







Introduction to Agile: Software Development at the Speed of the Web

By Sahana Anantha Murthy





Table of Contents

The Evolution of the Software Lifecycle	
What Is Agile?	•
Why Is It Important?	
Featured Agile Processes)
How to be Agile	





The Evolution of the Software Lifecycle

Remember these? Cassettes, VCRs, disposable cameras, pagers, dial-up Internet, CDs, paper maps, and encyclopedias? They became obsolete as they were replaced by more effective solutions.

The Waterfall model, a type of software development lifecycle, also fell to the back of the pack. In the Waterfall model, change happens slowly and incrementally within the development process. If phase 1 is not complete, no other part of the project can be started. As the marketplace demanded great digital experiences, requiring constant updating and innovation, the need for agility took center stage. This led to the Agile software development lifecycle model. "The Agile method is more analogous to how things actually work in today's software world, where nothing is black or white."





Chapter 1 What Is Agile?

The Agile methodology is an evolution in the software development process that enables software teams to move from the conventional waterfall model to something much more dynamic. It eliminates the "waiting" that teams experience in the waterfall model to allow various moving pieces to interact while one phase is going on. Collaborative efforts, successful planning and execution, and flexibility to tackle projects and changes head on are all promoted through this approach.

The Agile method is more analogous to how things actually work in today's software world, where nothing is black or white. Agile developers often say that software development cannot be made on an assembly line, because everything cannot be synchronized and predicted.

"Never limit the innovative" is a resonating tag that runs in developers' minds when thinking of Agile. Development is done in small iterations, usually bi-weekly sprints where use cases are created, developed, and tested. At the end of each of these sprints, there is a shippable, deployable product available. Changes also can be incorporated dynamically and then delivered in the next sprint drop. This cannot happen in the Waterfall model until the initial design and changes are fully implemented and deployed. With the Agile method, developers are expected to be, well, agile. More often than not, requirements, design, development, and testing phases can all happen in the same sprint.



Acould

Chapter 2 Why Is It Important?

While it seems like a lot to organize, because Agile is incremental, the project is typically divided into smaller chunks that are added to the sprint backlogs and prioritized accordingly. This enables the team to focus on just the right amount of action items for each sprint. Development is done in small chunks iteratively; tested and verified frequently, allowing teams to be flexible and more responsive to customer requirements.

Inspired from "lean" manufacturing concepts, Agile takes into account the unpredictability and dynamic nature of software development and embraces it.

Being Agile in today's market and throughout the development process will make various tasks and functions run smoother and more efficient, delivering a better quality end product.

"Inspired from "lean" manufacturing concepts, Agile takes into account the unpredictability and dynamic nature of software development and embraces it."



Chapter 3 Featured Agile Processes

1. Scrum—Perhaps the most popular implementation of this methodology, Scrum is a type of software management technique in Agile methodology that offers control, direction, and structure to teams using Agile.

If Scrum is followed by the book, each Scrum team should have the three main roles: Product Owner, Scrum Master, and a cross-functional Development Team. The product owner assumes maximum responsibility with the success of the product, ROI, and the sprint deliverables. The Scrum Master however, is the main arbitrator for Scrum and ensures that continuous integration and best practices are adhered to.

One of the most popular aspects of Scrum is the stand-up, daily Scrum meeting. It's a quick, succinct stand-up meeting meant to address what each person in the team did yesterday, what they plan to do today, and what roadblocks they will face to achieve their goals. It's a great recap of what everyone in the team is trying to achieve.

Apart from the daily Scrum stand-up meetings, there are other periodic meetings for sprint planning, sprint demos, and addressing backlogs and caveats. When done right (which many companies don't), this can be extremely effective for lean software development.

2. Extreme Programming (XP)—XP is a programming methodology primarily meant to improve the quality of code that teams deliver. Along with smaller sprints, XP encourages frequent checkpoints in the form of pair programming and code review. There are frequent cross testing and validation of code quality and best practices. Not only does pair programming help deliver good quality code, but it also creates camaraderie in the team.





Customer service is another key facet to this methodology. Rather than implementing the designs and features the customer wants, it takes only what is necessary or what they need to make corefunctionality optimal. Extreme Programming keeps the task at hand simple and clean. Moreover, since the model is customer-centric, it allows changes to be made at any point in the development cycle, no matter how far into the process.

XP is also known for advocating frequent "releases" in short development cycles to adapt to customer requests and improve productivity within the project group. XP in theory is a great methodology for peer programming and other facets that make projects more effective both in collaboration and efficiency. However, XP isn't used as frequently today because of the skill and the amount of developer time required to execute it.

3. Kanban—Inspired by the just-in-time lean manufacturing process, this methodology applies similar principles to software development. The underlying concept of incremental development resonates with Kanban just as it does with Scrum, but it is quite different from typical Scrum management. Kanban does not impose a set of roles for team members. In fact, one of the principles of Kanban is to get started with what we have now and change how we function.

The basic idea is to not overwhelm the developers with a set of sprint action items like in Scrum. With Kanban, the team creates a framework around the workflow of the development process, allowing developers to pull out and work on a single action item at a time. This ensures that work-in-progress is limited to the single item pulled out by each of the members. As a result, there can be continuous delivery of code without having to wait until the sprint drop. Moreover, it allows developers to innovate and improve code dynamically on a daily basis. As a result, Kanban is slowly becoming the popular choice for Agile management among companies.





Chapter 4 How to be Agile

The key to success with the Agile model lies in quality execution, which is precisely why we find so many organizations falling short while attempting to use this model.

To be Agile, each team needs to outline well-defined roles that will be assigned and that communicate effectively to make sure that each group is a cog in the larger wheel of successful and efficient development. This way, all moving parts can exist together to make the process more "agile."

The Agile methodology is revolutionizing the way software teams function, inspiring them to adopt many similar processes such as continuous integration and source code management to improve their workflow. Using this model effectively requires excellent teamwork and communication, as well as the ability to adapt to changes on the go. The Agile model is not only bringing products to market faster, but making collaborative efforts more cost-effective and time-efficient.

Today, being a developer means being Agile.





