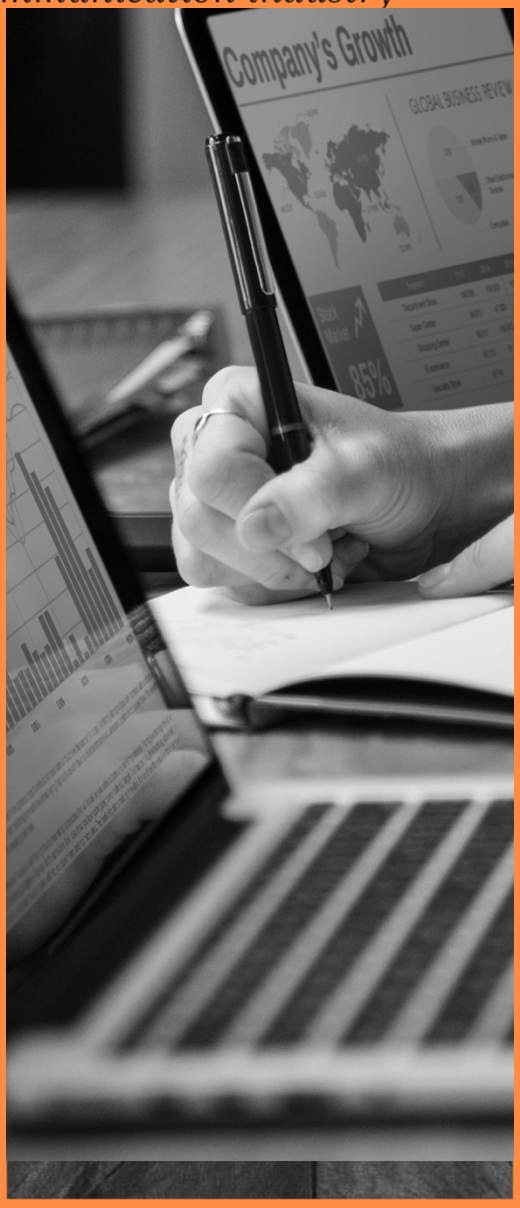
Mercer University

AGILE DEVELOPMENT FOR TECH COMM

A White Paper for the technical communication industry



Prepared by Sydnei Mayers

March 2, 2021

Table of Contents

Introduction

Many organizations are stuck in development cycles that are slow and unresponsive to customer feedback. Ideally, Agile development would be used in place of the complicated and unresponsive traditional Waterfall methodologies. Adopting the Agile framework lets teams adapt to changes in the marketplace or feedback from customers quickly. Planning and developing in small, frequent increments allow your team to gather feedback on each change and integrate it into future plans at a minimal cost. This document

- Provides a basic understanding of Agile development
- Discusses the importance of Agile development in the technical communication industry
- Introduces how teams can adopt the Agile approach

What is the Problem?

In the 1990s, software development faced a crisis. The industry realized that it could not move fast enough to meet customer demands and requirements. Traditional development models such as the Waterfall methodology were based on a timeline approach; development happened sequentially. The final product was not delivered to customers until the very last step. As a result, by the time an actual application was finished, it was highly likely that the requirements and systems of the project's original objectives had changed. With time, money, and efforts wasted, professional leaders of the software community thought it was time for a new, refreshed approach (Musilihat, 2018). Their solution was Agile software development. Agile prevented them from wasting their time, money, and other resources.

What is Agile?

In 2001, Agile was formally launched when 17 technologists drafted the Agile Manifesto. The Agile Manifesto is a declaration of four foundational values and 12 key principles expressed in Agile methodology. The manifesto aims to find better ways of developing software by providing a clear, measurable structure that promotes iterative development, team collaboration, and change recognition (Muslihat, 2018). The four foundational values are shown in Figure 1.

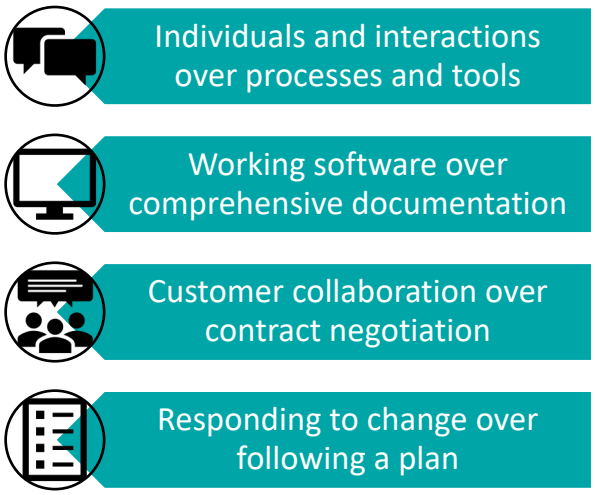


Figure 1. Agile Foundational Values

Agile software development refers to a collection of software development methodologies based on iterative development, where solutions and requirements change through collaboration between cross-functional teams. The Agile approach does not apply only to software development. Agile development refers to any development process aligned with the Agile Manifesto concepts (Cprime, n.d.).

Agile Development Cycle

In Agile development, there are five phases that require frequent meetings. The meetings save time by optimizing development tasks and reducing errors during the planning stages. Figure 2 shows the Agile development cycle and the traditional Waterfall cycle.

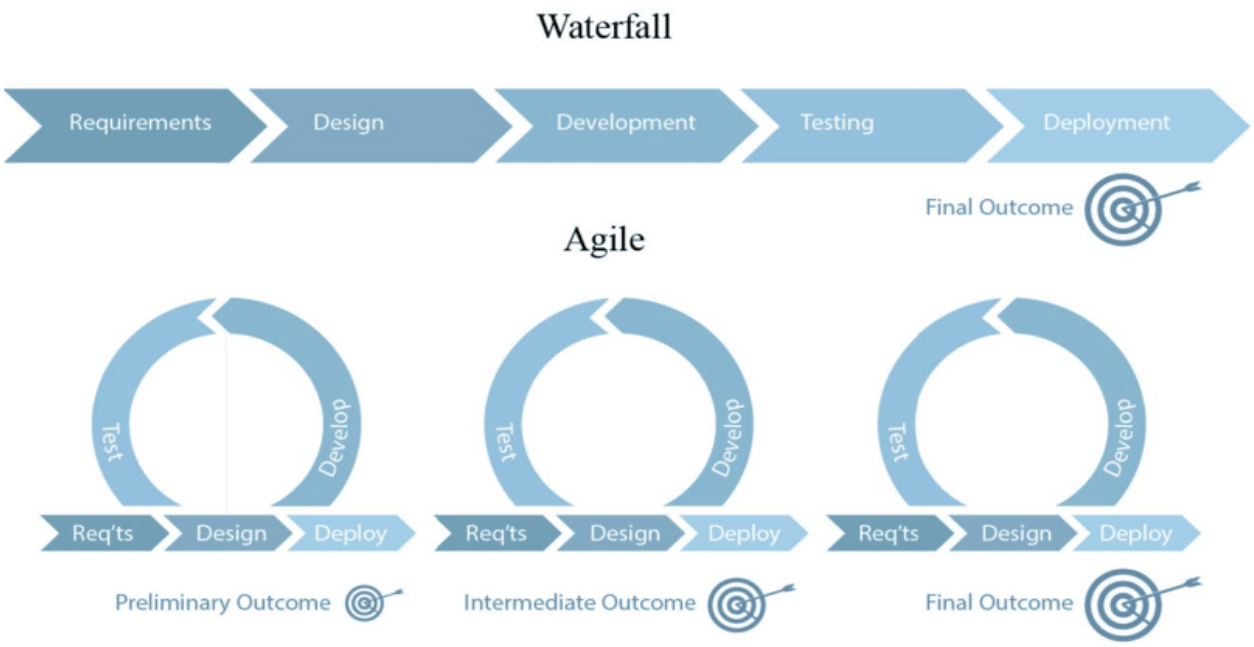


Figure 2. Agile Development Cycle as Compared to Waterfall (Eby, 2017)

Phase 1: Requirements

As product owners begin designing their project, they need to create a list of initial requirements that assists in creating the initial documentation. The initial requirements are:

- The end result that the project is going to achieve.
- The features that the project will support.
- The features that the project will not initially support (Feoktistov, n.d.).

Phase 2: Design

The system and software design are prepared from the requirements identified during the requirements phase. The team must think about what the product or solution will look like. The team also comes up with a test strategy or plan to proceed (Eby, 2017).

Phase 3: Development

This phase is focused on creating features and scheduling iterations for testing and deployment. As this is the first iteration in this phase, it includes preparing the environments, finalizing contracts, and funding (Eby, 2017).

Phase 4: Test

Once the product has been developed, it is tested against the requirements to ensure the product is actually solving customer needs. During this phase, integration testing, unit testing, system testing, and acceptance testing are done (Eby, 2017).

Phase 5: Deployment

After testing is completed, the product is delivered to the customers. However, deployment is not the end of the project. Customers may run into problems while using the product that the project team will need to address (Eby, 2017).

Agile Scrum Methodology

As Sacolick (2020) mentions, many Agile frameworks that provide information about development processes align to a software development life cycle. *Scrum* is the most popular Agile framework. It focuses on a delivery method called a *sprint* and meeting structures that include the following:

- **Planning** where the product owner identifies the sprint priorities

- **Commitment** where the team reviews a list of user stories and decides how much work they can accomplish during the sprint
- **Daily standup meetings** where the teams give updates on their development and testing status and strategies

Sprints end with a demo meeting where the product owner is shown the product, followed by a retrospective meeting where the team discusses what went well and what needs improvement. Many organizations hire scrum masters or coaches to help teams manage the scrum process (Sacolick, 2020).

How Could the Tech Comm Industry Benefit from Agile?

According to Smith and Gale (2014), teams will discover the strategies needed to be successful as Agile technical writers by working together in a supportive Agile environment. Agile teams succeed or fail together. As a result, every team member can help design, test, and document the product's features. For technical writers, this increased collaboration leads to greater access to developers and subject matter experts and a greater understanding of the development process. In turn, Agile saves an organization money with its iterative development.

In an Agile environment, more insight is gained into the work of the supporting teams. The backlog is the tool used to outline and prioritize the team's development work. By viewing the upcoming work, you can calculate the pace at which features are being developed and plan work accordingly. The "last-minute" features that require documentation occur less frequently than before.

One of Agile's major benefits to writers is the even distribution of work throughout the project. When collaborating with Agile teams, documentation of features occurs during development. The product and its documentation are tested along the way, which leads to fewer errors or last-minute changes before deployment. As a technical writer, knowing that the product is likely to evolve in future iterations means there will be an opportunity to improve the documentation quality before it is delivered to the users.

Agile teams are learning to integrate user experience (UX) design with prototyping and quick iteration to evolve designs within a sprint. Being involved with the team while features are being developed leads to a deeper understanding of those features and makes it easier to document. Having just designed and tested the product, technical reviewers have immediate knowledge of the features, so their feedback is more accurate and detailed (Smith & Gale, 2014).

What Should You Look Out For?

According to Eby (2017), while the level of flexibility in Agile is usually a positive, it also comes with trade-offs. It can be hard to establish a definite delivery date, documentation can be overlooked, or the final product can be very different than initially planned. Here are a few disadvantages of Agile:

- **Planning can be less concrete.** Because Agile is based on time-boxed delivery, it is possible that some items may not be delivered on time.
- **Documentation can be neglected**. As stated earlier, one of the four Agile foundational values is working software over comprehensive documentation, so team members may feel it is less important to focus on documentation.
- **Final product can be very different.** Since the initial Agile project might not have a definitive plan, the final product can look much different than initially planned.

Best Ways to Adopt the Agile Approach

Now that a basic overview of Agile development has been given, let's discuss how to adopt Agile. Shown to the right in Figure 3 are six Agile methodology steps that a team can follow to transform its approach to project management and software development (Reynolds, 2020). Remember that there is not a right or wrong way to start with Agile. If Agile is not suitable for your team, the traditional Waterfall method may be a better match.

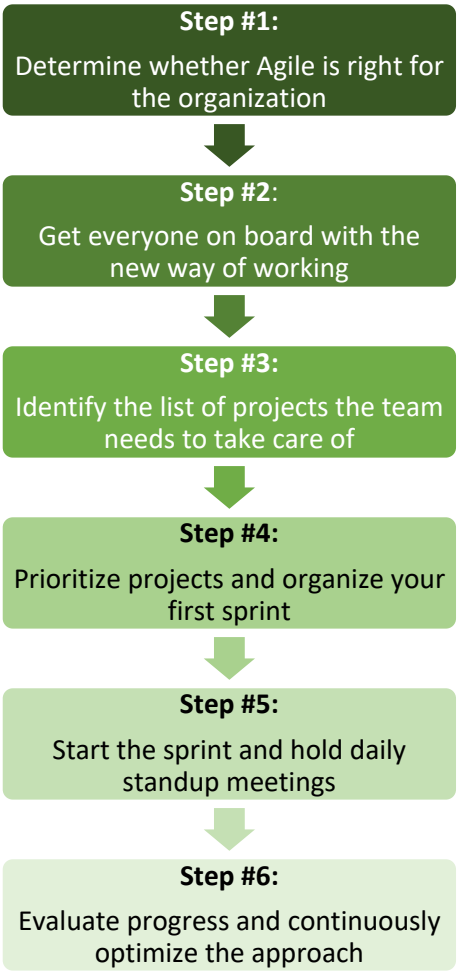


Figure 3. Steps to Adopting Agile Approach

Conclusion

This white paper details the advantages and challenges of Agile development, specifically for technical communicators. As professionals, we must always look for ways of improvement. Not only does Agile save time and money, but it also saves efforts that would have been wasted. Ultimately, Agile development is an excellent option for technical communicators to adopt into their workplace.

Special Acknowledgement

I would like to express a special thank you to my mentor, Dr. Jackie Damrau. She unselfishly shared with me her immense wealth of knowledge regarding Agile development.

References

Cprime. (n.d.). *What is Agile? What is Scrum?* https://www.cprime.com/resources/what-is-agile-what-is-scrum/.

Eby, K. (2017, February 15). *What's the difference? Agile vs Scrum vs Waterfall vs Kanban.* Smartsheet. https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban.

Feoktistov, I. (n.d.). *Agile software development lifecycle phases explained.* Relevant Software. https://relevant.software/blog/agile-software-development-lifecycle-phases-explained/.

Muslihat, D. (2018, March 2). *Agile methodology: an overview.* Zenkit. https://zenkit.com/en/blog/agile-methodology-an-overview/.

Reynolds, J. (2020, November 12). *Adopting Agile methodology: 6 steps to improved delivery.* https://www.plutora.com/blog/adopting-agile-methodology-steps.

Sacolick, I. (2020, February 25). *What is agile methodology? Modern software development explained.* https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html.

Smith, K., & Gale, P. (2014). Agile for technical communication. *Intercom*, *61*(10), 7–9.