



Which methodology is better, Waterfall or Agile?

by Harry Saluja and Rob Carstons
Sep 16, 2021

Introduction

Which methodology is better, Waterfall or Agile? This question often comes up as companies progress on their Agile journey. To answer this question and to determine the best approach to solve any problem, one must first understand the difference between *Complex* and *Complicated* contexts².

Complex Context

COVID-19 is the perfect case-in-point. Even though some might have predicted a global pandemic, there were no good practices in place or checklists that would help guide us in how someone would solve this problem. Industries spent many months learning how to adapt not just how people live their lives, but also how one can respond as a business. There were lots of experiments going on throughout the world to develop treatments and vaccines, quarantine, reopening and shutting down the economy, and working remotely. This event was unpredictable and complex, and everyone had to *learn* in order to figure out what to do next. Most innovation projects fall into a *Complex Context*. There are also some aspects of small-capital projects that may fall into a *Complex Context*.

Complicated Context

Airplanes and Refineries are highly *Complicated* systems with feats of engineering and science. While the typical layperson may not understand what keeps an airplane in flight or a refinery humming, there are highly trained engineers that understand how they work and how to diagnose and solve problems as they arise. The systems are *Complicated*, yet the engineers know everything about them and how to maintain those systems. Solving problems in a *Complicated Context* is difficult, but usually there are good practices in place, and there are constraints that help govern the way in which problems are solved.

How work gets executed

Traditional Project Methodologies

Project methodologies, such as Waterfall³, utilize a linear, phased gate approach for doing project work. These methodologies include all necessary artifacts, checklists, and funding gates with the goal of mitigating risk, eliminating uncertainty, and ensuring a predictable project delivery. The expectation is that you've planned upfront for all eventual risks and outcomes in the early phases of the project. These types of methodologies work well for projects that have high levels of certainty and low levels of *Complexity* but may be highly *Complicated*. Replacing a large piece of familiar equipment in an oil refinery with no compounding circumstances is highly *Complicated* and difficult, but there are engineers who understand the technical details of how this equipment works and how to replace it. It's rare that there's much discovery happening, and the engineers can likely predict what challenges they may encounter and have ways of dealing with those challenges. Traditional project methodologies work well for these types of projects.

¹ manopjk © 123RF.com

² <https://hbr.org/2007/11/a-leaders-framework-for-decision-making>

³ <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>

Agile Frameworks

Agile has two facets:

1. *Mindset*: The values and principles that guide how individuals, teams, and organizations think about and act upon the work they do. Mindset includes, but is not limited to, things like Individuals and Interactions, Responding to Change, Simplicity, and Customer Collaboration⁴.
2. *Practices*: These include Scrum⁵, User Story writing⁶, and Relative Estimation⁷, just to name a few. Each of these practices help to reinforce the mindset.

Agile frameworks, such as Scrum, are iterative and incremental (build iteratively and deliver incrementally) frameworks that incorporate empirical learning⁸. Teams generally work in short timeboxes called Sprints that are usually 1 to 2 weeks in length. Sprints function as short feedback loops that help to mitigate risk, reduce uncertainty, and ensure the team is delivering the highest value to its customers. Each Sprint allows the team to focus on delivering a small increment of value, get feedback on that increment of value, then adjust their plan based on that feedback. The Scrum framework typically works best in *Complex Contexts* where learning and discovery is essential to providing the best solution possible.

Agile Pitfalls

- **Applying Agile Frameworks to Everything**: One of the biggest pitfalls organizations run into is trying to apply a single framework, such as Scrum, to everything they're doing without first understanding the context in which it's being applied. It's kind of like the old saying from Abraham Maslow, "*If the only tool you have is a hammer, you will start treating all your problems like a nail*".⁹ In this case, Agile or Scrum becomes the hammer, applied anywhere and everywhere, even in domains where it has no business being applied. While agile frameworks, such as Scrum, should not be blindly applied everywhere, agile principles, such as visualizing work and customer collaboration add value in any environment. Understanding the domain in which you're working helps to ensure that you're applying the right tool for the job.
- **Applying Agile Practices by the book**: Companies in asset heavy industries are process focused (for the right reasons: safety, reliability etc.); and sometimes struggle to adapt processes and practices making them fit for purpose. In the Scrum framework, the intent of the Daily Scrum is to reinforce collaboration, communication, plan the day's work, and raise impediments that impact achieving the Sprint Goal, so they can be resolved quickly. The Daily Scrum is a solution to a problem where teams may not be communicating and collaborating, and they struggle to continually re-plan their work and raise impediments. If your team has these challenges, then the Daily Scrum may be helpful in reinforcing these patterns. If your team collaborates all day long, and your entire team is fully engaged, then the Daily Scrum may not add as much value.
- **Agile is ONLY for Projects**: Agile frameworks and methods are tools that reinforce teaming skills and help teams manage work. Traditionally, teams have been built around projects, completed those projects, then disbanded the teams at project close. When the next project comes along, the same process is repeated, with many team members fractionalized across multiple project teams, and teams going through Tuckman's Stages of Team Development¹⁰ (forming, storming, norming and performing) time and again. The concept of a persistent team or stable team that is made up of complementary cross-functional skills that provide a specific service to the organization is a concept that allows teams to manage all work (project and operational), allowing teams to prioritize a single backlog of work and more effectively manage capacity with demand. All work flows to the team vs. building teams around the work (i.e. projects).

Context is Everything

So, back to the original question of which methodology is better, Waterfall or Agile? Each framework and method has its purpose, and one isn't better than the other. The decision to use one method or framework over another is *contextual*. You wouldn't use Scrum to build a bridge. It's a highly *Complicated* project, but it's been done before, and there are engineers that specialize in this type of work. You also wouldn't expect to use a Waterfall methodology to develop a corporate strategy or develop a new innovative solution to a *Complex* problem. Each framework and method are simply tools that have value when applied in the right context. Teams should first understand the work they are trying to accomplish, whether it's predictable, whether there's some discovery involved, and whether teams expect to learn as they go. There are frameworks that can help with understanding what domain or context your work may fall¹¹. Understanding the context of your work will help guide what approach might work best.

Disclaimer: reviews and opinions presented in this article are those of author and not his employer.

⁴ <http://agilemanifesto.org/>

⁵ <https://scrumguides.org/scrum-guide.html>

⁶ <https://www.agilealliance.org/glossary/user-story-template/>

⁷ <https://www.agilealliance.org/glossary/relative-estimation/>

⁸ <https://scrumguides.org/scrum-guide.html>

⁹ Abraham H. Maslow (1966). *The Psychology of Science*. p. 15. ISBN 9780976040231

¹⁰ <https://hr.mit.edu/learning-topics/teams/articles/stages-development>

¹¹ <https://hbr.org/2007/11/a-leaders-framework-for-decision-making>