



# Adopting Agile in Highly Regulated Medical Device Industry – Myth, Realities, and Implementation

## Table of Contents

01	<b>Overview</b>
02	<b>Regulatory Perspective</b>
03	<b>Unwrapping the Myth</b>
04	<b>Bridging the gap between Agile and QSR perspectives</b>
05	<b>Summary</b>
06	<b>Annexure A</b>

## OVERVIEW

Over the past few years, the Healthcare industry has undergone a transformation and it continues to evolve rapidly. So has the Medical Devices industry. The previously underserved segments such as assisted-living, geriatric care, chronic disease management, and wearables have made impressive strides in a global industry, which is worth more than \$300 billion. This boom is partly fueled by the advent of consumerism in the Healthcare industry and the widespread adoption of mature smartphone systems by consumers. As a result, these systems are increasingly being used as integration platforms for medical devices. The preference for these systems has resulted in the Food and Drug Administration (FDA) issuing a formal advisory for m-health apps.

This rapidly changing landscape and consumerism means that the medical device manufacturers or solution providers will have to be highly responsive and agile to consumer demands. The Medical Devices

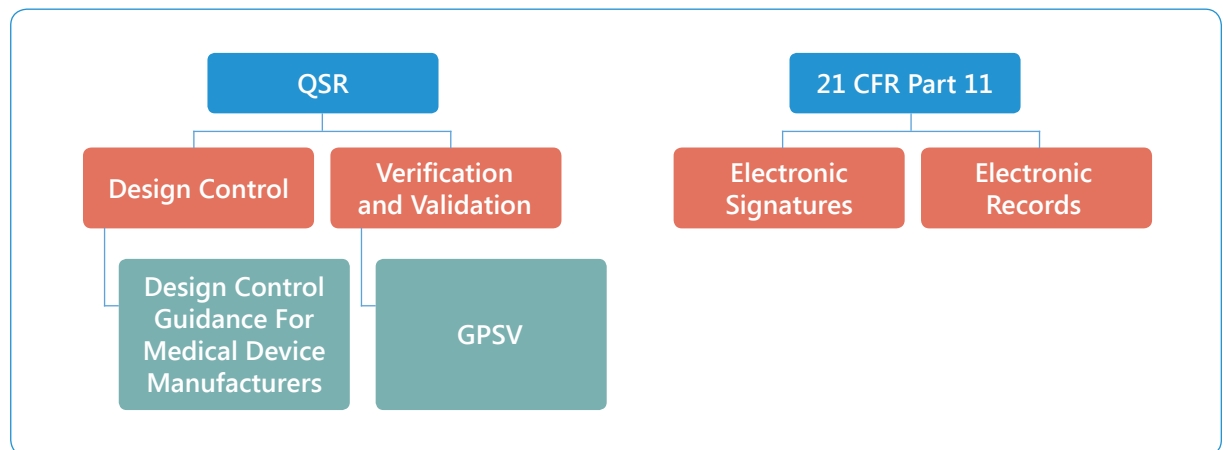
industry is highly regulated with emphasis on quality and safety of medical devices. Historically, it is accustomed to working in sequential Software Development Life Cycle (SDLC) methodologies such as Waterfall.

It won't be far off to state that, the general perception in the industry is modern software development methodologies such as Agile dispense with documentation and tracking, and hence are counterproductive or even disapproved from a regulatory perspective.

In this whitepaper, we aim to dispel the myth that the Agile development methodology is not suited for medical devices or that it is disapproved. We will also suggest how the existing development processes can be mapped to the prevailing regulations for compliance. For the sake of brevity, we have considered the regulations mandated by FDA.

## REGULATORY PERSPECTIVE

Here is a high-level overview of the regulatory landscape mandated by the US FDA for medical devices. In the USA, FDA's Center for Devices and Radiological Health (CDRH) regulates medical devices.



The key predicate regulation governing medical device software development under the US FDA Code of Federal Regulations (CFR) is 21 CFR Part 820, the Quality System Regulation. The subpart C "Design Controls" of CFR 21 Part 820.30 is specifically meant for medical device software developers and covers design and development planning, design input, design output, design review, design verification, design transfer, design changes, and design history file.

Apart from the predicate regulation, the QSR, FDA also requires medical device manufacturers to adhere to regulation 21 CFR Part 11, electronic records, electronic signatures; and commercial off-the-shelf software.

Software that runs medical devices or is used together with medical devices is automatically classified in the same safety classification as other medical devices. The software has to follow the same laws as the rest of the product. The medical device regulations thus affect the software development team's way of working. To safeguard public safety and delivery of safe, effective, and quality medical devices, the FDA requires medical device manufacturers to produce concrete documentation to prove that processes were followed, features validated, issues addressed, and risks mitigated.

## UNWRAPPING THE MYTH

With the basic regulatory overview done, let's take a closer look at the prevailing industry myths about the Agile methodology, the FDA regulations, and the applicability of Agile methodology to medical device software development.

**Myth 1:** Agile methodology is an antithesis to FDA-regulated software development because Agile eliminates documentation and the Agile Manifesto states:

**“Individuals and Interactions over processes and tools”**

**“Working software over comprehensive documentation”**

**Reality:** It might appear that the aforesaid key principles of the Agile Manifesto are contrary to the regulatory requirements. However, Agile methods neither mention working software “instead” of wide-ranging documentation, nor does it state that individuals and interaction are used instead of processes and tools. The operating word here is “over”. Further, Agile processes do not eliminate the need of documentation. This is clarified by the Agile Manifesto:

**“While there is value in the items on the right, We value items on the left more”**

In fact, in the current context, we can say that documentation has a tangible business value.

**Myth 2:** Medical device manufacturers have to register the medical device requirements with regulators prior to development, making redundant Agile's main principle of adopting changing requirements.

**Reality:** Regulators do require medical device manufacturers to register the medical device requirements. However, they ask for high-level requirements. Moreover, the FDA has stated that:

**“The plans shall be reviewed, updated, and approved as design and development evolves.”**

So, as long as robust validation and verification measures are in place, regular updates to the original design are not a problem.

**Myth 3:** Regulatory requirements recommend Waterfall development methodology and actively discourage the adoption of Agile methods.

**Reality:** A detailed analysis of the regulatory requirements reveals that no specific SDLC is recommended. Also, the rules do not restrict the adoption of the Agile methodology. In fact, the FDA's General Principles of Software Validation (GPSV) state:

**“This guidance does not recommend any specific life cycle model or any specific technique or method.”**

Also, the FDA's General Controls state that:

**“Although the waterfall model is a useful tool for introducing design controls, its usefulness in practice is limited [...] for more complex devices, a concurrent engineering model is more representative of the design processes in use in the industry.”**

**Myth 4:** Verification and validation activity should commence once development is complete.

**Reality:** FDA states that, “software validation and verification activities be conducted throughout the entire software life cycle”. In Agile, verification and validation activity starts at the user story level, providing traceability. Validation is an integral part of Agile development.

**Myth 5:** Risk Management activity can't be implemented with Agile teams.

**Reality:** In Agile development methodology, risk could be addressed right from the story level to the validation and verification activity. Risk analysis, evaluation, and risk control can be performed at the story level, as well as the increment, feature, and system levels.

**Myth 6:** FDA endorses only planned sequential development methodology.

**Reality:** Neither QSR nor other medical device regulations prescribe any particular development methodology. FDA actually cautions against using Waterfall for complex devices:

**“The Waterfall model's usefulness in practice is limited, for more complex devices. A concurrent engineering model is more representative.”**

In fact, FDA has endorsed Agile as a consensus standard. The consensus standard, AAMI TIR45: Guidance on the use of AGILE practices in the development of medical device software, covers several key topics such as documentation, evolutionary design and architecture, traceability, verification and validation, managing changes, and “done” criteria.

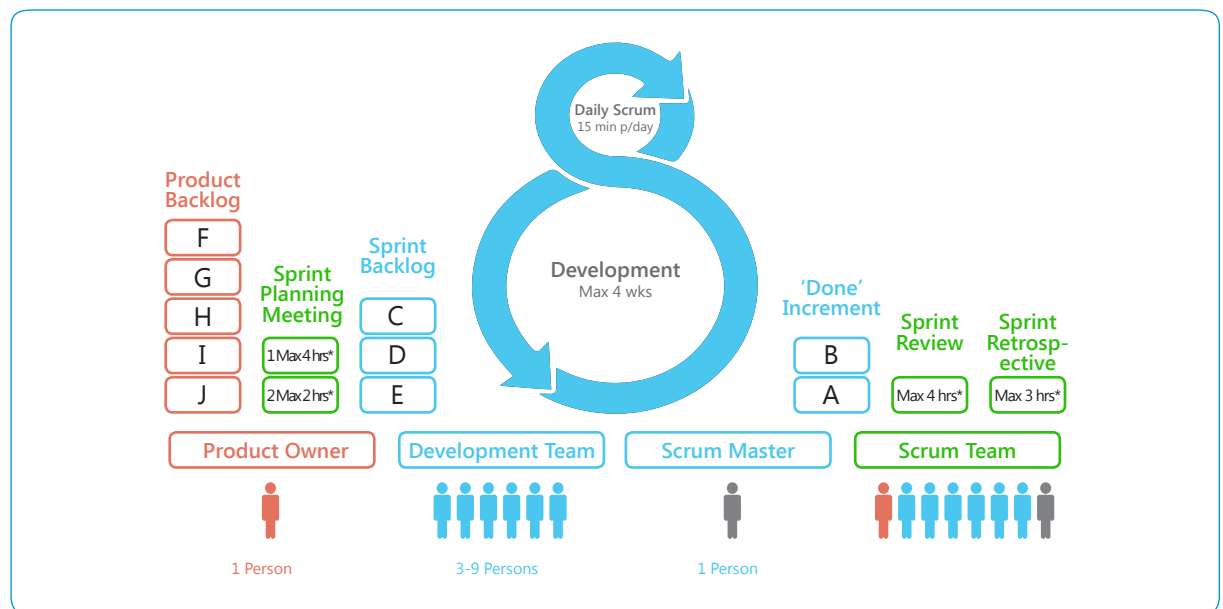
In short, regulations do not advise the use of a specific software development methodology and developers are free to choose an option that suits them best.

### BRIDGING THE GAP BETWEEN AGILE AND QSR PERSPECTIVES

Now that we have busted the popular myths related to implementation of Agile-based development practices in a regulated environment. Let's take a closer look at how we can use Scrum, one of the most popular Agile methodologies, to address the regulatory needs of documentation, evolving design and architecture, traceability,

verification and validation, managing changes, and finally “done” criteria in Agile terms. We will map Agile practices to the 21 CFR 820, QSR criteria.

The following illustration depicts a typical Scrum life cycle adopted by a medical devices software development team:



In Scrum, user stories under product or sprint backlog are considered as requirements. These user stories act as the input (§ 820.30 (c) Design input) to functional and technical specifications (story points). Here, every user story will have a definition of “done”, which assures that the story is properly coded, reviewed, tested, and accepted into the product. The definition of “done” includes all the quality, safety, security, and regulatory requirements that need to be satisfied for completeness.

The QSR talks about documentation of requirements. This requirement is addressed by Scrum through the product backlog (requirements) and the design document. Both, the design and product backlog are live documents and evolve as the product evolves. The product backlog document is focused on the end user and prioritizes the requirements on the basis of ‘Must’ and ‘Good for the Market’. Thus, the QSR expectation for documentation requirements and revisions is addressed.

Section § 820.30 (i) Design changes requirement of QSR, refers to accommodating requirement changes during a development cycle. In Agile, the product owner makes a decision on accommodation of changes in the course of a sprint. The product owner, based on the priority, the requirement, and the user requirement, decides which user stories (requirements) from the product backlog should be

reshuffled from the sprint backlog; and if required, updates the product backlog. Thus, Agile suffices the FDA-QSR § 820.30 (i) Design changes requirements.

Coding and verification is done for every story point (functional and technical specifications). An array of test cases are designed and executed to validate and verify the implemented code, thus satisfying and addressing the regulatory requirement of § 820.30 (f) Design verification requirements. The results generated through testing are captured and mentioned in the test report. At the end of every sprint, a demo to the customer is given to validate the working application. Thus, the regulatory requirement of design validation (§ 820.30 (g)) is also addressed. Based on the demo and the inputs received, the action items are incorporated in the upcoming sprint as user stories.

The FDA expects independence when reviewing, verifying, and validating the product, which can be of significant concern in the Scrum methodology. In Agile, in an ideal scenario, a developer can also test. However, the FDA requires that testing must be performed by groups outside of development—those that are not directly responsible for the item being reviewed. Therefore, to address this, ensure that all the requirements with medium risks and high risks are independently reviewed.

---

## SUMMARY

Medical device software development organizations are bound to adhere to regulatory requirements. Regulator places a large emphasis on requirements, safety and efficacy, and risk management when developing medical device software. Agile brings value to medical device software, if applied within the context of a Quality Management System.

Using Agile medical device software, developers can accommodate changes caused because of rapidly changing business scenarios while still complying with mandatory regulatory requirements.

Annexure A			
QSR Sections	Agile Terminology	CMMI Process Areas	Artifacts
§ 820.30 (b) Design and Development Planning	Product backlog	Requirements Management (REQM), requirements Development (RD)	Product backlog with prioritization
§ 820.30 (b) Design and Development Planning.	Sprint planning	Project planning, risk management	Sprint backlog, risk identification
§ 820.30 (c) Design input.	Sprint backlog	Requirement development	User story
§ 820.30 (c) Design input.	User story	Technical solution	Story points
§ 820.30 (c) Design input.	Story points	Technical solution	Functional and technical specifications
§ 820.30 (c) Design input.	Coding	Technical solution	Code
§ 820.30 (f) Design verification	Testing, review	Verification	Test report, review report
§ 820.30 (g) Design validation	Client demo, testing	Validation, risk management	Action items, test report, risk identification
§ 820.30 (i) Design changes	Action items, test report, review report inputs from stakeholders	Verification, validation	Sprint deliverables
§ 820.30 (d) Design output	Sprint deliverables	Requirement management and Project Monitoring and Control (PMC)	Release note, Read Me
§ 820.30 (e) Design review	Retrospection	PMC	MOM

Table: QSR requirements mapped to Agile methodology and CMMI process



### Cybage Software Pvt. Ltd.

HQ: Cybage Towers, Survey No 13A/ 1+2+3/1, Vadgaon Sheri, Pune 411014 |

Tel: 91 20 6604 4700 | Fax: 91 20 6604 1701

Pune | Hyderabad | Gandhinagar | Redmond | New Jersey | California | Atlanta | Texas | London | Frankfurt | Amsterdam | Sydney | Singapore | Japan

[www.cybage.com](http://www.cybage.com)