

Applying Agile Software Principles to the Medical Device Life Cycle

Varun Pandit, Senior Software Engineer and Lao Thao, Software Engineer

In the development of software for medical devices, international standard IEC 62304 defines process life cycle requirements. Linear software development life cycle (SDLC) models, such as the waterfall model, fit in well with IEC 62304. The waterfall model is a sequential design process, commonly used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

Minnetrnix employs the V-model, a very popular extension of the waterfall model. The V-model is simple to understand and makes it easy to plan all stages of a project (requirements, design, implementation, testing, etc.). The V-model (Fig. A) offers some improvements to the waterfall model, but drawbacks exist because it cannot easily adapt to obstacles in day-to-day progress.

Some characteristics of the model that limit its efficiencies are as follows:

- Lack of tracked progress
- Late detection of software bugs
- Late detection of conflicting requirements
- Incomplete requirements are not uncovered until the test phase
- Lack of feature prioritization
- Lack of early hardware and software integration

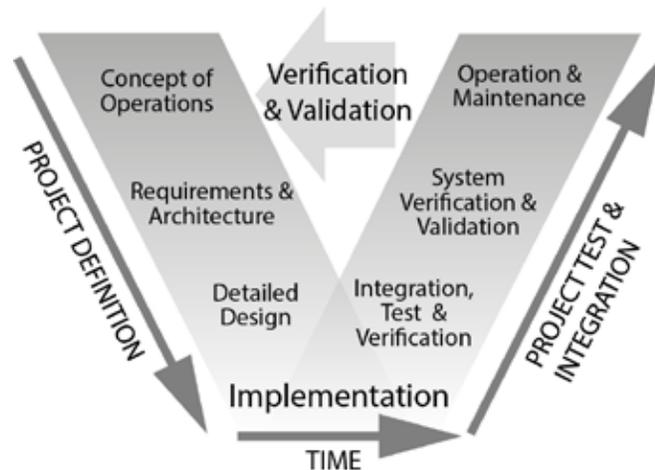


Fig A: V-Model Software Development¹

Agile is a software development process that promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and rapid and flexible response to change. The Agile Manifesto is based on 12 key principles (Fig. B). Adapting certain Agile software development principles to fit within the V-model can offer valuable enhancements to the medical device software development process.

The Agile Principles

1. Customer satisfaction by early and continuous delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity – the art of maximizing the amount of work not done – is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstance

Fig B: The Twelve Principles of the Agile Manifesto²

At Minnetronix, we are currently integrating Agile principles in the following phases of the V-model:

Requirements definition

- System and software prototyping
- Start creation of adaptive high-level design

High-level design, detailed design

- Refine requirements based on early stage delivery of prototypes and customer feedback
- Simplify requirements in use cases

Implementation, unit testing

- Continuous integration of hardware with software releases through Sprints
- Availability of functional software throughout the project

Integration testing

- Early detection of bugs and application of fixes through verification dry-runs

One of the Agile processes that Minnetronix applies to software development is the use of Sprints. Sprints are repeatable work cycles in which a set of defined tasks is planned to be completed within a defined period of time. Sprints have their origins in Scrum, an iterative Agile software development methodology. Sprints are two to three calendar weeks long; this short-term view provides a more tangible tracking of progress.

Every Sprint should result in an intermediate working product that incrementally builds towards the final product. This intermediate working product is released to QA to perform some preliminary testing. This creates a feedback loop where bugs are discovered early. It also helps the QA team develop test procedures that are used in software system testing, as required by regulatory bodies. By contrast, in a traditional V-model, many problems would not have been uncovered until the final testing phase.

The philosophy of a Sprint is that the team works only on tasks that were defined during Sprint planning. In most cases, the team should defer any new tasks that arise in the middle of a Sprint. All the team members need to respect this philosophy; organization and execution of a specific plan is the foundation of a Sprint. Daily stand-up meetings promote team dynamics by ensuring frequent communication; problems are exposed daily. In this manner, the Minnetronix team can adapt to situations that arise and fix problems early, regardless of whether they are requirement conflicts or design issues. Sprints help our team keep objectives simple and focused. We find that Sprints draw attention to the need for prioritizing work that is aligned with higher-level project goals and milestones.

As we strive to further incorporate Agile principles into software development, we are leveraging TIR 45, a guidance issued by the Association for the Advancement of Medical Instrumentation (AAMI). TIR 45 provides direction on how to incorporate Agile into the medical device life cycle. The guidance is very expansive, and we have identified certain of its recommendations and addressed them as follows:

Agile is to be used within an established QMS in a way that causes minimal disruption to the QMS.

- Our QMS defines procedures for activities that are the foundation of any medical device life cycle. Our QMS provides us the flexibility to incorporate Agile and still adhere to IEC 62304.

Define the life cycle model to set the proper expectations among the software development team and regulatory stakeholders.

- Through the Sprint planning mechanism, we identify points when intermediate software releases can be created. This also helps identify when completed software will be available.

The software configuration system must be robust in handling frequent change.

- We leverage a version control system that developers use day-to-day. The same system is used to capture software releases.

Inputs and outputs must be "DONE" to call the work DONE.

- Properly defined requirements and integration are the key to calling work "done". Multiple related tasks might need to be defined in order to implement a certain feature. The feature is not done until all those tasks are complete, integrated, and tested with the rest of the system.

Establish effective RETROSPECTIVE and reflection practices to support continuous improvement.

- We hold Sprint retrospectives to reflect upon the work completed in the Sprint. We address topics such as what worked, what didn't work, was communication effective, were estimates correct, etc.
- We strive to provide opportunities for all team members to provide critical and constructive feedback.

Use AGILE's continuous integration practices as the core of an effective integration strategy.

- A stable and working build is the intent of every Sprint. Any issues or bugs that threaten stability and functionality are dealt with as the highest priority. Team members continuously coordinate and integrate to expose such problems.

Continuously address customer needs through an active customer role.

- The intermediate working products that are released to QA can also be supplied to the customer as working prototypes or demos. They serve as a great tool to get feedback from the customer, which helps refine requirements.

Incorporating Agile principles into specific aspects of the V-model is part of Minnetronix' overall commitment to continuous improvement and best practices. On an ongoing basis, we strive to uncover and incorporate new ways to improve our software development processes. With a singular focus on medical applications, our software engineers integrate state-of-the-art technology into medical products that demonstrate outstanding performance, reliability, and regulatory compliance.

1. Clarus Concept of Operations. Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005.
2. Beck K et al. (2001). "Principles behind the Agile Manifesto". Agile Alliance. Archived from the original on 14 June 2010. Retrieved 6 June 2010.

Minnetronix Medical, Inc.
www.minnetronixmedical.com

Minnetronix Medical is a trademark of Minnetronix Medical, Inc.
©2018 Minnetronix Medical, Inc. DM-0000-189-101 A