STANDARD OPERATING PROCEDURE

DOC TITLE:	**SOFTWARE DEVELOPMENT LIFE CYCLE**
DOC ID:	**SOP-1705**
VERSION:	**24.0**
PROCESS CATEGORY:	**SDLC PROCESS**

TABLE OF CONTENTS

1 Objective

The purpose of this standard operating procedure (SOP) is to outline the Software Development Life Cycle (SDLC) to be followed from product planning through archival and retirement.

This procedure defines the minimum activities to perform development, hosting, maintenance and other activities that support J&J business applications and commercial software.

This procedure also defines the security and compliance review and mitigation measures applicable to software used in J&J.

2 Scope

All technology assets owned, developed, and managed by Johnson & Johnson Technology (JJT) shall adhere to the applicable Johnson & Johnson Technology – Technology Services (JJT-TS) approved SDLC procedure.

This procedure may be adopted by any J&J operating company or enterprise function as the guiding methodology for software development activities. Where a J&J operating company or other J&J legal entity accountable for software development requires an alternate software development lifecycle methodology, this procedure still dictates the applicable support that JJT-TS will provide. In those instances, this procedure will serve as the basis for defining a common set of practices between JJT-TS and the accountable entity.

3 Definitions

It is assumed the audience of this SOP understands basic IT terms and acronyms, which are not included in this section.

3.1 Terms and Acronyms

- **Functional Requirements –** Requirements that describe the behavior of the solution and the information managed, allowing for developers to fully understand how the software must function and what attributes are needed to meet user requirements. In the case of a system or application, these are the features and functions of the system.
- **Non-functional Requirements –** Requirements that define the qualities of the solution or the environmental conditions under which the solution will remain effective (e.g., response time, security, availability).
- **Product Backlog –** A prioritized list of items, managed by the Business Owner, which may be included in a product. Examples include features, user stories, bugs, technical work, knowledge acquisition (e.g. spikes), documentation or other artifacts plus regulatory, internal controls, and security requirements.
- **Software Asset –** All software used to support business processes or digital products that are internal- or external-facing.

3.2 Role Descriptions

The following roles and organizations are required for compliance and risk determination and may subsequently be required based on the outcome of the assessment at various stages of the software development life cycle.

- **Business Owner** – The main sponsor or owner of the system who drives the product vision, roadmap, and owns the product backlog. This individual is accountable for product planning and acceptance of the system prior to its release for use.
- **Technical Owner** – The individual responsible for the end-to-end delivery of the service (design, engineering, deployment, and operations), who ensures quality and timely delivery of service to consumers, and who determines functionality roadmap and milestones for the provided service.
- **Product Team –** The group of individuals assigned to a software development and/or implementation effort. Personnel may include Business Owner, Business Analysts, Solution Architects, Project Managers, Testers, Developers, Technical Owner, Quality Representatives, and others as needed.
- **Quality** – A representative from a Quality organization (for example, Technology Quality (TQ), Solutions Quality Assurance (SQA), Business Quality) that ensures product team delivers software following applicable guidance on best quality and validation practices. This may involve providing coaching and guidance on best quality and validation practices, while making sure that the delivered software, the related deliverables, and applied processes comply with defined J&J quality standards and policies. The Compliance Analysis must be reviewed and approved by a Technology Quality (TQ) representative.
- **Privacy Officer**– A representative from Privacy organization that may be involved to provide guidance and review strategies and approaches to address data subject rights and the secure management of personal data.
- **Records and Information Management (RIM)** – The organization that provides guidance on the systematic creation, distribution, use, maintenance and disposal of J&J records and information in compliance with the Worldwide RIM Policy and Standards.
- **Regulatory Affairs (RA) Specialist** – A representative of a given sector or operating company that determines the health authority classification of software for a regulated product/medical device and the strategy to engage with health authorities on the development based on software classification.
- **Security Officer** – A representative from Information Security and Risk Management (ISRM) responsible for performing the security assessment and providing the expertise and tools to design, build and operate secure application software consistent with the Information Asset Protection Policies (IAPP). The ISRM Officer also assesses the privacy sensitivity of data used in applications.

4 References

- **Doc ID:** – N/A **Title:** - S-10 Worldwide Policy for System Development Lifecycle (for IAPP)
- **Doc ID:** – POL-1769 **Title:** - Johnson & Johnson JJT-TS Quality Manual
- **Doc ID:** – TMP-1138 **Title:** - Compliance Analysis

5 SDLC Framework and Activities

The SDLC Framework (see Figure 1 SDLC Framework) outlines activities to be performed both iteratively and incrementally starting from the lowest unit of work which is a requirement or a user story. Requirements are derived from the intended use and compliance and risk determination.

The ***requirements*** are divided into small pieces of desired functionality and are sized to be completed in a single development iteration. The ***development iteration*** involves creating software design for the prioritized set of requirements, configuration and/or development, testing and verification of a software unit as well as integration testing. An ***increment*** is the integration of outcomes from a series of development iterations. With each development iteration, the increment increases in terms of usable functionality delivered and contributes to a ***release***. One or more releases may be planned as part of a project. Requirements for each release may be managed through a product backlog. For simple software (e.g., some off-the-shelf software or websites), requirements may be defined through intended use and minimal configuration requirements necessary for the software to perform as intended.

The framework enforces a scalable, closed loop feedback mechanism with traceability from requirements to release and feedback through the monitoring process during the operational life of the software product. The SDLC activities performed as part of this framework may be carried out in sequence or in parallel.

Operational controls are created and maintained in parallel with the software development iteration prior to releasing the system for use. These controls govern how the software will be deployed, monitored and maintained. These controls also cover administrative practices for managing user accounts and system data.

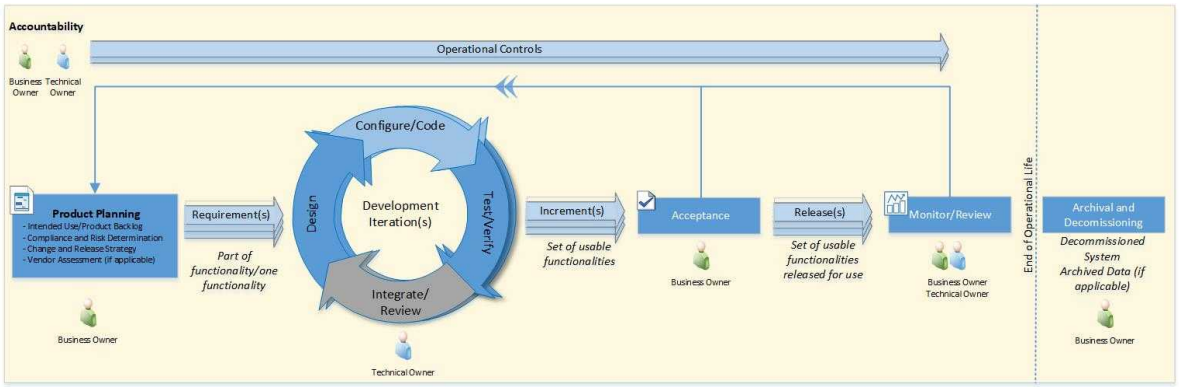

Figure 1 SDLC Framework

Note:

- **Scope of the release(s) – please note there may be multiple releases planned simultaneously for large and complex software engineering projects**
- **All interfaces and integrations should be documented and taken into consideration while defining the scope of the release.**

For every change that results in a release of the software product/system, the Product Team should follow the activities outlined in the below sections.

5.1 Product Planning

The below activities should be carried out and documented at the software product level to ensure requirements are taken into consideration during design and development, including applicable operational, regulatory, security and any other associated controls.

- Defining the Intended Use (Creation or Refinement of Product Backlog)
  - The intended use of the application or product should be clearly defined to serve as the mechanisms for selecting or developing a solution and the basis for evaluating the effectiveness of the software release.
  - A Product Backlog may be established covering requirements including but not limited to new features, enhancements, changes to existing features, bug fixes and infrastructure changes.

- Compliance and Initial Risk Determination
  - Clear description of the system covering the purpose and intended use of the system, the users of the systems, modalities to be used, types of data collection, design, hosting and support details
  - Determination of the following aspects given the intended use:
    - Classification of data sensitivity
    - Applicable regulations and policies that include but are not limited to GxP, SOX, Privacy, RIM and any other external regulations or laws
    - Corresponding risks and implications for each category of regulation or policy that applies
    - Retention Requirements
    - Business risk(s) in terms of business processes and organizations impacted directly or indirectly in case of a prolonged service disruption or outage
    - Security risk profile of the application that may require other additional activities and/or assessments to be performed to protect dimensions of security
  - All software assets will be managed by a JJT-approved asset tracking tool.
  - Any software developed under this procedure will use the J&J approved Compliance Analysis tool.

- Vendor Assessment
  - For third-party vendors, a vendor risk assessment focused on Information Security and Privacy must be carried out by an ISRM representative to

ensure they have sufficient controls to protect data and maintain daily operations.
    - As applicable, Vendors should also be assessed for their Quality Management System (QMS) to ensure alignment to J&J's requirements for building good quality software and regulatory compliance. The Business Owner or the Technical Owner is accountable for initiating a vendor risk or QMS assessment.

- Change and Release Strategy
    - All products or software should have a defined Change and Release strategy covering the following:
    - Description of process activities that will be completed to enable changes to an application/system as part of a release, with minimum disruption to end-user services, into an environment. (i.e., for code promotion to quality and production environments)
    - All changes should adhere to the strategy laid out for authorization, scheduling, and approval prior to implementation.

5.2 Requirements Definition

Requirements must be captured to define the features and operational expectations of software. Requirements must be clearly defined such that stakeholders have a common understanding and are aligned on how the software should function. A Product Backlog may be used to prioritize requirements or user stories for each ***development iteration***.

Depending on the complexity of the software and the framework (e.g. Agile, Waterfall) used, requirements may be memorialized as terms in agreements with third parties (e.g., for off-the-shelf software), a requirements document, or a set of user stories documented in a tool that has been assessed and deemed suitable for us by J&J. Requirements may be specific to features and functions within the software or may include operational and administrative controls governing maintenance and system administration.

The compliance and risk determination will assess applicability of specific legal, security and compliance requirements. Where applicable, the requirements described below must be included in release requirements.

- GxP regulatory and quality standard requirements for product quality, patient safety, and data integrity, globally. This may include but is not limited to requirements related to managing quality records, functions, or decisions
- Security controls intended to prevent malicious or inadvertent exposure or loss of data, disruption of operations, impact to data integrity, and unauthorized access to the software.
- Privacy requirements should also address any laws requiring data and operations hosting, including where data must reside and where it may be legally transferred. Financial controls to assure the accuracy of financial transactions and disclosures, and to enable audits and reports on those controls.
- Design, features and functions to allow our software to be accessible to people with a wide range of abilities.

Commercial or public-facing software should also address potential requirements for adverse event and product quality complaint reporting, content management and possible regulatory classifications within deployed regions. All public-facing content is subject to formal copy review processes governed by the publishing entity.

5.3 Design and Configure/Code

The design should enable development team(s) to implement both functional and non-functional requirements. This may include configuration and/or customization involving development or modification of existing code. Code should be reviewed and aligned to the coding standards of a specific programming language/technology and should undergo unit testing.

5.4 Testing and Acceptance

The purpose of this activity is to right-size the extent and types of testing required to declare the application/release fit-for-purpose. While this procedure mandates that testing be performed, this can be tailored based on the scope of the release and the application in consultation with the quality representative. This procedure recommends that testing be carried out to ensure the system components work together as intended and is acceptable for business use.

This methodology encourages the use of automated testing, where feasible, to allow greater consistency and robustness for verification of functionality and performance. Where verification supports the overall acceptance of the release, business owners should tailor this type of testing to challenge the use of the software in an environment reflective of its production environment and with users who represent the user base.

Documentation of the test strategy must cover types of testing that have been agreed on by the product team and quality representative.

- Test Planning should involve creation of test scripts with traceability to requirements ensuring appropriate coverage
- Performing and documenting test executions provide verification that the requirements were implemented correctly. Instances of unintended outcomes that occur during testing should be tracked as defects.
- All defects must be tracked to closure, even if the defect is not to be resolved until a future increment/release. Defects that will remain open after the release goes live must be recorded at the end of the test cycle and should be tracked to closure in subsequent releases.
- Requirements requiring testing must be traceable to the test scripts and test results to demonstrate that the software is functioning as designed. Operational requirements will be verified through functionality and procedures, as applicable.

5.5 Deployment and Operations Planning

Deployment planning should take the following into consideration:

- Criteria for production readiness checks
- Planning for transition to support personnel that includes but is not limited to:

- Knowledge transition from the development to operations team, shadow support and transition to full support
- Determination of criteria to ensure the smooth handover from the development to operations team

Operations Planning should address how the following will be carried out, and can be included in user procedures, contracts or Service Level Agreements (SLAs):

- Performance Monitoring – Tracking and managing the ability of the asset to meet SLAs for uptime, respond to requests and perform transactions. This also involves anticipating performance issues and initiating preventive actions as required for both application and the associated infrastructure
- Logging and Monitoring – Tracking and alerting for operational and data anomalies that may signal unauthorized activity or potential system failure
- Incident and Problem Management - Ensuring all incidents and problems are closed to restore normal service operation and prevention of additional incidents
- Data Maintenance - Establishing controls and safeguards for data protection, secure handling of storage devices and preservation of records
- Backup, Archival and Recovery - Establishing mechanisms to store copies of data and software code and restoring the application and data in case of failure
- Access Management – Provisioning and de-provisioning user access based on required privileges
- Business Continuity Planning and Disaster Recovery - Establishing mechanisms to restore data and operations in the event of disaster. This involves ensuring that recovery processes can meet established SLAs for data loss and time to return to operation.

5.6 Archival and Retirement

The end of the operational life of an application/system must be managed and follow the appropriate decommissioning process. For all records and information in production systems and any unique records and information existing in non-production systems, a determination of retention requirements and Legal Hold preservation obligations must be completed prior to archival and/or retirement. This review must be conducted by a cross-functional committee including J&J Technology, RIM, the Law Department and J&J-provided legal counsel. The Business Owner is accountable for initiating and ensuring completion of data archival and decommissioning of the system.

A plan for decommissioning should cover the following:

- Rationale for the retirement of the system
- Determination of impact of system retirement to other applications and users
- Record retention strategy based on record retention requirements
- Verification of Legal Holds
- Retirement schedule for the system
- Description of how system specific SOPs and documentation will be retired and archived.

The results of retirement activities including deviations should be summarized with the appropriate evidence post system decommissioning.

6 Document History

6.1 Superseded Document(s)

- **Doc ID:** – N/A **Title:** – N/A **Effective Date:** – N/A

6.2 Revision History

Version	DD-Mmm-YYYY	Author	Change Summary	Major/Minor Change
24.0	10-Oct-2024	Paul Langdon	No changes	Minor
23.0	29-Aug-2022	Paul Langdon	Complete rewrite of document. The SOP was rewritten to represent an activity based SDLC framework instead of the deliverables based framework. Version History from 1.0 to 22.0 have been intentionally removed – they can be found in superseded versions if required.	Major

UNCONTROLLED PRINT

J&J SERVICES, INC.

CONFIDENTIAL - USE PURSUANT TO COMPANY INSTRUCTIONS

Title: Software Development Life Cycle

Signed By: Langdon Paul Decision: Approved Decision Date: 10/10/2024 10:01:17 AM Role: Owner Purpose: Approve Document Meaning Of Signature: As the Owner, I confirm that this document is technically correct and complete, and that the appropriate persons are reviewing/approving this document
Signed By: King Cassandra Decision: Approved Decision Date: 10/11/2024 3:12:51 PM Role: Q&C Functional Approver Purpose: Approve Document Meaning Of Signature: As the Q&C functional approver, I confirm that this document is correct and complete from a quality perspective.
Signed By: Sinha Riya Decision: Approved Decision Date: 10/17/2024 5:24:29 AM Role: Q&C Document Approver Purpose: Approve Document Meaning Of Signature: As a Q&C approver, I confirm that this document complies with Document management process.