Standard Operating Procedure (SOP) for Design Control of Infusion Pumps

- **SOP Number**: SOP-003
- **Title**: Software Development LifeCycle
- **Version**: 01
- **Effective Date**: 15-Oct-2024

1. Purpose

The purpose of this SOP is to define the software development life cycle (SDLC) process for infusion pump software, ensuring compliance with IEC 62304, FDA 21 CFR 820, and ISO 13485. This SOP aims to provide a systematic approach to software development, from planning and design through to implementation, verification, validation, and maintenance.

2. Scope

This SOP applies to all software development activities related to infusion pumps at [Company Name], including embedded software, user interfaces, and any associated software components.

3. Responsibilities

- **Software Development Team**: Responsible for executing the SDLC phases and ensuring software quality.
- **Project Manager**: Coordinates the overall software development process and manages timelines and resources.
- **Quality Assurance (QA) Team**: Ensures compliance with quality standards and conducts software verification and validation.
- **Regulatory Affairs**: Ensures adherence to applicable regulatory requirements throughout the SDLC.
- **Cross-functional Team**: Provides input and feedback during design and review phases.

4. Definitions

- **Software Development Life Cycle (SDLC)**: A structured process for planning, creating, testing, and deploying software.
- **Verification**: The process of evaluating work products to ensure they meet specified requirements.
- **Validation**: The process of evaluating software to determine whether it meets user needs and intended uses.

---

5. Procedure

5.1 SDLC Overview

The SDLC consists of the following phases, as defined by IEC 62304:

1. **Software Development Planning**
2. **Software Requirements Analysis**
3. **Software Design**
4. **Software Implementation**
5. **Software Verification**
6. **Software Validation**
7. **Software Release**
8. **Software Maintenance**

5.2 Software Development Planning Phase

Objective: Define the scope, objectives, and resources for the software project.

Procedure:

1. Conduct a feasibility study to assess the technical and commercial viability.
2. Develop a Software Development Plan (SDP) outlining timelines, resources, and responsibilities, including risk management strategies.
3. Identify regulatory requirements and applicable standards for the software.

Key Decisions:

- Confirm project scope and objectives.
- Approve the Software Development Plan.

Exit Criteria:

- Approved Software Development Plan.

5.3 Software Requirements Analysis Phase

Objective: Gather and document software requirements based on user needs and regulatory standards.

Procedure:

1. Collaborate with stakeholders to gather functional and non-functional requirements.
2. Document requirements in a Software Requirements Specification (SRS).
3. Perform a requirements review with cross-functional teams and obtain approval.

Key Decisions:

- Validate completeness and clarity of requirements.

Exit Criteria:

- Approved Software Requirements Specification.

5.4 Software Design Phase

Objective: Create a design that meets the documented requirements.

Procedure:

1. Develop high-level and detailed design documents, including architecture, interfaces, and data flow.
2. Include design specifications for safety and risk management.
3. Review design with cross-functional teams to ensure alignment with requirements and obtain approval.

Key Decisions:

- Approve design documents for implementation.

Exit Criteria:

- Approved Design Documents.

5.5 Software Implementation Phase

Objective: Develop the software based on the approved design.

Procedure:

1. Write code according to coding standards and best practices.
2. Conduct unit testing during the development process.
3. Maintain documentation for all code changes and utilize version control.

Key Decisions:

- Confirm completion of coding and unit testing.

Exit Criteria:

- Completed software build ready for verification.

5.6 Software Verification Phase

Objective: Verify that the software meets its requirements.

Procedure:

1. Conduct software verification activities, including code reviews, integration testing, and static analysis.
2. Document verification results in the Software Verification Report (SVR).

Key Decisions:

- Approve verification results.

Exit Criteria:

- Approved Software Verification Report.

5.7 Software Validation Phase

Objective: Validate that the software meets user needs and intended uses.

Procedure:

1. Perform validation testing, including system testing and user acceptance testing (UAT).
2. Document validation results in the Software Validation Report (SVVR).

Key Decisions:

- Approve validation results.

Exit Criteria:

- Approved Software Validation Report.

5.8 Software Release Phase

Objective: Release the software into the production environment.

Procedure:

1. Prepare release documentation, including installation and user manuals.
2. Conduct deployment in accordance with the Release Plan.
3. Train end-users on software usage, if necessary.

Key Decisions:

- Confirm readiness for deployment.

Exit Criteria:

- Successful release of software.

5.9 Software Maintenance Phase

Objective: Maintain and support the software post-deployment.

Procedure:

1. Monitor software performance and gather user feedback.
2. Address any bugs or issues identified through monitoring or user reports.
3. Update the software as necessary, following the SDLC for any major changes.

Key Decisions:

- Determine necessity for software updates based on performance data.

Exit Criteria:

- Updated documentation reflecting any changes or bug fixes.

6. Deliverables

Phase	**Key Deliverables**
Planning	Software Development Plan (SDP), Feasibility Study Report
Requirements Analysis	Software Requirements Specification (SRS), Initial Traceability Matrix
Design	Software Architecture Design, Detailed Design Specification, Preliminary Hazard Analysis, Updated Traceability Matrix
Implementation	Source Code, Unit Test Cases and Results, Version Control Records
Verification	Software Verification Plan, Software Verification Report, Final Traceability Matrix
Validation	Software Validation Plan, Software Validation Report, Risk Management Report
Release	Software Release Documentation, Release Plan, Release Approval
Maintenance	Software Maintenance Plan, Bug Tracking Reports, Change Control Records

7. References

- IEC 62304 (Medical Device Software – Software Life Cycle Processes)
- FDA 21 CFR 820 (Quality System Regulation)
- ISO 13485 (Medical Devices – Quality Management Systems)

8. Revision History

Version	Date	Description of Changes	Author
1.0	15-Oct-2024	Initial release	Raj Solai