10 QMS Musts for SaMD and Devices Featuring Software

AUGUST 5TH, 2021

greenlight guru

RQS

MEDICAL DEVICE QUALITY IS ALL WE DO, AND WE'RE ALWAYS AHEAD OF THE GAME.

75
years industry experience

275k
podcast listeners

#1
blog and podcast in the industry

114k
look to us for the latest in quality

FEATURED IN

THE WALL STREET JOURNAL.

THE VERGE · TechCrunch · Forbes · QUALITYDIGEST

MDDI · Inc. · MedTech Intelligence · MED DEVICE ONLINE

MedicalDesign & OUTSOURCING · TNW THE NEXT WEB · Entrepreneur · MPO

G2 | CROWD

LEADER

QMS SOFTWARE

★★★★☆

Since Winter 2019

"Best eQMS I have ever used..."

This is the easiest eQMS I have used in the 20 years I have been in the Medical Device Industry. ***It is simple, intuitive and easy to use...*** We are successfully implementing a Quality Culture.

- Director of Regulatory Affairs
& Quality Assurance

"Modern QMS Software and Outstanding Customer Service."

★★★★★

"Demystifying QMS and Regulatory Requirements"

★★★★★

"Makes your QMS Simple and Effective"

★★★★★

greenlight guru

Rook Quality Systems

Rook Quality Systems is a consulting firm dedicated to helping startup to mid-sized medical device companies develop and maintain effective and efficient quality systems.

We provide specialized and custom consulting services for all classes of medical devices, including medical software and combination devices.

- Quality System Design
- DHF/ TF Creation
- Audit Support
- Software Validation
- Design Control
- Risk Management
- Regulatory Submission Support (Int'l)
- Quality System Training

Presentation Outline

1. How a software device differs from a traditional medical device
2. Overview of regulations related to software devices
3. Why it's important to start documentation early in the process
4. How to exclude things that don't apply to SaMD
5. Making the QMS efficient
6. Capturing risk analysis for SaMD
7. Documenting the Software Development Plan
8. Identifying requirements
9. Managing changes and new software releases
10. How to properly document your testing

Key Differences with SaMD

- **Some aspects of the standards do not apply to the QMS of SaMD companies**
  - Things like manufacturing, incoming inspection, monitoring devices, process validation, storage are not applicable and can be confusing when creating a QMS
  - IT infrastructure and change management can be particularly important

- **Important to know the quality and regulatory requirements for your device and how to properly document the development, design, and testing of your SaMD**
  - Ensure performance of SaMD on various OTS computing platforms via V&V activities
  - Ensure cybersecurity risks are adequately addressed
  - Ensure change control and version control processes are followed

SaMD Definition (IMDRF N10)

Software intended to be used for **one or more medical purposes** that perform these purposes **without being part of a hardware medical device**.

Intended Use of SaMD (IMDRF N12)

To identify the intended medical purpose of SaMD,

- Treat or diagnose
- Drive Clinical Management
- Inform Clinical Management

To state the healthcare situation or condition that the SaMD is intended for

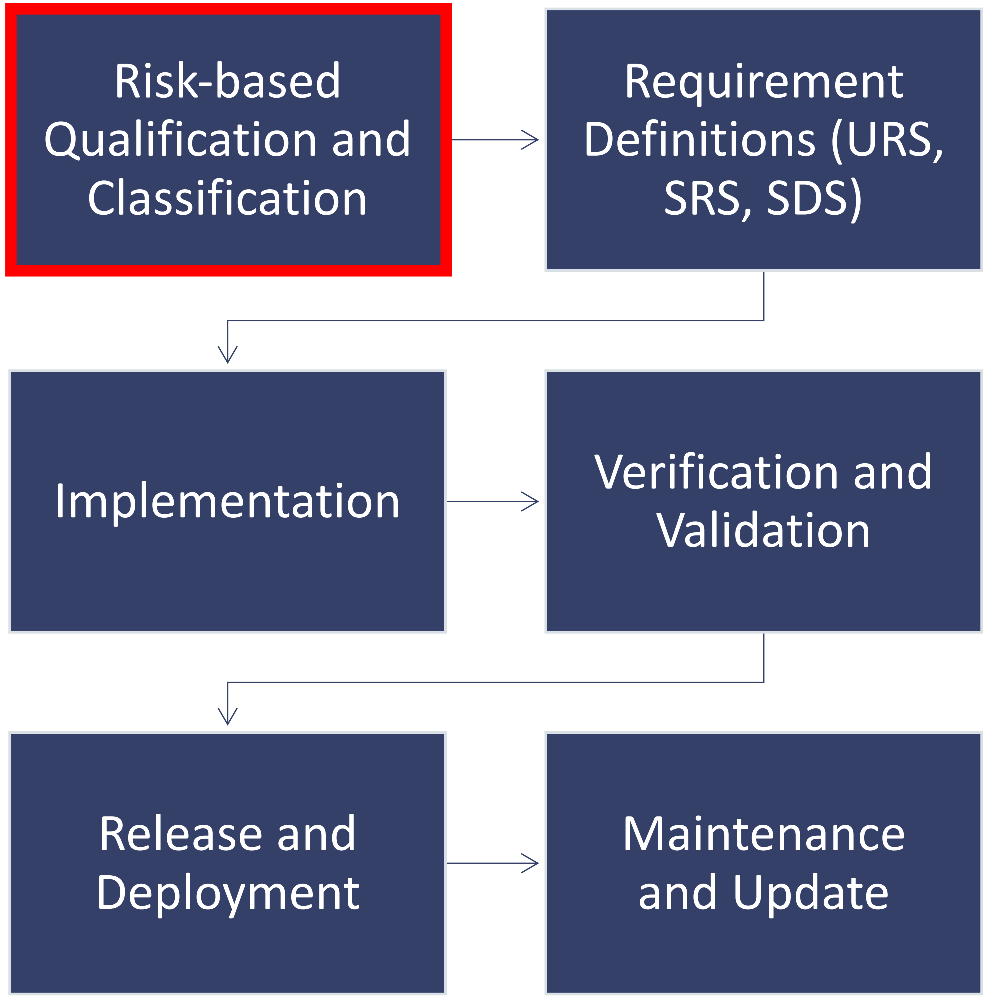
- Critical
- Serious
- Non-Serious

To describe the core functionality of SaMD

Key Regulations

- IEC 62304/82304
- AAMI TIR45
  - TIR45 provide clarity on mapping Agile practices to regulatory requirements
  - US FDA recognized AAMI TIR45 in Jan 2013
- ISO 13485:2016
- FDA 21 CFR 820
- ISO 14971
- 21st Century Cures Act
- IMDRF N10 SaMD Definitions
- IMDRF N12 SaMD Risk Categorization

Regulatory Strategy inform software development strategy



Risk-based Qualification and Classification

Requirement Definitions (URS, SRS, SDS)

Implementation

Verification and Validation

Release and Deployment

Maintenance and Update

Risk-based Qualification and Classification

- CDS Guidance
- IMDRF N10 SaMD Definitions
- IMDRF N12 SaMD Risk Categorization
- Risk Analysis (e.g. PHA or FMEA)
- Most importantly…talk to your customers! Understand the consequences and implications when SaMD malfunctions

Level of Concern FDA

- Determine the documentation requirements for the SaMD by Major, Moderate, and Minor LoC
- Level of Concern should be driven by the hazard analysis in the absence of mitigations, regardless of the effects of the mitigations on the individual hazards

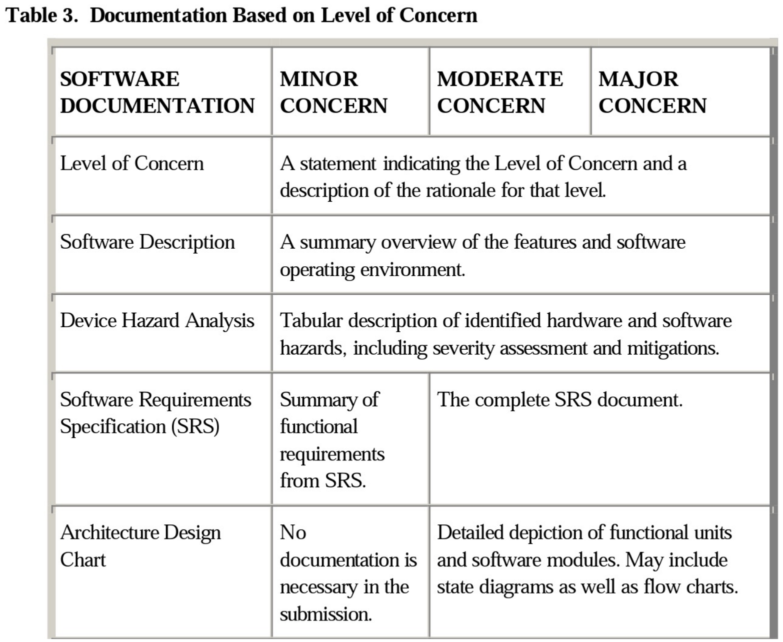
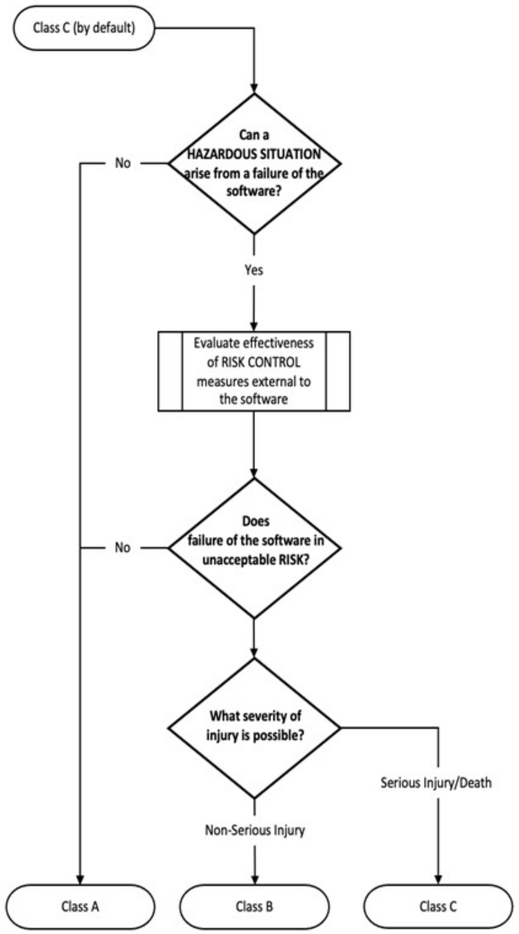


Table 3. Documentation Based on Level of Concern

SOFTWARE DOCUMENTATION	MINOR CONCERN	MODERATE CONCERN	MAJOR CONCERN
Level of Concern	A statement indicating the Level of Concern and a description of the rationale for that level.		
Software Description	A summary overview of the features and software operating environment.		
Device Hazard Analysis	Tabular description of identified hardware and software hazards, including severity assessment and mitigations.		
Software Requirements Specification (SRS)	Summary of functional requirements from SRS.	The complete SRS document.	
Architecture Design Chart	No documentation is necessary in the submission.	Detailed depiction of functional units and software modules. May include state diagrams as well as flow charts.	

Software Safety Classification IEC 62304


Class C (by default)
Can a HAZARDOUS SITUATION arise from a failure of the software?
No
Yes
Evaluate effectiveness of RISK CONTROL measures external to the software
Does failure of the software in unacceptable RISK?
No
What severity of injury is possible?
Serious Injury/Death
Non-Serious Injury
Class A
Class B
Class C

a) The MANUFACTURER shall assign to each SOFTWARE SYSTEM a software safety class (A, B, or C) according to the RISK of HARM to the patient, operator, or other people resulting from a HAZARDOUS SITUATION to which the SOFTWARE SYSTEM can contribute in a worst-case scenario as indicated in Figure on the left

Beginning Software Documentation

QMS

vs

DHF

(you need both)

QMS includes high level procedures and templates that define how you manage the product and quality

DHF/ TF Creation

Design History File includes activities related to requirements, coding, testing, and risk analysis

Starting Documentation

- Most early stage and startup companies have started the DHF without any defined or documented practices on
  - What they are creating
  - How they are developing the code
  - How they are tracking software updates and bug fixes
  - What the risks of the software are
  - How it will be tested
  - What the acceptance criteria are

"We tested it, but we didn't document the testing or results"

Implementing a Pre-Market QMS

- Identify the QMS documents required to get your device ready for regulatory review
- Review requirements outlined in FDA/ISO/IEC and 510(k) requirements to build foundation of the QMS
- Focus mainly on design and risk activities
- Recommended documents for Pre-Market QMS for SaMD
  - **Quality Manual**
  - **Document Control/Quality Records**
  - **Software Design and Development**
  - **Risk Management**
  - **Personnel and Training**
  - **Supplier Assessment**
  - **Configuration Management**
  - **Cybersecurity**

Implementing a Pre-Market QMS

- **Quality Manual**
  - Outlines the scope of the company and should define the nature of the software device(s)
  - Provides exclusions for the aspects of 21 CFR and ISO 13485 that do not apply. Justifications should be included
- **Document Control/Quality Records**
  - Defines how SOPs and records are created and maintained
  - Outlines use of eQMS for Doc Control process
- **Personnel and Training**
  - Outlines how employees will be trained in QMS
  - Should include requirements for external contractors or developers

Implementing a Pre-Market QMS

- **Risk Management**
  - Defines the process and timeline for risk analysis of the SaMD
  - Should incorporate requirements from IEC 62304 and ISO 14971
  - Level of Concern and Software Safety Classification templates can be incorporated
  - Risk must be incorporated in traceability matrix and testing process
  - P1/P2 method recommended per IEC 62304 for analyzing software bugs

- **Supplier Assessment**
  - Define requirements for any external service providers or developers
  - Determine how they will be tracked and evaluated and if training in the QMS is needed
  - Document external tools or libraries used in development as suppliers
  - Track supplier performance throughout development

Risk Management

- Risk management will span across the entire life cycle of the software
  - Prior to release
    - The risk management process should start early with risk analysis to guide implementation of controls to reduce risk
    - During software testing, defects that arise need to be evaluated for risk criticality to prioritize fixes
  - After release
    - Risk management process needs to be re-applied to future updates with the same outputs as prior to release

Development Strategy

Development Strategy	Description	Intput/Output	Risks
Waterfall	Performing the development process a single time, or so called “one-through” strategy.	Inputs must be complete and approved before implementation	Delaying the start of build process
Incremental / Kanban	Performs sequence of builds with defined customer needs and system requirements.	Permit outputs to be produced before all the inputs are available (e.g. software item can be implemented and verified before software architecture is finalized)	Change or development in one output might invalidate another output.
Evolutionary	Performs sequence of builds with customer needs and system requirements partially defined up front, then are refined in each succeeding build.		

Implementing a Pre-Market QMS

- **Design and Development**
  - Procedure should outline requirements and phases of design.
  - Should incorporate relevant standards including 62304, AAMI TIR45
  - Should reference additional templates or plans including
    - Software Development Plan
    - Software Detailed Description/Architecture
    - Software V&V SOP/Plan
    - Traceability Matrix
  - Procedure should describe process and timeline for design reviews
  - Procedure should document how Software of Unknown Provenance (SOUP) is tracked, tested, and reviewed

Tips for Efficiency – Design Process

- Utilize practices and processes that are already implemented or familiar to your team
  - Incorporate tools that may be used or high-level practices that have already been implemented by the development team
    - JIRA
    - Confluence
    - Trello
    - Automated testing tools
    - Git repositories
  - All of these can still be used for medical device but need to be formally incorporated in your process.
  - The key is to get the team to buy in to documentation but not overwhelm with new processes or slow down development

Software Development Plan

- Plan should be created to outline and track development process
- Recommended stages

I. Software development schedule (phases/sprints)

II. Software design process

  a) Code requirements and platform

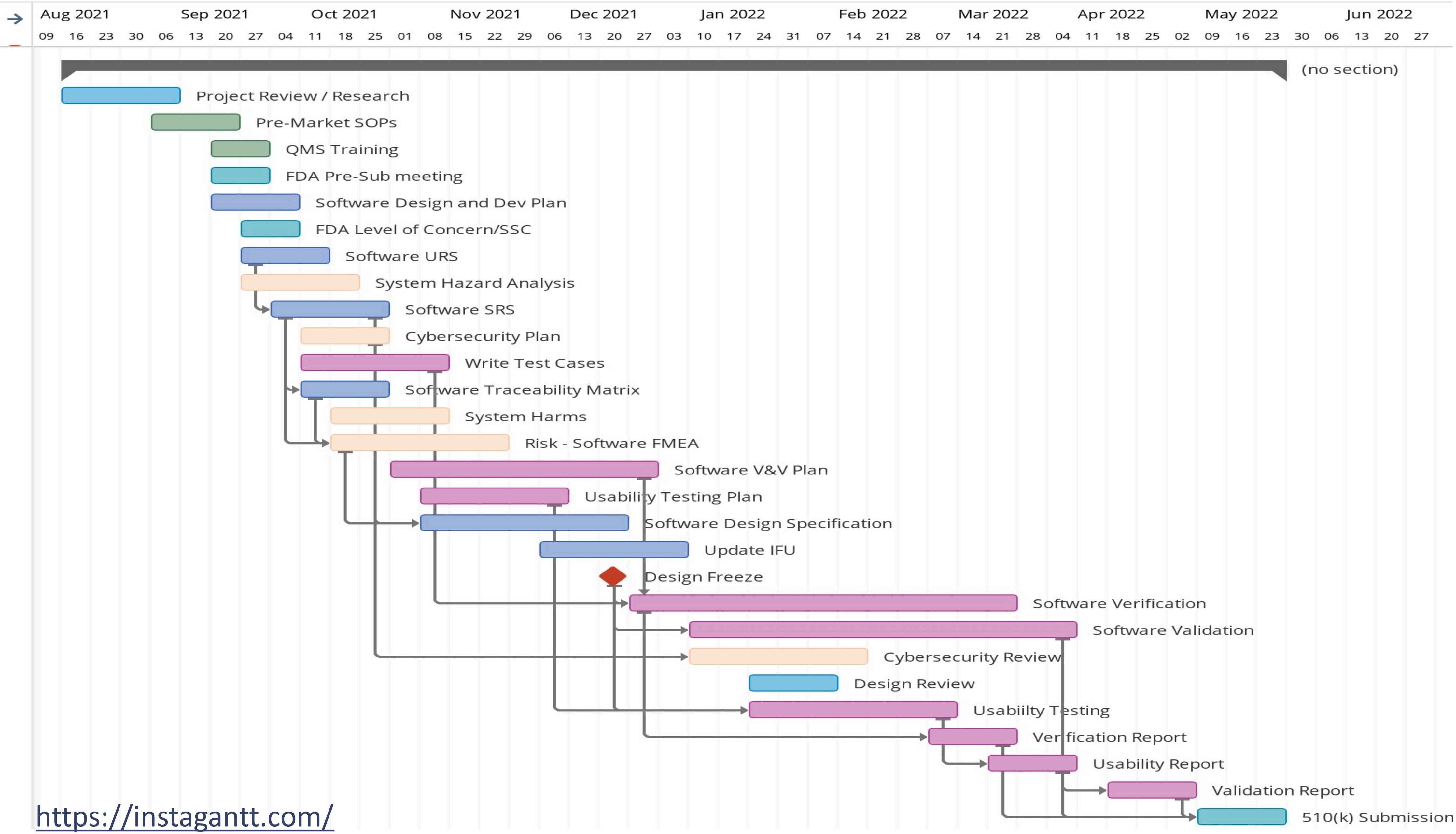
  b) Architecture description and/or diagram

III. Software implementation

  a) Implementation progress should be tested, and the bugs fixed during implementation process

IV. Software Testing

V. Software Maintenance



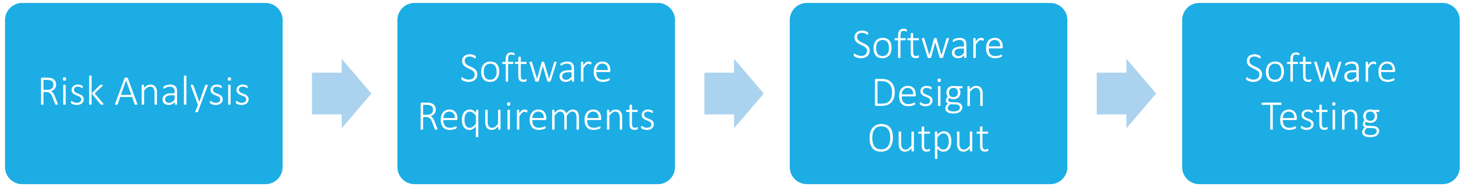
https://instagantt.com/

Requirement Definitions

- Requirements/Specifications must be defined early in the process.
  - Typically, these are separated between User Requirement Specification (URS) and Software Requirement Specification (SRS)
- Software requirements examples from 62304:
  - Performance (e.g., purpose of software, functionalities),
  - Implementation (e.g., code language, platform, operating system),
  - Cybersecurity
  - Usability
  - Maintenance/Update

Traceability Matrix

- Each defined requirement needs to have established traceability to identified risks, the outputs of the software design process, and subsequent testing
  - Risk analysis will help drive requirement definition
  - Design output will demonstrate implementation of software requirements
  - Testing will verify correct implementation of the software requirements


Risk Analysis

Software Requirements

Software Design Output

Software Testing

Release Management

- Quality Team is responsible for coordinating with the DevOps to enact and monitor the continuous integration and delivery in the DevOps pipeline.
- Setup a release management system and follow through with that system.
  - Clarify business needs
  - Release planning
  - Build/test/deploy
  - Plan for the future
- Scaling the release management process means using tooling to:
  - Release checklist
- Standardize procedures across the organization
  - Get the visibility needed to understand dependencies

QMS impact during release

- Core DHF (require revisions)
  - Product Requirements Document
  - Regulatory Assessment and Plan
  - Requirement Traceability Matrix
  - Risk Management File
  - Software Requirement Specification
  - Validation Planning

QMS impact during release

- Release DHF (newly generated)
  - Validation and Test Plan
  - Test Report
  - Maintenance Release Plan
  - Release Approval
  - Deployment Plan
  - Final Design Review Minutes
  - Release Approval

Documenting Software Verification Testing

- Test plans and test cases should be created as early as possible in the software development process
  - Unit and integration testing
  - Functional testing
  - Regression testing
- Test Reports
  - Results
  - Deviations
  - Unresolved defects
  - Version tested

Software Change Management

- Pre-release Design Change
  - Issue CR and identify source of feedback (e.g. internal sources, external sources)
  - Evaluate impact of changes to requirements, risk management files, IFU, usability, etc.
  - Regression testing
  - Release Approval
- Configuration Management
  - SaMD itself
  - Embedded OTS libraries

Upcoming RookQS Events

- Project Medtech Podcast
  - https://projectmedtech.com/
- 2021 European Medical Device & Diagnostic Software Development and Compliance Virtual Conference.
  - October 5-7th
  - https://www.medtechsoftwareengineering.com/
- AdvanSE Life Sciences Conference
  - October 27-29th
  - https://southeastlifesciences.org/2021-conference/

Questions?

www.RookQS.com

Make sure to visit out website to learn more about our services and consulting team.

Contact info@rookqs.com for more questions, comments, or to set up a meeting. One of our consultants will be sure to reach out to assist!