ptc

WHITE PAPER

Tailoring Agile Practices to Medical Device Development

Agile in Regulated MedTech Environments

Innovative ways of working can often seem at odds with achieving regulatory compliance. Despite the promise of an edge in an increasingly competitive market, some medical device developers are wary of trying Agile. Many still feel that the popular methodology often used to increase production speed and efficiency is wildly incompatible with highly regulated industries like medical technology. However, there are plenty of practices that you can use to adapt Agile to the needs of **medical device development**. Used in the right way, Agile can even accelerate your compliance efforts.

Why do some medical device developers shy away from Agile practices? Because medical device development must pay attention to regulatory compliance. This is in contrast to the areas where agile methods have emerged. They focus more on functional aspects, while regulatory compliance often involves nonfunctional requirements and process requirements like risk management procedures. When you combine this with the constantly changing requirements and decentralized development that characterize Agile development, it's no wonder MedTech developers feel that Agile practices don't fulfill the demands of their line of work.

That being said, software in and as a medical device is becoming more and more important, presenting limitless new opportunities to developers while also highlighting key challenges in the MedTech sphere. Between stringent regulatory compliance requirements, the struggle of combined hardware/software development, and the increasingly important role of software, is why MedTech companies all around the world are reimagining medical device development. Despite common industry concerns outlined above, Agile is in fact a reliable and proven way to help with that.

This white paper presents agile practices that are particularly well-suited for regulated MedTech environments.

Implementing Agile in Medical Device Development

Becoming Agile is an ongoing journey, not a destination. Unlike other working methodologies, the Agile Manifesto is not a prescriptive framework that lays out the perfect way to implement and use it. In other words, there isn't one single way to 'become Agile'.

This means that you are free to apply Agile principles and processes in a way that best suits your MedTech organization's needs. The tricky part is to decide where to start and what to do first.

Here are the basic steps you will need to take into consideration as medical device developers when embarking on your transition to Agile:

Learn and follow the basic principles of Agile

Everything in Agile is rooted in providing as much value to consumers as possible, in the most efficient way possible. Here are the **basic principles of Agile** that you need to familiarize yourself with when beginning your Agile journey:

1. Put the customer first, always
2. Welcome change (even late in the game)
3. Deliver software frequently and consistently
4. Use cross-functional collaboration
5. Build projects around motivated individuals
6. Use face-to-face conversations where possible
7. Measure progress with working software
8. Promote a sustainable pace of development
9. Emphasize good design and technical excellence
10. Strive for simplicity
11. Let teams self-organize
12. Reflect and tweak regularly

Find the right Agile method for you

As we've already mentioned, Agile refers to a variety of project management and software development methods based on the principles laid out in the Agile Manifesto.

Which one is the best fit for your organization will depend on a few factors such as:

- Organization size
- Team types
- Company culture

Here are some of the most common Agile frameworks to start with:

- **Scrum:** Breaks down projects into smaller parts which are carried out in short iterations.
- **Kanban:** Organizes work in a highly visual way on a board where project statuses are tracked as you go along.
- **XP (Extreme Programming):** Like Scrum, XP emphasizes frequent releases and continuous value for customers.
- **FDD (Feature-Driven Development):** Focuses on the development team and requires more rigid organization and documentation.
- **Crystal:** A lightweight Agile framework designed for short-term projects carried out by a team of developers using a single workspace.
- **LeSS (Large-Scale Scrum):** Allows you to scale Scrum to multiple teams who are working together on the same product.
- **SAFe® (Scaled Agile Framework®):** Implementing Agile for enterprises, focuses on streamlining collaboration and delivery from multiple Agile teams.

Mix and match Agile practices as you go along

Since Agile is an umbrella term that refers to several agile methodologies, there is no single way to go about implementing them in your organization. Feel free to explore the variety of frameworks and come up with a combination of processes and best practices that best suit the needs of your project, team, or organization.

Decide how you want to implement Agile

There are a ton of Agile methods and they all come with recommended implementation approaches. Some may suggest a top-down or bottom-up rollout, and others recommend starting small and spreading incrementally as you go along.

Essential Agile best practices for MedTech development

Before your organization starts on its journey to Agile, it's important to get the fundamentals right. You'll need to carefully plan processes to align Agile with MedTech regulatory requirements. You might want to consider taking a hybrid approach to ease into Agile and control any risks related to this change. Finally, you'll also need to build Agile GMS to ensure product quality in your new, high-velocity environment.

Once you have the basics figured out, the following best practices help make sure your Agile transition is successful:

Definition of Done

The Definition of Done, or DoD as it's commonly referred to, refers to a set of criteria that need to be fulfilled for each work item in an Agile project. It is particularly characteristic of an Agile methodology known as Scrum. Think of it as a checklist with a set of boxes that need ticking before you can label a task as done. In order for a piece of work to be included in an iteration's product release, it needs to meet those listed conditions.

Having a well-established DoD is also excellent for representing and ensuring the achievement of regulatory compliance requirements in Agile projects.

The Definition of Done typically covers three basic criteria:

- Individual nonfunctional requirements statements
- Groups of nonfunctional requirements
- Institutionalized processes and automated testing

How to implement it? The Definition of Done needs to be an agreement between the product owner and developers, with buy-in from the whole team. In a situation with multiple teams, they must collaborate to establish the product's Definition of Done to reflect organization-wide quality standards and make sure everyone's on the same page. The DoD can be implemented and visualized using written lists, central online documentation, backlog checklists, or by using separate tasks in an interaction backlog.

The Definition of Done can relate to each individual work item of an iteration (in Scrum called Backlog Item, typically defined in the form of a User Story) or the iteration (or Sprint) as such.

To give you an idea of what to include, here are some of the criteria that are commonly used to establish the Definition of Done:

Backlog Item Criteria for Potentially Shippable and Definition of Done(*):

- Code and tests checked in *
- Unit tests complete and pass *
- Code documentation complete
- Integration succeeds *
- Integration tests complete and pass *
- Static analyses complete and pass
- Packaging and staging succeed *
- System tests complete and pass *
- Acceptance tests complete and pass

Iteration Criteria for Potentially Shippable and Definition of Done(*):

- Performance tests complete and pass
- Risk management complete and documented
- All defects closed or prioritized for later iterations
- Installation packages available *
- Operations documentation complete *
- User documentation complete
- Marketing material complete

Criteria marked by an asterisk (*) are what the teams can currently accomplish within each sprint. They form the current Definition of Done.

Traceability between work items

Traceability across the chain of work items is a crucial requirement of being able to demonstrate that you have achieved regulatory compliance in a product. Instead of predefined work instructions, requirements tracing clearly connects the path between work products and results to demonstrate compliance. This is done by displaying the relationships between software artifacts and the overall product development lifecycle. To give you an example, trace links can start with functional requirements to represent the demands of a legal regulation like EU regulation 2017/745, then connect to their implementations and test cases to document their status. This way it is easy to see how the product, or different parts of it, satisfy certain regulatory demands.

The most important thing to remember when tracing requirements using Agile is:

- Comprehensively planning how you will capture and implement requirements

- Using the right tool to track their development status and progress, something that covers the whole product lifecycle

- Leveraging an integrated infrastructure that combines development information and assets

Incremental compliance

Too many projects still address regulatory compliance late in the development lifecycle. It's often the case that most, if not all compliance activities are undertaken after finishing product development. Leaving it all to the end can cause untimely delays, unnecessary rework of what is supposed to be a finished product, and inevitably hikes up those pesky project costs.

Agile development encourages incremental compliance, giving you the opportunity to check in on compliance management efforts throughout the development lifecycle, instead of just at the end. This has a huge effect on your team's productivity. The more you integrate compliance, the faster and more efficiently the team can go about their development activities.

So how exactly can Agile practices help you achieve incremental, and eventually, continuous compliance?

- **Including compliance criteria in the Definition of Done** to make sure that that compliance measures are undertaken for every backlog item or iteration
- **Continuously reduce "undone" work** to strengthen the Definition of Done
- **Adding compliance-related functional and technical capabilities in the product backlog** to be implemented just like any other product element

Additional Agile Practices for MedTech

Compliance management can choose from a wide range of practices that help establish regulatory compliance in Agile product development. This section lists important and useful information sources.

Backlog Constraints

Backlog Constraints are a practice proposed by SAFe® for storing non-functional requirements, and for making them visible to the project team. Backlog Constraints is a data structure that collects the non-functional requirements. It is associated in a suitable way with the product backlog. Teams and organizations are free to decide about their exact implementation of the data structure.

Agile Release Train

Agile Release Trains (ART) are organizational structures that gather all teams and stakeholders needed to develop a set of related product solutions. As an agile practice that has originally been defined in SAFe® they can be used in MedTech to orchestrate the interaction between the traditional compliance manager role and agile cross-functional teams.

Test Automation

Test Automation is a wide field, with many powerful solutions relevant to compliance management. The basic strategy is that automated tests demonstrate fulfillment of compliance criteria after every relevant product change. Once the infrastructure for automated tests has been established, each individual test cycle has a low direct cost. The benefits can outweigh the costs significantly.

Behavior-Driven Development

Behavior-Driven Development (BDD) is a method that combines acceptance testing, test automation, and a specific approach to requirements definition. Its elements encompass the entire development lifecycle of agile iterations. BDD is sometimes referred to as Specification by Example. It defines user story details in the form of test scenarios. An underlying automation framework makes it possible that textual requirements and test definitions can be executed and run automatically. The method requires some upfront investment and continuous maintenance of its technical infrastructure. However, it can make test-driven development highly effective.

How to implement Agile best practices in MedTech development?

Now that we've explored some of the core Agile practices which lend themselves well to both MedTech development and regulatory compliance, you may be wondering how to go about implementing them in your organization. Collaborating with other Agile teams, targeting low-hanging fruit, and continuously improving as you go along by conducting retros for example are reliable ways to get started. You can also use hybrid development methods in order to leverage Agile benefits without fully divorcing the way your organization already works. There are a whole host of other Agile practices worth exploring to see how you can mix and match them to your advantage.

All in all, Agile provides significant advantages in comparison to traditional and rigid plan-based development methodologies, while hybrid Agile (e.g. the combination of Waterfall and Agile) allows you to keep the best of both worlds. Established Agile practices not only comply with healthcare regulations but can actually help you optimize the process of complying with them for increased speed, efficiency, productivity, and stakeholder satisfaction across the board.

Get Started Now

Driving innovation in life sciences with Agile is a real possibility, all you need is to get started. PTC's Codebeamer can help you transition to Agile processes and bring quality MedTech products to the market quickly and at optimal cost.

Explore how Codebeamer's templates for frameworks like SAFe®, Scrum, or Agile-Waterfall Hybrid help you tailor and adopt agile practices with minimal effort. Leverage MedTech templates with baked-in domain knowledge to simplify adherence to the requirements of ISO 13485, EU MDR, FDA Title 21 CFR, and more.

When you are ready for a closer look, sign up for a free 30-day trial.

DIGITAL TRANSFORMS PHYSICAL

121 Seaport Blvd, Boston, MA 02210 : ptc.com