Software Development Plan

(Template)

GTC Software Engineering

# Contents

# Introduction

## Purpose

Provides overall plan for conducting a software development effort. Describes processes to be followed and determines methods and tools to be used. Serves to communicate procedures and goals to members of the development team.

## Scope

The scope of this guideline is medical device software.

## Typography

The following typograpic conventions are used throughout this document:

|  |
| --- |
| **Warnings**  Warnings are displayed with a red background. |

|  |
| --- |
| **Notes**  Notes are displayed with a yellow background. |

|  |
| --- |
| **Recommendations**  Recommendations are displayed with a green background. |

|  |
| --- |
| **Examples**  Examples are displayed with a gray background. |

|  |
| --- |
| **Template Meta-Information**  Template meta-information, e.g. instructions how to use a template, are displayed with a blue margin line.  Make sure to remove these information when using the template! |

# Software Development Project

## Project Description

The goal of the project is to develop an updated firmware for the Ultra Dialysis Machine to provide additional functionality.

## Project Identification

|  |  |
| --- | --- |
| Project Name | New UDM Firmware |
| Project Number | P4711 |

## Software to Develop

|  |  |
| --- | --- |
| Software Title/Name | Ultra Dialysis Machine Firmware |
| Software Manufacturer | Fresenius Medical Care Deutschland GmbH |
| Software Version | 47.1.1 |

## Software Classification

Based on the Product Risk Analysis, the initial classification of the software is as follows:

|  |  |
| --- | --- |
| Software Safety Class | C |
| Software Level of Concern | Major |

The detailed classification of the software and the justification for the classification can be found in the Medical Device Software Classification.

## Coordination With Product Development

|  |
| --- |
| This section adresses the following required contents:   * References to inputs for software development from product development (e.g. product requirements or component requirements, product architecture, product risk analysis). * References to procedures for coordinating software development with product development such as product integration, verification and validation. * Alignment of major milestones between software development and product development.   Required by milestone: SW-MS2 |

The software development is part of the development of a product. The product development project is performed according to CQMS-SOP-000163 (Process of Product Development).

The following product level documents are input into the software development:

|  |  |
| --- | --- |
| Document Class | Document Instance/s |
| Application Specification | Ultra Dialysis Machine Application Specification |
| Design and Development Plan | Ultra Dialysis Machine Design and Development Plan |
| Product Architectural Design | Ultra Dialysis Machine Product Architectural Design |
| Product Risk Analysis | Ultra Dialysis Machine Product Risk Analysis |
| Product or Component Requirements | Ultra Dialysis Machine Product or Component Requirements |
| Threat Model | Ultra Dialysis Machine Threat Model |

## Roles and Responsibilities

|  |
| --- |
| This section adresses the following required contents:   * Reference to or definition of project-specific software engineering roles. * Assignment of responsible persons to software engineering roles.   Required by milestone: SW-MS2 |

The roles defined for the software development are based on the common role definitions in PEC.MAQ/PEC.MID Role Definitions.

The following table shows the roles and the assigned people for this project, as well as the responsibilities of each role:

|  |  |  |
| --- | --- | --- |
| Role | Assignee/s | Responsibilities |
| Design Quality Assurance | J. Doe | * Justify Deviations From Acceptance Criteria |
| Requirement Engineer | J. Doe | * Define Initial Software Requirements * Trace Initial Software Requirements to Source * Verify Initial Software Requirements * Trace Initial Software Requirements to Software Architecture * Refine Software Requirements * Add Risk Control Measures to Software Requirements * Add Security Control Measures to Software Requirements * Trace Software Requirements to Source * Verify Software Requirements * Trace Software Requirements to Software Architecture * Ensure Traceability of Software Hazards |
| Security Advisor | J. Doe | * Review the Threat Model * Estimate and Evaluate the Security Risks * Define Security Risk Control Measures * Review and Update the Threat Model * Monitor Effectiveness of Risk Control Measures |
| Software Architect | J. Doe | * Define Coding Standards * Define Secure Design Best Practices * Define Software System Context * Define Software Items * Design Secure Software Architecture * Define Interactions of Software Items * Define Segregation * Verify Initial Software Architecture * Refine Software Architecture * Identify SOUP Items * Specify Requirements for SOUP Items * Specify Prerequisites for SOUP Items * Define Granularity of Software Unit * Verify Software Architecture * Document Threats * Ensure Security Capabilities Traceability |
| Software Configuration Manager | J. Doe | * Define Software Development Environment * Protect Software Development Environment * Define Software Integration Approach * Review Software Configuration Management Plan * Plan Software Configuration Management * Identify Software Configuration Items * Provide Cybersecurity Bill of Materials * Define Software Build Process * Integrate Software * Verify Software Integration * Provide Integrity Verification Mechanism * Create Release Candidate Build * Create Master Software Image * Document Software Build * Document Released Version * Document Released Configuration * Assure Reliable Delivery * Archive Software |
| Software Engineer | J. Doe | * Decompose Software Items Into Software Units * Develop Detailed Design of Software Unit * Develop Detailed Design of Interfaces * Verify Software Detailed Design * Implement Software Units * Perform Software Unit Verification * Verify Software Integration Test Protocols |
| Software Risk Manager | J. Doe | * Review Software Risk Management Plan * Plan Software Risk Management * Determine Initial Software Safety Class * Determine Software Level of Concern * Identify Contributing Software Items * Identify Potential Causes for Contribution * Define Risk Control Measures * Determine Software Safety Class * Evaluate Residual Anomalies |
| Technical Project Leader Software | J. Doe | * Define Software Engineering Life Cycle Model * Coordinate between Software Engineering and Product Development * Assign Software Engineering Roles and Responsibilities * Document the Level of Independence * Define Software Deliverables * Define Software Traceability Approach * Define Software Development Standards and Methods * Establish Software Unit Verification Approach * Plan Software Problem Resolution * Define Software Integration Test Approach * Plan Procedures for Avoiding Common Software Defects * Organize Project In Releases * Review Software Development Plan * Define Software Release Scope * Provide Release Candidate * Release Software for Utilization on Product Level * Plan Software Feedback Monitoring * Plan Timely Delivery of Security Updates * Plan Software Maintenance * Plan SOUP Maintenance * Monitor Feedback * Monitor Public Incident Reports * Evaluate Anomaly's Impact on Safety * Evaluate Impact of Changing the Software * Verify Security Updates * Approve Changing the Software * Resolve Anomaly * Communicate Software Change * Communicate Software Security Updates * Deliver Security Updates |
| Verification Engineer | J. Doe | * Specify Software Integration Tests * Specify Security Tests * Specify Threat Mitigation Tests * Specify Vulnerability Tests * Specify Penetration Tests * Perform Software Integration Test |

## Project/Release Planning

|  |
| --- |
| This section adresses the following required contents:   * Information on softwarereleases planned in the course of the project. * Information on how features (or functionality) will be released and delivered.   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on the scope for each softwarerelease (e.g. features or functionalities, bugfixes).   Required by milestone: SW-MS2.1 |

# Software Engineering Life Cycle Definition

## Software Development Process

|  |
| --- |
| This section adresses the following required contents:   * References to standard operating procedures to be taken into account when determining the Software Engineering Life Cycle. * References to standards and norms to be taken into account when determining the Software Engineering Life Cycle. * Reference to or definition of the Software Engineering Life Cycle Model to be used.   Required by milestone: SW-MS2 |

### Standard Operating Procedures

The following standard operating procedures are taken into account to define the software development process:

|  |  |
| --- | --- |
| SOP | Title |
| CQMS-SOP-000174 | Software Life Cycle Processes |
| CQMS-SOP-000163 | Process of Product Development |
| CQMS-SOP-000160 | Design and Development Planning |
| CQMS-SOP-000189 | Process of R&D Risk Management |
| CQMS-SOP-000164 | Process of Design Change |
| CQMS-SOP-000169 | Pre-Production Defect Management |

### Standards

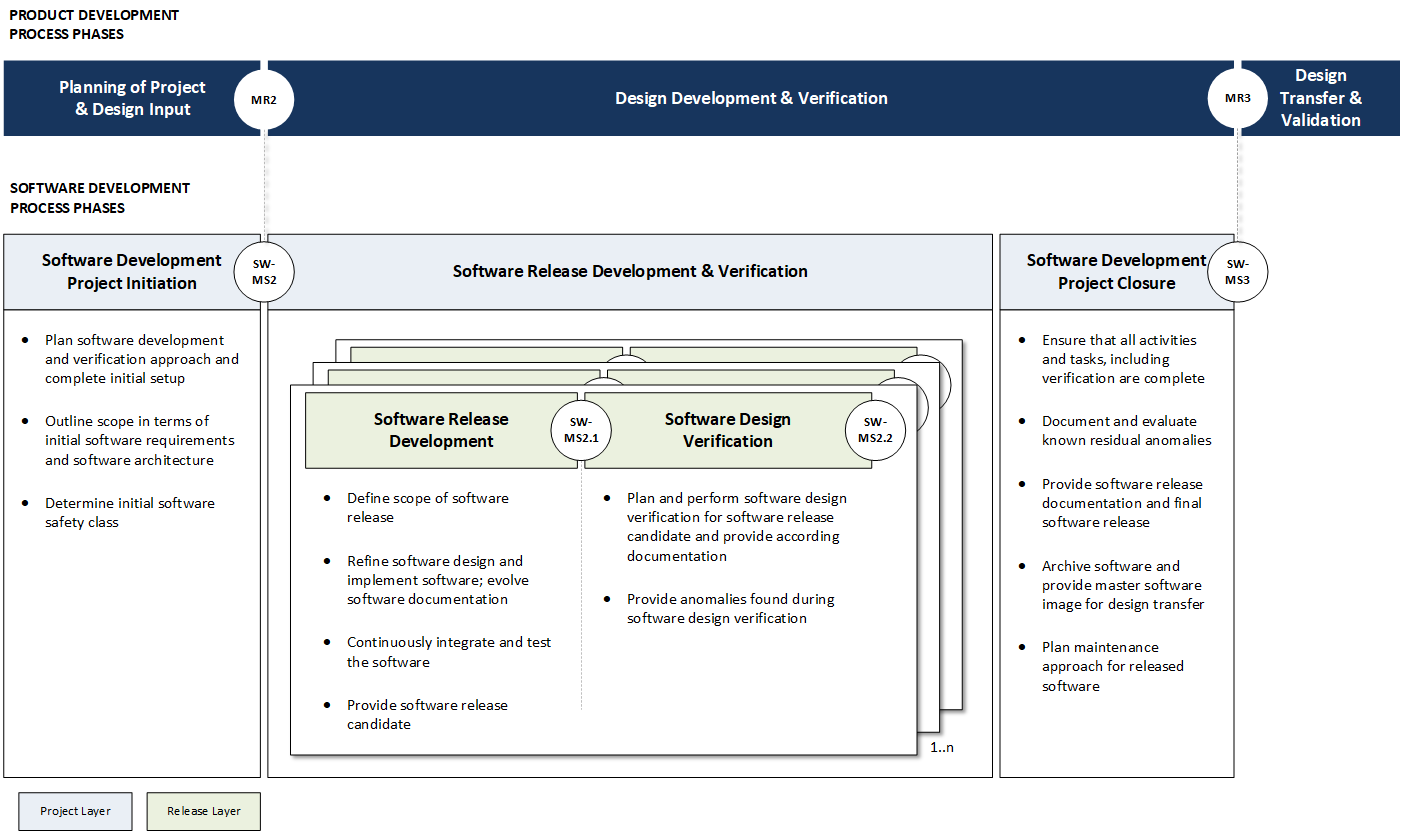
The following standards are taken into account to define the software development process:

|  |  |
| --- | --- |
| Standard | Title |
| IEC 62304:2015 | Medical device software - Software life cycle processes |
| ISO 14971:2019 | Medical devices - Application of risk management to medical devices |
| IEC/TR 80002-1:2009 | Medical device software - Part 1: Guidance on the application of ISO 14971 to medical device software |
| IEC 81001-5-1:2020 | Health software and health IT systems safety, effectiveness and security - Part 5: Security - Part 5-1: Security - Activities in the product life cycle |
| AAMI TIR45:2018 | Guidance on the use of AGILE practices in the development of medical device software |
| FDA GPSV:2002 | General Principles of Software Validation |
| FDA OTS:2019 | Off-The-Shelf Software Use in Medical Devices |
| FDA CONT:2005 | Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices |

### Software Engineering Life Cycle

The software engineering life cycle follows the model defined in the “Software Engineering”.

The following figure provides a brief overview on the Software Engineering Life Cycle spanning across the product development phasesPlanning of Project & Design Inputs and Design Development & Verification of the gPDP:



Software Engineering Life Cycle Overview

The Software Engineering Life Cycle distinguishes two layers in which software engineering activities are executed:

* Project layer: The project layer comprises the complete set of activities needed to deliver a finished software ready for being released for Design Transfer & Validation.
* Release layer: The release layer comprises activities to create a usable software. A software development project is made up of one or more software releases. A software release might generate a software that is or could be released to Design Transfer & Validation, or it might generate a completed, consistent set of functionalities intended only for internal use (e.g. for product integration and/or verification and validation).

In addition to the execution layers, the Software Engineering Life Cycle defines distinct work phases to produce the software in a systematic manner:

* Development Project Initiation: This phase is run through only once and comprises the planning of the software development and verification approach as well as the initial setup (e.g., software development environment, build process, or configuration management) for the software development project. The scope is outlined by an initial software requirements analysis and software architecture. In addition, the initial software safety class of the software system is determined.
* Software Release Development & Verification: This phase comprises the development and verification of the software releases. For each software release, it splits up into the following sub-phases:
* Release Development: Based on the scope of the software release, the software design is further refined and implemented which also includes the advancement of the documentation. The software is continuously integrated and tested on each level of integration. Eventually, a Software Release Candidate is created which is a stable software version that is considered complete, i.e. all features and functionalities planned for this software release have been designed, implemented and integration tested.
* Design Verification: During this phase, design verification activities for the software release candidate are planned and executed formally, i.e. producing formal verification records. These activities typically rely on and use already existing tests prepared in previous phases but may also leverage older verification results. Note that meanwhile, the development of the next software release may start.
* Development Project Closure: In this phase, the final software release (“production release”) is prepared. It is checked that all activities have been conducted and the documentation has been completed. The software has passed all verifications and known residual anomalies are evaluated and considered as acceptable. Handing over the final software release to phase Design Transfer & Validation includes the archiving of the software as well as the creation of a Master Software Image.

## Project Specific Tailoring

|  |
| --- |
| This section adresses the following required contents:   * Information on applicable tailoring criteria (e.g. software safety class, target market, embedded vs. stand-alone software, connected vs. unconnected devices). * Information on process adjustments, including rationale based on tailoring criteria. * Information on intentional process deviations, including rationale why these deviations are acceptable.   Required by milestone: SW-MS2 |

### Tailoring Criteria

The development process is adjusted to meet the demands of the development project. The following criteria guide the adjustments:

|  |  |
| --- | --- |
| Criterion | Characteristic |
| Target Markets | EU, US, China |
| Software Safety Class | C |
| Level of Concern | Major |
| Embedded vs. Standalone Software | Embedded Software |
| Connectivity | Device is not connected to any network |

### Tailoring Decisions

Based on the criteria above, the following process adjustments are done:

|  |  |
| --- | --- |
| Adjustment | Justification |
| No cybersecurityactivities are performed | The Ultra Dialysis Machine has no network connection, so no cybersecurity risks are possible. |

# Methods & Tools

|  |
| --- |
| This section adresses the following required contents:   * Information on standards to be applied during software development (e.g. design guidelines, coding standards). * Information on methods to be applied during software development (e.g. UML, continuous integration, static code analysis, unit testing)   Required by milestone: SW-MS2 |

The following sections contain more detailed information about how to conduct the activities of the software engineering processes.

## Common Software Defect Avoidance

|  |
| --- |
| This section adresses the following required contents:   * Information on common software causes based on programming technologies.   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:  For each identified common software cause:   * Strategy how to avoid the common software cause   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:  For each identified common software cause:   * Strategy used to verify that the avoidance strategy has been applied successfully   Required by milestone: SW-MS2 |

The following subsections describe the approach to avoid common software defects for each programming technology used in the project.

### C++ Programming

|  |  |  |
| --- | --- | --- |
| Common Cause | Avoidance Strategy | Verification Strategy |
| Division by zero | Catch runtime exception | Code Review |
| Uninitialized variable | Initialize every variable at its declaration | Static Code Analysis |

## Software Requirements Analysis

TBD

## Software Architectural Design

|  |  |
| --- | --- |
| Guideline | SOUP Usage |
| Methods | * SOUP Selection * SOUP Qualification * SOUP Integration * SOUP Monitoring |

## Software Detailed Design

|  |
| --- |
| This section adresses the following required contents:   * Information on design best practices to be applied during software development   Required by milestone: SW-MS2 |

Software Detailed Design will make use of to the following:

|  |  |
| --- | --- |
| Guideline | Software Detailed Design |
| Methods | * Model Driven Development * Test Driven Development * Code Centric Development |

## Software Unit Implementation

|  |
| --- |
| This section adresses the following required contents:   * Information on programming technologies to be applied.   Required by milestone: SW-MS2 |

The following sections contain information about the programming technologies being used in the project, and the coding standards applied for each.

### C++ Programming

For software developed in C++, the following coding standards will be applied:

* ISO/IEC 14882:2014 (Programming languages - C++)
* MISRA C++ (Guidelines for the Use of the C++ Language in Critical Systems)
* AUTOSAR C++ 14 Guidelines (Guidelines for the use of the C++14 language in critical and safety-related systems)

## Software Unit Verification

|  |
| --- |
| This section adresses the following required contents:   * Information on strategies, methods and procedures for verifying software units * Information on acceptance criteria for software units * Assignment of methods to acceptance criteria to be verified   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on unit test coverage expected to be reached.   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on rules for the static code analysis   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on rules for the static model analysis   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on rules for the dynamic code analysis   Required by milestone: SW-MS2 |

Software Unit Verification will make use of to the following:

|  |  |
| --- | --- |
| Guideline | Software Unit Verification |
| Methods | * Static Code Analysis * Dynamic Code Analysis * Code Review * Software Unit Testing |

## Software Integration

|  |
| --- |
| This section adresses the following required contents:   * Information on strategies, methods and procedures for integrating software. * Planning of integration of software items (including SOUP) into software systems.   Required by milestone: SW-MS2 |

Software Integration will make use of to the following:

|  |  |
| --- | --- |
| Guideline | Continuous Integration |
| Methods | * Build Automation * Containerization * Virtualization for Continuous Integration * Infrastructure as Code * Pipeline as Code * Artifact Management |

## Software Integration Testing

|  |
| --- |
| This section adresses the following required contents:   * Information on strategies, methods and procedures for software integration testing.   Required by milestone: SW-MS2  Required by milestone: SW-MS2.1 |

## Software Design Verification

Software Design Verification is done according to the Design Verification process defined in the Design Verification.

## Software Release

TBD

## Software Risk Management

Software Risk Management will make use of to the following:

|  |  |
| --- | --- |
| Guideline | Software Risk Management |
| Methods | * Software Safety Classification * Common Software Defects Avoidance * Software Failure Mode and Effects Analysis * Software Fault Tree Analysis * Segregation of Software Items |

## Software Configuration Management

Software Configuration Management will make use of to the following:

|  |  |
| --- | --- |
| Guideline | Software Configuration Management |
| Methods | * Feature Branch * Release Branch * Production Branch * Pull Request * Trunk / Master Only Development Workflow |
| Practices | * Pull Requests with Azure |

## Software Problem Resolution

|  |
| --- |
| This section adresses the following required contents:   * Reference to the overall Pre-Production Defect Management. * Information on software-specific aspects to be considered * Information on handling issues that might occur during software development but do not fall under Pre-Production Defect Management.   Required by milestone: SW-MS2 |

# Software Development Environment

|  |
| --- |
| This section adresses the following required contents:   * Information on tools to be used for software development. * Reference to or planning of toolvalidation.   Required by milestone: SW-MS2 |

## Software Development Tools

|  |
| --- |
| This section adresses the following required contents:   * Information on the tools to be used for static code analysis.   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on the tools to be used for static model analysis.   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on the tools to be used for dynamic code analysis.   Required by milestone: SW-MS2 |

|  |
| --- |
| This section adresses the following required contents:   * Information on the tools to be used for developing the implementation model.   Required by milestone: SW-MS2 |

## Controls for the Software Development Environment

|  |
| --- |
| This section adresses the following required contents:   * Information on controls for protecting the software development environment from unauthorized access, corruption and deletion.   Required by milestone: SW-MS2 |

## Tool Validation

# Software Documentation

## Required by Milestone SW-MS2

|  |  |
| --- | --- |
| Cybersecurity Bill of Materials | |
| A list that includes but is not limited to commercial, open source, and off-the-shelf software components that are or could become susceptible to vulnerabilities.  Content required at milestone:   * Component List | Document Instance/s  Ultra Dialysis Machine Cybersecurity Bill of Materials  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| List of Software Deliverables | |
| Provides all deliverables that will be created during software engineering, together with the milestones at which each deliverable is required.  Content required at milestone:   * List of Software Deliverables * Approval Matrix | Document Instance/sUltra Dialysis Machine List of Software Deliverables  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Medical Device Software Classification | |
| Provides the software safety classes for the software systems of the medical device. For US-marketed medical devices, the level of concern is also provided.  Content required at milestone:   * Software System Safety Classification * Software Item Safety Classification * Software System Level of Concern | Document Instance/sUltra Dialysis Machine Medical Device Software Classification  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| SOUP Documentation | |
| Provides information about SOUP selection, qualification, integration, and monitoring.  Content required at milestone:   * Basic Documentation - SOUP Identification * Basic Documentation - Computer System Specification * Basic Documentation - SOUP Requirements * Level of Concern / Software Safety Class of SOUP | Document Instance/sUltra Dialysis Machine SOUP Documentation  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| SOUP Selection Documentation | |
| Collects the documentation created to select a SOUP item.  Content required at milestone:   * SOUP Selection Criteria * SOUP Evaluation | Document Instance/sUltra Dialysis Machine SOUP Selection Documentation  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Architecture Specification | |
| Provides structural items of the software system and identifies their key responsibilities, their externally visible properties, and the relationship among them. This also includes relationships to hardware and to data flows such as networking.  Content required at milestone:   * Software System Context * Software Structure * Software Behavior * Security Concept | Document Instance/sUltra Dialysis Machine Software Architecture Specification  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Coding Standards | |
| Provides standards for specific programming language regarding recommended programming style, practices, and methods for each aspect of a program written in that language.  Content required at milestone:   * Coding Standards | Document Instance/sUltra Dialysis Machine Software Coding Standards  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Configuration Management Plan | |
| Provides the plan for conducting the activities and tasks of the Software Configuration Managementprocess. Determines items to be controlled and when they are to be placed under configuration control.  Content required at milestone:   * Software Configuration Management Process | Document Instance/sUltra Dialysis Machine Software Configuration Management Plan  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Configuration Status Report | |
| Provides configuration items of a particular software configuration, including their versions.  Content required at milestone:   * Configuration Identification | Document Instance/sUltra Dialysis Machine Software Configuration Status Report  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Development Plan | |
| Provides overall plan for conducting a software development effort. Describes processes to be followed and determines methods and tools to be used. Serves to communicate procedures and goals to members of the development team.  Content required at milestone:   * Software Engineering Life Cycle * Coordination With Product Development * Roles and Responsibilities * Traceability * Development Standards and Methods * Programming Technologies Used * Common Software Causes * Avoidance Strategy for Common Software Causes * Verification Strategy for Avoidance of Common Software Causes * Secure Design Best Practices * Software Unit Verification Approach * Software Problem Resolution Approach * Software Integration Plan * Software Integration Test Approach * Expected Unit Test Coverage * Software Development Environment * Software Development Environment Controls * Selected Tools for Static Code Analysis * Selected Tools for Static Model Analysis * Selected Tools for Dynamic Code Analysis * Selected Tools for Implementation Model * Release Plan * Static Code Analysis Rules * Static Model Analysis Rules * Dynamic Code Analysis Rules | Document Instance/sUltra Dialysis Machine Software Development Plan  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Requirements Specification | |
| Provides the requirements for the software. This typically includes functional, performance, interface, and other requirements for the software. In effect, it describes the observable behavior of the software systems as a black-box.  Content required at milestone:   * Software Requirements | Document Instance/sUltra Dialysis Machine Software Requirements Specification  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Risk Management Plan | |
| Provides the plan for conducting the activities and tasks of the Software Risk Managementprocess, including the management of risks relating to SOUP.  Content required at milestone:   * Software Risk Management Process | Document Instance/sUltra Dialysis Machine Software Risk Management Plan  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Security Risk Analysis | |
| Analysis of the securityrisks associated with the medical device software.  Content required at milestone:   * Software Security Risk Estimation | Document Instance/sUltra Dialysis Machine Software Security Risk Analysis  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Traceability Matrix | |
| Provides softwaretraceability.  Content required at milestone:   * Traces Product Requirements To Software Requirements | Document Instance/sUltra Dialysis Machine Software Traceability Matrix  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

## Required by Milestone SW-MS2.1

|  |  |
| --- | --- |
| Deviation Register | |
| Registers deviations from specified software unit acceptance criteria and provides justifications why these are acceptable.  Content required at milestone:   * Deviations From Acceptance Criteria * Deviations from Static Code Analysis Rules * Deviations from Static Model Analysis Rules * Deviations from Dynamic Code Analysis Rules | Document Instance/sUltra Dialysis Machine Deviation Register  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| SOUP Documentation | |
| Provides information about SOUP selection, qualification, integration, and monitoring.  Content required at milestone:   * SOUP Prerequisites * Re-Qualification Decision * Selected SOUP Qualification Methods * Results of SOUP Verification * Evidence of Common Use Experience * Evidence of Prior Use Experience * Evaluation of SOUP Development Process * Evaluation of SOUP Test Suite * Results of SOUP Code Review * Decision to Use / Not Use the SOUP | Document Instance/sUltra Dialysis Machine SOUP Documentation  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Architecture Specification | |
| Provides structural items of the software system and identifies their key responsibilities, their externally visible properties, and the relationship among them. This also includes relationships to hardware and to data flows such as networking.  Content required at milestone:   * Software Structure * Software Behavior * Software Unit Granularity * SOUP Items * Segregation Concept | Document Instance/sUltra Dialysis Machine Software Architecture Specification  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Archive | |
| Provides the archive of the final software release.  Content required at milestone:   * Released Software | Document Instance/sUltra Dialysis Machine Software Archive  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Build Configuration | |
| Configuration items needed to configure the build system for building the software.  Content required at milestone:   * Build Scripts * Integrated Static Code Analysis * Integrated Static Model Analysis * Integrated Dynamic Code Analysis * Integrated Code Generation | Document Instance/sUltra Dialysis Machine Software Build Configuration  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Build Report | |
| Provides information on the procedure and environment used to create a software build.  Content required at milestone:   * Build Report | Document Instance/sUltra Dialysis Machine Software Build Report  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Configuration Management Plan | |
| Provides the plan for conducting the activities and tasks of the Software Configuration Managementprocess. Determines items to be controlled and when they are to be placed under configuration control.  Content required at milestone:   * Build Tools | Document Instance/sUltra Dialysis Machine Software Configuration Management Plan  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Detailed Design Specification | |
| Provides detailed descriptions of how the software is implemented. Refines the software items into software units. Specifies algorithms, data representations, and interfaces among different software units.  Content required at milestone:   * Software Unit Identification * Software Unit Design * Threat Description * Implementation Model | Document Instance/sUltra Dialysis Machine Software Detailed Design Specification  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Development Plan | |
| Provides overall plan for conducting a software development effort. Describes processes to be followed and determines methods and tools to be used. Serves to communicate procedures and goals to members of the development team.  Content required at milestone:   * Software Integration Test Approach * Release Scope | Document Instance/sUltra Dialysis Machine Software Development Plan  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software FMEA | |
| Bottom-up analysis to identify hazardous situations caused by software failures.  Content required at milestone:   * Scope of the Software FMEA * Software Items Analyzed in Software FMEA * Major Functionalities of Software Items * Potential Failures of Software Item * Prevention / Detection Controls | Document Instance/sUltra Dialysis Machine Software FMEA  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software FTA | |
| Top-down analysis identifying potential software causes for hazardous situations.  Content required at milestone:   * Scope of the Software Fault Tree Analysis * Functional Decomposition * Evaluation of SOUP Anomalies * Potential Causes for Contributions | Document Instance/sUltra Dialysis Machine Software FTA  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Integration Build | |
| Provides a software build of the integrated software that incorporates a specified subset of the functionalities and capabilities of the final software.  Content required at milestone:   * Integrated Software System * Build Log | Document Instance/sUltra Dialysis Machine Software Integration Build  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Integration Test Protocols | |
| Specifies test protocols for software integration tests.  Content required at milestone:   * Software Integration Test Protocols | Document Instance/sUltra Dialysis Machine Software Integration Test Protocols  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Integration Test Record | |
| Records the execution of Software Integration Test Protocols.  Content required at milestone:   * Software Integration Test Results | Document Instance/sUltra Dialysis Machine Software Integration Test Record  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Release Candidate | |
| Provides an operational and installable version of a Software Integration Build intended to be used for Design Verification.  Content required at milestone:   * Release Candidate | Document Instance/sUltra Dialysis Machine Software Release Candidate  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Release Candidate Build | |
| Provides an build of the software that can be used for Design Verification.  Content required at milestone:   * Integrated Software | Document Instance/sUltra Dialysis Machine Software Release Candidate Build  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Release Report | |
| Confirms the completion and correct execution of activities performed for software release (candidate or final), including verification.  Content required at milestone:   * Software Release Checklist * Released Software Deliverables | Document Instance/sUltra Dialysis Machine Software Release Report  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Requirements Specification | |
| Provides the requirements for the software. This typically includes functional, performance, interface, and other requirements for the software. In effect, it describes the observable behavior of the software systems as a black-box.  Content required at milestone:   * Software Requirements * Risk Measures * Risk Measures for SOUP | Document Instance/sUltra Dialysis Machine Software Requirements Specification  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Revision Level History | |
| Registers all software versions released during the course of development, including a brief description of changes.  Content required at milestone:   * Revision History | Document Instance/sUltra Dialysis Machine Software Revision Level History  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Risk Analysis | |
| Provides software items and potential software causes that could contribute to hazardous situations and determines appropriate controls to manage the risks and reduce the impact of hazardous situations.  Content required at milestone:   * Contributing Software Items * Potential Causes * Risk Control Measures | Document Instance/sUltra Dialysis Machine Software Risk Analysis  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Traceability Matrix | |
| Provides softwaretraceability.  Content required at milestone:   * Traces Product Requirements To Software Requirements * Traces Software Requirements To Software Architecture * Traces Software Requirements To Software Units * Traces For Risk Control Measures | Document Instance/sUltra Dialysis Machine Software Traceability Matrix  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Unit Implementation | |
| Provides the implementation of the software units. Represents the starting point for composition of the executable software.  Content required at milestone:   * Source Code * Detailed Design Information | Document Instance/sUltra Dialysis Machine Software Unit Implementation  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Unit Test Scripts | |
| Provides source code of automated software unit tests.  Content required at milestone:   * Software Unit Test Source Code | Document Instance/sUltra Dialysis Machine Software Unit Test Scripts  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Unit Verification Results | |
| Provides the results of the Software Unit Verification to confirm acceptance criteria for software units.  Content required at milestone:   * Software Unit Verification Results * Code Review Findings * Static Analysis Results * Dynamic Analysis Results * Unit Test Results | Document Instance/sUltra Dialysis Machine Software Unit Verification Results  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

## Required by Milestone SW-MS3

|  |  |
| --- | --- |
| Deviation Register | |
| Registers deviations from specified software unit acceptance criteria and provides justifications why these are acceptable.  Content required at milestone:   * Deviations from Static Code Analysis Rules * Deviations from Static Model Analysis Rules * Deviations from Dynamic Code Analysis Rules | Document Instance/sUltra Dialysis Machine Deviation Register  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Feedback on Released Software | |
| Provides feedback received on released software.  Content required at milestone:   * Feedback | Document Instance/sUltra Dialysis Machine Feedback on Released Software  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| List of Residual Anomalies | |
| Lists all known residual anomalies that remain in the final software release, including a rationale why they do not lead to unacceptable risk.  Content required at milestone:   * Risk Evaluation | Document Instance/sUltra Dialysis Machine List of Residual Anomalies  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Master Software Image | |
| Provides an operational and installable version of a Software Integration Build intended to be handed over to Design Transfer & Validation.  Content required at milestone:   * Software Image * Installation Procedure | Document Instance/sUltra Dialysis Machine Master Software Image  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| SOUP Monitoring Record | |
| Record of the monitoring of SOUPanomalies.  Content required at milestone:   * SOUP Anomalies | Document Instance/sUltra Dialysis Machine SOUP Monitoring Record  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Archive | |
| Provides the archive of the final software release.  Content required at milestone:   * Released Software | Document Instance/sUltra Dialysis Machine Software Archive  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Build Report | |
| Provides information on the procedure and environment used to create a software build.  Content required at milestone:   * Build Report | Document Instance/sUltra Dialysis Machine Software Build Report  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Configuration Status Report | |
| Provides configuration items of a particular software configuration, including their versions.  Content required at milestone:   * Status Report | Document Instance/sUltra Dialysis Machine Software Configuration Status Report  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Maintenance Plan | |
| Provides the overall plan for conducting the Software Maintenance for a released software.  Content required at milestone:   * Procedures for Monitoring Feedback * Software Maintenance Used Processes * SOUP Monitoring Intervals * SOUP Information Sources * Security Update Timeframes | Document Instance/sUltra Dialysis Machine Software Maintenance Plan  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Release Report | |
| Confirms the completion and correct execution of activities performed for software release (candidate or final), including verification.  Content required at milestone:   * Software Release Checklist * Released Software Deliverables | Document Instance/sUltra Dialysis Machine Software Release Report  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

|  |  |
| --- | --- |
| Software Revision Level History | |
| Registers all software versions released during the course of development, including a brief description of changes.  Content required at milestone:   * Revision History | Document Instance/sUltra Dialysis Machine Software Revision Level History  AuthorJ. Doe  ReviewerJ. Doe  ApproverJ. Doe  Action   |  |  | | --- | --- | | ☑ Create | ☐ Use Existing | | ☐ Update | ☐ Not Needed | |

# Traceability

|  |
| --- |
| This section adresses the following required contents:   * Reference to the overall design requirements traceability approach * Information on the software-specific traceability approach (e.g. items to be traced, types of traceability). * Information on methods for ensuring traceability between requirements, testing/verification and risk control measures   Required by milestone: SW-MS2 |

Referenced DocumentsFDA Glossary:1995Glossary of Computer System Software Development TerminologyVersion: 08/1995FDA GPSV:2002General Principles of Software ValidationVersion: 01/2002FDA OTS:2019Off-The-Shelf Software Use in Medical DevicesVersion: 09/2019FDA CONT:2005Guidance for the Content of Premarket Submissions for Software Contained in Medical DevicesVersion: 05/2005IEC 61508-4:2010Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviationsVersion: 2010Edition: 2.0IEC 62304:2015Medical device software - Software life cycle processesVersion: 2006+AMD1:2015Edition: 1.1IEC 81001-5-1:2020Health software and health IT systems safety, effectiveness and security - Part 5: Security - Part 5-1: Security - Activities in the product life cycleVersion: TBDEdition: 1IEC/TR 80002-1:2009Medical device software - Part 1: Guidance on the application of ISO 14971 to medical device softwareVersion: 2009Edition: 1ISO/IEC 33001:2015Information technology - Process assessment - Concepts and terminologyVersion: 2015Edition: 1AAMI TIR45:2018Guidance on the use of AGILE practices in the development of medical device softwareVersion: 2012/(R)2018ISO 9000:2015Quality management systems - Fundamentals and vocabularyVersion: 2015Edition: 4ISO 13485:2016Medical devices - Quality management systems - Requirements for regulatory purposesVersion: 2016Edition: 3ISO 14971:2019Medical devices - Application of risk management to medical devicesVersion: 2019Edition: 3IEEE 610.12:1990Standard Glossary of Software Engineering TerminologyVersion: 1990Edition: 1ISO/IEC 14882:2014Programming languages - C++Version: 2014PEC.MAQ/PEC.MID Role DefinitionsRole definitions of PEC.MAQ and PEC.MIDVersion: 1.0 (to appear)CQMS-SOP-000160Design and Development PlanningVersion: 1.0CQMS-SOP-000163Process of Product DevelopmentVersion: 1.0CQMS-SOP-000164Process of Design ChangeVersion: 3CQMS-SOP-000169Pre-Production Defect ManagementVersion: 1CQMS-SOP-000174Software Life Cycle ProcessesVersion: 1.0 CQMS-SOP-000189Process of R&D Risk ManagementVersion: 1MISRA C++Guidelines for the Use of the C++ Language in Critical SystemsVersion: 2008AUTOSAR C++ 14 GuidelinesGuidelines for the use of the C++14 language in critical and safety-related systemsVersion: 2017Continuous IntegrationContinuous Integration (Guideline)Version: 1.0 (draft)Design VerificationDesign Verification (Life Cycle Model)Version: 1.0 (draft)Pull Requests with AzurePull Requests with Azure (Practice)Version: 1.0 (draft)Software Configuration ManagementSoftware Configuration Management (Guideline)Version: 1.0 (draft)Software Detailed DesignSoftware Detailed Design (Guideline)Version: 1.0 (draft)Software EngineeringSoftware Engineering (Life Cycle Model)Version: 1.1Software Risk ManagementSoftware Risk Management (Guideline)Version: 1.1 (draft)Software Unit VerificationSoftware Unit Verification (Guideline)Version: 1.0 (draft)SOUP UsageSOUP Usage (Guideline)Version: 1.1 (draft)Defined TermsActivity A set of one or more interrelated or interacting tasks.

## Standards and Guidances

## Quality Management System

## Guidelines and Best Practices

IEC 62304:2015, 3.1 Work that, when consistently performed, contributes to achieving a specific process purpose and outcome.

ISO/IEC 33001:2015, 3.3.2 An activity usually transforms some portion of process inputs into desired outputs.

FMC interpretationAnomaly Any condition that deviates from the expected based on requirements specifications, design documents, standards, etc. or from someone’s perceptions or experiences.

IEC 62304:2015, 3.2Availability Property of being accessible and usable upon demand by an authorized entity.

IEC 81001-5-1:2020, 3.7Coding Standard A set of guidelines for a specific programming language that recommend programming style, practices, and methods for each aspect of a program written in that language. Component RequirementComponent requirements refer explicitly to a product, these are derived from the product requirements and show the splitting of the product from a technical point of view, ideally split into hardware and software components. Confidentiality Property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

IEC 81001-5-1:2020, 3.8Configuration Item Entity that can be uniquely identified at a given reference point.

IEC 62304:2015, 3.5

IEC 81001-5-1:2020, 3.9Deliverable Required result or output (includes documentation) of an activity or task.

IEC 62304:2015, 3.6Document Information and its supporting medium.

ISO 9000:2015, 3.7.2Dynamic Code Analysis A method to check software by executing it on a real or virtual processor and automatically analyzing the software's execution'. Firmware Permanently installed software in a technical device that is required for its operation. It is installed by the manufacturer of the technical device and cannot be easily changed or exchanged by the user of the technical device.

FMC interpretationGuideline Provides broad advice in following a procedure or process in terms of strategies and methods. Harm Injury or damage the health of people or damage to property or the environment.

ISO 14971:2019, 3.3

IEC 62304:2015, 3.8Hazard Potential source of harm.

ISO 14971:2019, 3.4

IEC 62304:2015, 3.9Hazardous Situation Circumstance in which people, property, or the environment are exposed to one or more hazards.

ISO 14971:2019, 3.5

IEC 62304:2015, 3.35Integrity Property of accuracy and completeness.

IEC 81001-5-1:2020, 3.17Intended Use Use for which a product, process or service is intended according to the specifications, instructions and information provided by the manufacturer.

IEC 81001-5-1:2020, 3.19Level of ConcernMinor Level of ConcernWe believe the Level of Concern is *minor* if failures or latent design flaws are unlikely to cause any injury to the patient or operator.Moderate Level of ConcernWe believe the Level of Concern is *moderate* if a failure or latent design flaw could directly result in minor injury to the patient or operator. The Level of Concern is also moderate if a failure or latent flaw could indirectly result in minor injury to the patient or operator through incorrect or delayed information or through the action of a care provider.Major Level of ConcernWe believe the Level of Concern is *major* if a failure or latent flaw could directly result in death or serious injury to the patient or operator. The Level of Concern is also major if a failure or latent flaw could indirectly result in death or serious injury of the patient or operator through incorrect or delayed information or through the action of a care provider.

FDA CONT:2005, Life Cycle Series of all phases in the life of a product or system, from the initial conception to final decommissioning and disposal.

IEC 81001-5-1:2020, 3.20Manufacturer Natural or legal person responsible for construction activities in the life cycle of medical device software.

IEC 81001-5-1:2020, 3.22Medical Device

Any instrument, apparatus, implement, machine, appliance, implant, in vitro reagent or calibrator, software, material or other similar or related article, intended by the manufacturer to be used, alone or in combination, for human beings for one or more of the specific purpose(s) of

* diagnosis, prevention, monitoring, treatment or alleviation of disease,
* diagnosis, monitoring, treatment, alleviation of or compensation for an injury,
* investigation, replacement, modification, or support of the anatomy or of a physiological process,
* supporting or sustaining life,
* control of conception,
* disinfection of medical devices,
* providing information for medical purposes by means of in vitro examination of specimens derived from the human body,

and which does not achieve its primary intended action in or on the human body by pharmacological, immunological or metabolic means, but which may be assisted in its function by such means.

IEC 62304:2015, 3.11

ISO 13485:2016, 3.7Medical Device SoftwareSoftware system that has been developed for the purpose of being incorporated into the medical device being developed or that is intended for use as a medical device.

IEC 62304:2015, 3.12Method A particular way for accomplishing or approaching something, especially a systematic or established one. Milestone Point in time where the common goals of a phase have been achieved. Model-Based Design Design approach based on designing models using elementary blocks and using these models to automatically generate the source code for embedded systems. Outcome An observable result of the successful achievement of the process purpose.

ISO/IEC 33001:2015, 3.3.11Phase A collection of activities within product development that have common goals and end at a milestone. Procedure A specified way to carry out an activity or a process.

ISO 9000:2015, 3.4.5 Consider a procedure as a high-level definition or a strategic method of control. Procedures may be shared with externals and rarely hold confidential data.

FMC interpretationProcess A set of interrelated or interacting activities that transforms inputs into outputs.

ISO 9000:2015, 3.4.1 A set of interrelated or interacting activities that produce an outcome.

FMC interpretationProduct Output of an organization that can be produced without any transaction taking place between the organization and the customer.

IEC 81001-5-1:2020, 3.24 The product is the actual medical device that contains the medical device software.

FMC interpretationProduct RequirementProduct requirements are requirements related to an explicit product, which describe the requirements under technical aspects. RecordDocument stating results achieved or providing evidence of activities performed.

ISO 9000:2015, 3.7.6 Evidence of performing a process or activity or an output of a process.

Records are used to prove conformity to requirements or specifications and enable to review the effectiveness of an activity and appraise the results against criteria. Typically, templates and forms are used to create records, checklists, or other documents.

FMC interpretationRelease Particular version of a configuration item that is made available for a specific purpose.

IEC 62304:2015, 3.37Residual AnomalyAnomaly still present in a released software. Risk Combination of the probability of occurrence of harm and the severity of that harm.

ISO 14971:2019, 3.18Risk Control Measure Measures by which risks are reduced to, or maintained within, specified levels.

ISO 14971:2019, 3.21 A risk control measure is a measure defined in the device risk analysis.

FMC interpretationRole Set of defined responsibilities for performing activities usually assigned to individuals. Security A state where information and systems are protected from unauthorized activities, such as access, use, disclosure, disruption, modification, or destruction to a degree that the related risks to confidentiality, integrity, and availability are maintained at an acceptable level throughout the life cycle.

IEC 81001-5-1:2020, 3.30Software Programs, procedures, rules, and any associated documentation pertaining to the operation of a system.

FDA Glossary:1995 Note that firmware of an electronic component or programmable hardware (e.g. GAL, FPGA) is not treated as software, if the electronic component or programmable hardware is sufficiently qualified.

FMC interpretationSoftware Defect

Condition that may cause a reduction in, or loss of, the capability of a software item to perform a required function.

(A software defect is an internal cause that may lead to an observable software failure.)

IEC 61508-4:2010, 3.6.1Software Development Environment

Set of tools and other infrastructure required to develop (i.e. design, implement, build and test, etc.) the software.

IEC 81001-5-1:2020, 3.33Software Engineering Life Cycle Conceptual structure spanning the full life cycle of the software which:

* identifies the process, activities and tasks involved in development of a software product
* describes the sequence of and dependency between activities and tasks
* identifies the milestones at which the completeness of specified deliverables is verified

IEC 62304:2015, 3.24

AAMI TIR45:2018, 3.21Software Failure

Inability of a software item to perform a required function.

(A software failure is an externally observable (mis-)behavior of a software item, caused by a software defect.)

IEC 61508-4:2010, 3.6.4Software Item Any identifiable part of a computer program, i.e., source code, object code, control code, control data, or a collection of these items.

IEC 62304:2015, 3.25Software Safety Class (SSC) Designation (A, B, or C) assigned to each software system / software item according to the possible harm on the patient, operator, or other people resulting from a hazardous situation to which the software system / software item can contribute.

FDA GPSV:2002, 4.3Software System Integrated collection of software items organized to accomplish a specific function or set of functions.

IEC 62304:2015, 3.27 The integrated software that is deployed on one processor.

FMC interpretationSoftware UnitSoftware item that is not subdivided into other software items.

IEC 62304:2015, 3.14

1. A separately testable element specified in the design of a computer software element
2. A logically separable part of a computer program.

FDA Glossary:1995Software of Unknown Provenance (SOUP)Software item that is already developed and generally available and that has not been developed for the purpose of being incorporated into the medical device (also known as off-the-shelf software) or software item previously developed for which adequate records of the development processes are not available.

IEC 62304:2015, 3.29 A generally available software item, used by a medical devicemanufacturer for which the manufacturer cannot claim complete softwarelife cycle control.

FDA OTS:2019, IIIStatic Code Analysis A method to check software without actually executing it. In most cases the analysis is performed on the software source code, and in the other cases, some form of the object code. Static Model Analysis A method to check an implementation model developed with Model-Based Design without actually executing it. Strategy A high level plan to achieve one or more objectives based on conscious decisions (e.g., selecting appropriate methods). Task A single piece of work that needs to be done.

IEC 62304:2015, 3.31Template A standardized, pre-formatted file serving as basis for creating a deliverable (e.g. document, report, default configuration). Testing The process of operating a system or component under specified conditions, observing or recording the results and making an evaluation of some aspects of the system or component.

IEEE 610.12:1990, p. 76Tool A computer program that supports the execution of activities and tasks. Traceability Degree to which a relationship can be established between two or more products of the development.

IEC 62304:2015, 3.31Unit Test Coverage A software metric that measures how thoroughly a software unit has been exercised in unit testing.

FMC interpretationUnit Testing

1. Testing of a module for typographic, syntactic, and logical errors, for correct implementation of its design, and for satisfaction of its requirements.
2. Testing conducted to verify the implementation of the design for one software element; e.g., a unit or module; or a collection of software elements.

FDA Glossary:1995 An (automated) test that tests a software unit in isolation against its detailed design.

FMC interpretationValidation Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled.

IEC 81001-5-1:2020, 3.45Verification Confirmation, through provision of objective evidence, that specified requirements have been fulfilled.

|  |
| --- |
| In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirements of that activity.  When the result of the given activity is a document, then the verification is usually done in the form of a document review. |

IEC 62304:2015, 3.33

IEC 81001-5-1:2020, 3.46Vulnerability Flaw or weakness in a system’s design, implementation, or operation and management that could be exploited to violate the system’s security policy.

IEC 81001-5-1:2020, 3.47Weakness Kind of deficiency.

IEC 81001-5-1:2020, 3.48Version History

|  |  |  |
| --- | --- | --- |
| Version | Description | Date |
| 1.0 (draft) | Initial Version. | TBD |

Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
| Author | J. Doe | xxx | 2021-04-27 |
| Reviewer | J. Doe | xxx | 2021-04-27 |
| Approver | J. Doe | xxx | 2021-04-27 |

|  |
| --- |
| Additional reviewer/approvers may be added as desired. Author is responsible for ensuring all required approvers have reviewed and approved this document. |

Template Version History (to be removed)

|  |  |  |
| --- | --- | --- |
| Version | Description | Date |
| 1.0 (draft) | Initial version. | 2021-04-27 |