Harmonization of Agile Software Development and FDA Medical Device Design Control Requirements for SaMD

Presenter:

Lori-Ann Woodard CBA, CQE, CSQE
Medical Device Director, Compliance Practice
Lachman Consultant Services, Inc.

Lachman CONSULTANTS™

Experience. Excellence.™

Legal Notice

The information displayed on these presentation slides is for the sole private use of the attendees of the seminar/training at which these slides were presented. Lachman Consultant Services, Inc. (“Lachman Consultants”) makes no representations or warranties of any kind, either express or implied, with respect to the contents and information presented. All original contents, as well as the compilation, collection, arrangement, and assembly of information provided on these presentation slides, including, but not limited to the analysis and examination of information herein, are the exclusive property of Lachman Consultants protected under copyright and other intellectual property laws. These presentation slides may not be displayed, distributed, reproduced, modified, transmitted, used or reused, without the express written permission of Lachman Consultants.

Opening Remarks

SUMMARY:

- The goal of this presentation is to provide a strategy to develop SaMD (Software as a Medical Device) in compliance to the FDA Design Control requirements
- Step through 21 CFR 820.30 Design Control requirements and Agile development principles in parallel and present a least burdensome approach
- Discuss the importance of having a quality management system and the right resources to support SaMD development and post market maintenance

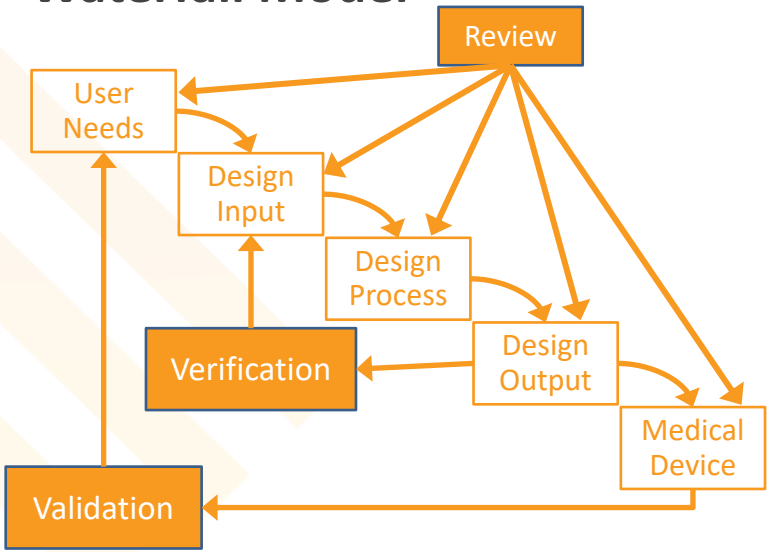
REFERENCES

- 21 CFR 820 Quality System Regulations for Medical Devices
- Association for the Advancement of Medical Instrumentation, AAMI TIR45:2012 Guidance on the use of AGILE practices in the development of medical device software, 20 August 2012
- International Electrotechnical Commission, IEC 62304:2006 Medical device software — Software life cycle processes, May, 2006
- International Medical Device Regulators Forum, IMRDF Software as a Medical Device (SaMD): Application of Quality Management System, 2 October 2015

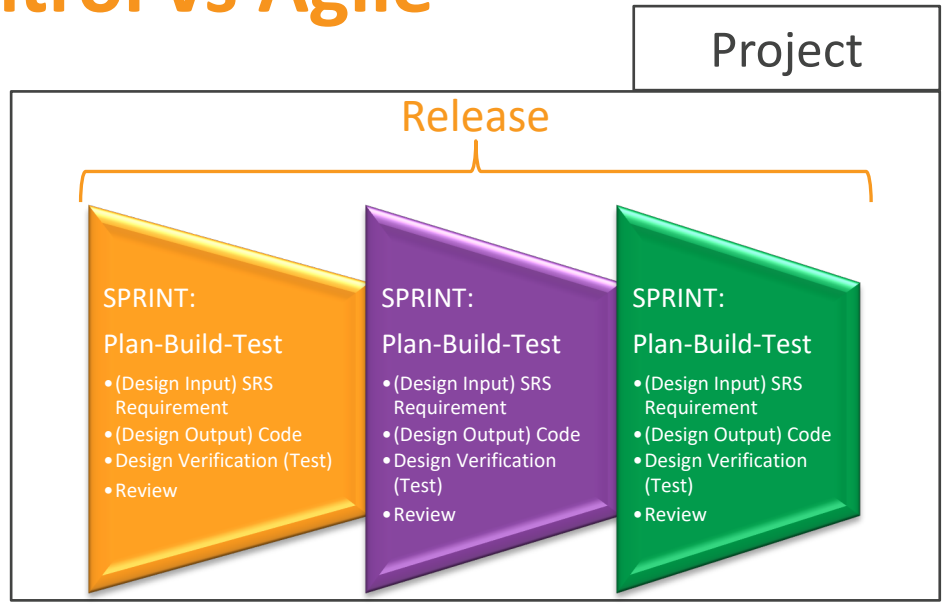
To begin, lets briefly discuss the differences between design control methodology and a common type of software development approach

Design Control vs Agile

Waterfall Model



- Design Control is a linear model with the establishment of high level user and product requirements that are developed first then cascades down to the testing and user validation

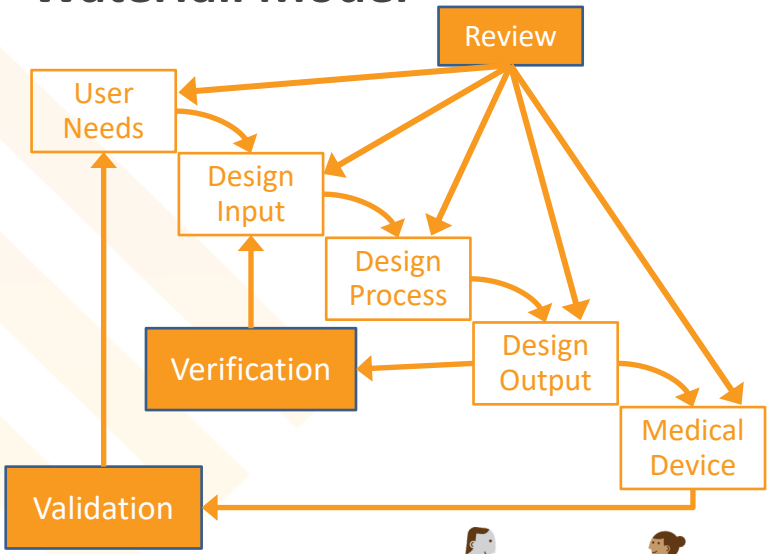


- Agile methodology is “A software development technique in which requirements definition, design, implementation and testing occur in an overlapping, iterative (rather than sequential) matter resulting in incremental completion of the overall software product*”

*Association for the Advancement of Medical Instrumentation, AAMI TIR45:2012 Guidance on the use of AGILE practices in the development of medical device software, 20 August 2012

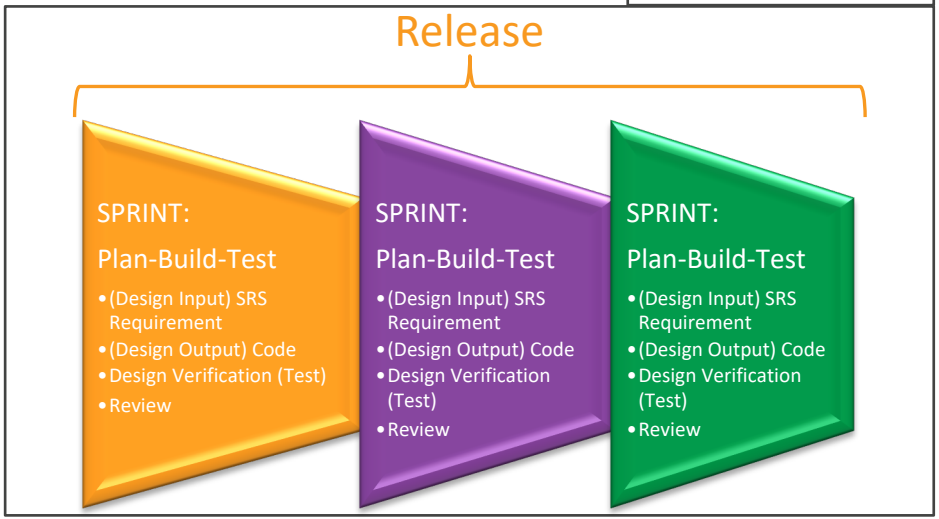
Design Control vs Agile

Waterfall Model



Relay Race
(linear)

Project



Rugby
(simultaneous)

Design Planning
Design Control vs Agile

- **Description**
- **Intended Use Statement**
- **Roles and Responsibilities**
- **Quality Management System that governs the process (list procedures)**
- **Development Process**
  - Describe the goals and deliverables and include timelines and scheduling
  - Risk Management Planning
  - Explain tool to use Traceability of requirements through verification and validation
    - Traceability Matrix is an excellent tool – explain when it will be updated
  - Human Factors Planning
  - Quality Planning – explain the QMS and reference procedures
  - Design Verification Planning
  - Design Validation Planning

Along with the elements listed to the left, you will need to include, at a minimum (ACCEPT EVOLUTION OF PLANNING THROUGHOUT THE DESIGN PROCESS):

- **Software Development Process**
  - Describe Agile process and structure (user stories, sprints, releases, etc.)
  - Software Architecture Design
    - Software components and interfaces between the components
    - Include external interfaces (hardware and human interfaces)
  - Align hard stops for design reviews and V&V activities with architectural design elements
  - Key is to manage a complex process
  - Note that this may change over time – explain how architecture/planning document will be updated accordingly
  - Describe the goals and deliverables and include timelines and scheduling
- **Tools used for testing and Risk Management**
- **Release Planning**
- **Configuration Management (how will changes be captured and traced?)**
- **Commissioning and Decommissioning of Software**

Design Inputs
Design Control vs Agile

- **Most of the User Requirements and Product Requirements are captured up front at the beginning of the process**
  - Use Requirements
  - Environmental Requirements
  - Dimensional Requirements
  - Functional Requirements
  - Regulatory Requirements
  - Etc.
- **Design Inputs can be traced forwards and backwards through design validation using a Traceability Matrix**

- **Software requirements using Agile methodology are determined throughout the iterative process and are not always determined up front but start with (not limited to) these, e.g.*:**
  - Functional and Capability Requirements
  - Usability Engineering Requirements
  - Cybersecurity Requirements
  - Alarms, warning and operator messages
  - Installation and servicing
  - Database/platform requirements
  - System interfaces
- **At the end of the determined stops at sprints and releases, all software requirements need to be rolled up to the high level system requirements**
- **Use a Software Traceability Matrix (STM) to document requirements rolled up to high level system requirements**

Design Outputs
Design Control vs Agile

- **Design Outputs are determined from Use and Product Requirements. Examples are:**
  - Part, assembly, and finished product drawings
  - Production, process and product specifications
  - Component and material specifications
  - Work instructions
  - Quality assurance specifications, procedures
  - Packaging and labeling specifications
  - Developed from design inputs and risk mitigations

- Software outputs are determined with software requirements (inputs)
  - **Code** corresponding to requirements
  - **Code** corresponding to Risk Mitigation
- Make sure you have a system to document and trace code. Document on the STM
- **Determine hard stops ("doneness")** at which point in the process when the STM must be updated

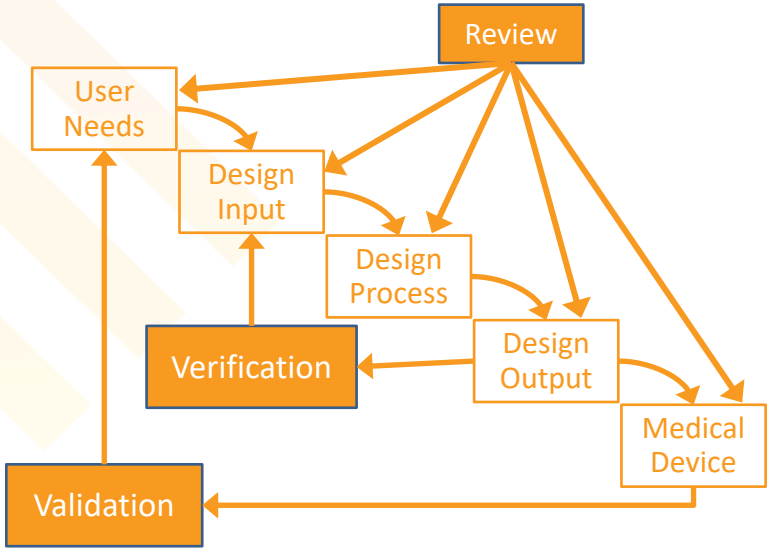
Design Verification & Validation
Design Control vs Agile

- **Verification**
  - Functional tests, Inspections and analysis to satisfy design inputs and risk mitigation
- **Validation**
  - User tests to validate applicable design input use requirements
  - Human Factors Summative studies
  - Risk Management Planning and Execution

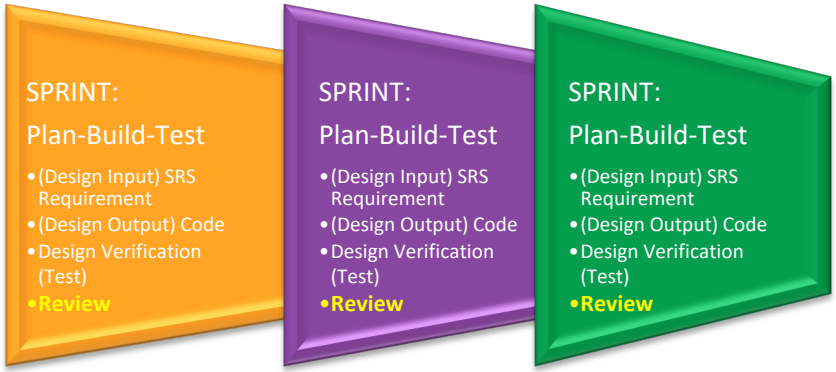
- **Describe all verification and validation activities in each layer throughout the iterative design process**
  - Determine what toolValidation Plan should be written to explain how and when the iterative and final validation of the product shall be dones are needed to perform verification testing
  - Tools need to be qualified prior to use
  - Installation and Uninstallation testing should also be included
- **Validation Plan should be written to explain how and when the iterative and final validation of the product shall be done**
  - Include Human Factors Usability Testing which may be covered under a separate plan
- **Risk Management Planning and Execution**
  - How will defects be captured and mitigated throughout the iterative process?
  - Use STM Traceability Matrix tool

Design Reviews
Design Control vs Agile

- **Design Reviews are traditionally held at the end of all or specific phases in the development process**



- **Under Design Planning, determine when and how many design reviews will be held**
  - Sprint Reviews should be documented however the official Design Review can be designated at with the ”hard stop” or “done stops” determined at critical points indicated in the system/software architecture
- **Make sure the ‘independent reviewer is trained and understands the software development process**



Design Transfer & Release
Design Control vs Agile

- **Design Transfer typically consists of:**
  - Design Transfer Plan
  - Validation Master Plan **(equipment and manufacturing process)**
  - Device Master Record which includes a list of the final "recipe" (design outputs)
  - Risk Management Completed
  - Human Factors Completed
  - Traceability Matrix is finalized with all references
  - Final Design History File
  - Final Design Review to verify all elements of the design control process is satisfied and allow release product to Market

- **Design Transfer Activities should consist of:**
  - Design Transfer Plan
  - Validation Master Plan **(final software product)**
  - Device Master Record which includes a list of the final **"recipe"** or **code**
  - Risk Management Completed
  - Human Factors Completed
  - Final Software Traceability Matrix is finalized with all references
  - Final Design History File **and/or DHF index**
  - **Deployment Process**
  - Final Design Review to verify all elements of the design control process is satisfied and allow release product to Market

Design Changes & Post Market Maintenance
Design Control vs Agile

- **Change Control**
  - All changes must be verified and/or validated to ensure there is no impact to the product
  - **Risk management** is closely tied into change control to determine/categorize the risk of the change
  - All changes are managed via a change control document system
- **Customer Complaints, CAPAs, Field Events, Recalls**
  - Responsible SMEs track, investigate and file with the FDA

- **Configuration Management** strategy must be determined (via procedure)and expressed in the Design Planning Document
  - How will software configuration files be stored and updated?
  - Traceability of change management is key
- **Change Control Tools**
  - **Automated systems and tools** used for change control must be validated
  - **Regression testing** to ensure that previously developed and tested software still performs after a change
  - Software maintenance (Post market Releases)
- **Software Problem Resolution, Help Desk, etc.**
- **Customer Complaints, CAPAs, Field Events, Recalls**
  - Need an SME on FDA requirements to ensure these systems are robust
- **BE PREPARED!!!**

Final Remarks

- **It is important to note that the 21 CFR 820.30 Design Control regulation requires that companies have procedures on the development process.**
- **All suggestions stated here, if implemented will require procedures and forms to document all activities because, if it was not documented, it is considered not done**
- **A good SME will be able to review FDA Regulation, Guidance, and IEC standards to generate a least burdensome approach for compliance to the FDA Design Control requirements**
- **Finally, the Development process must be embedded in a wholistic quality management system for SaMD compliance. In this presentation, we only covered one element of 21 CFR 820.**
  - Best practice would be to map out your quality management system procedures to applicable elements of 21 CFR 820 Quality System Regulation along with ISO 13485 (if applicable) and IEC 62304.
  - Mapping example: Use IMRDF Software as a Medical Device (SaMD): Application of Quality Management System, Appendix A as an example:
  - http://www.imdrf.org/docs/imdrf/final/technical/imdrf-tech-151002-samd-qms.pdf

THANK YOU!

Lori-Ann Woodard, CBA, CQE, CSQE
Medical Device Director, Compliance Practice
Lachman Consultant Services, Inc.
1600 Stewart Avenue, Westbury, NY 11590 (USA)
Telephone: 516-416-8277
l.woodard@lachmanconsultants.com
www.lachmanconsultants.com

Lachman CONSULTANTS™

Experience. Excellence.™