluate the compliance of a system or component with specified functional requirements and corresponding predicted results. Syn: black-box testing, input/output driven testing. Contrast with testing, structural.

Testing, Structural (IEEE [96])

(1) Testing that takes into account the internal mechanism [structure] of a system or component. Types include branch testing, path testing, statement testing. (2) Testing to ensure each program statement is made to execute during testing and that each program statement performs its intended function. Contrast with functional testing. Syn: white-box testing, glass-box testing, logic driven testing.

Test Case (ISO/IEC 25051 [106])

A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test Plan (ISO/IEC 25051 [106])

A document describing the scope, approach, resources, and schedule of intended test activities.

Test Procedure (ISO/IEC 25051 [106])

Detailed instructions for the set-up, execution, and evaluation of results for a given test case.

Verification (1: ISO 9000 [1], 2: ASTM E2500 [8])

(1) Confirmation, through the provision of objective evidence that specified requirements have been fulfilled. (2) A systematic approach to verify that manufacturing systems, acting singly or in combination, are fit for intended use, have been properly installed, and are operating correctly. This is an umbrella term that encompasses all types of approaches to assuring systems are fit for use such as qualification, commissioning and qualification, verification, system validation, or other.

User

The pharmaceutical customer or user organization contracting a supplier to provide a product. In the context of this document it is, therefore, not intended to apply only to individuals who use the system, and is synonymous with customer.



600 N. Westshore Blvd., Suite 900, Tampa, Florida 33609 USA
Tel: +1-813-960-2105, Fax: +1-813-264-2816

www.ISPE.org