A Comprehensive Guide to Agile Transformation, Enterprise Innovation, and Productivity

A Comprehensive Guide to Agile Transformation, Enterprise Innovation, and Productivity

By
Jayant Nath
Muralidhar Kurni
Srinivasa K. G.
Saurabh Suman Choudhuri

Cambridge
Scholars
Publishing

A Comprehensive Guide to Agile Transformation, Enterprise Innovation, and Productivity

By Jayant Nath, Muralidhar Kurni, Srinivasa K. G.
and Saurabh Suman Choudhuri

This book first published 2024

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-0364-0496-X
ISBN (13): 978-1-0364-0496-3

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Abstract

This chapter provides a comprehensive overview of Agile methodology, a product development and project management approach that prioritizes flexibility, collaboration, and customer satisfaction. The chapter outlines the values and principles of the Agile Manifesto, which serves as the foundation for Agile methodologies. It also explains the key features of Agile, such as its iterative approach, working software focus, and customer collaboration. The chapter concludes by highlighting the various Agile methodologies, such as Scrum, Kanban, and Lean, and how they all share the same core values and principles of the Agile Manifesto. This chapter is valuable for individuals and teams looking to understand and implement Agile methodology in their projects.

Introduction

In a constantly shifting world, where the only thing is change, traditional project management and software development approaches frequently find it challenging to stay up. This is where the Agile methodology comes into play; it is a method that is both flexible and adaptable, and it has completely changed how teams work and deliver outcomes. In this chapter, we will look into the fundamental ideas behind Agile.

What is Agile?

Agile is an iterative methodology for managing software projects and delivering features to clients more quickly. (Agile Alliance 2023). Instead of risking everything on a single "big bang" release, an agile team is expected to deliver smaller, more manageable chunks of work.

Agile approaches streamline project management and iterative software development, allowing businesses to provide more value to customers in

less time. Instead of risking everything on a single "big bang" release, an agile team prefers to roll out their work in incremental, more manageable chunks. When teams routinely assess requirements, plans, and outcomes, they can better adapt quickly to new circumstances.

Key Agile Concepts

The following are some of the essential Agile concepts.

- ***User Stories:*** Working with the customer or product owner, the team breaks down the project into smaller, more manageable chunks called "user stories." Each user story ought to contribute something substantial to the result.
- ***Daily Stand Up:*** The group gets together daily at the same time to report on their latest progress impediments and coordinate their efforts.
- ***Personas:*** The team will construct comprehensive, hypothetical user biographies for the project's target audience as needed.
- ***Team:*** In Agile, a "team" refers to a small group of people working together on the same project or effort, with most members working full-time.
- I***ncremental Development:*** In an Agile environment, incremental development is favored. With each new iteration, the product is enhanced in a way that is immediately apparent to the end user.
- ***Iterative development:*** Agile initiatives embrace iterative development, in which software development tasks are performed repeatedly, and the exact deliverables may be reviewed and revised.
- ***Milestone Retrospective:*** At the end of a project's run, the team takes one to three days to reflect on what went right and what could have gone better.

How Does Agile Work?

Here is how Agile functions.

- ***Define the problem and expected outcome:*** Keeping the customer at the center and looking at how they get their jobs done, the team and the client define the problem to be solved and construct a hypothesis on how this problem can be solved by building a specific solution. Along with this, some key outcomes are defined

along with leading indicators that tell whether the hypothesis is being proved right or wrong.

- ***Create a backlog:*** Product management converts this hypothesis into a product backlog of jobs ordered by priority. The backlog results from collaboration between the client, product owner, and development staff.
- ***Plan the sprint:*** Priority tasks from the backlog are chosen, and the team estimates how much work can be accomplished during the forthcoming sprint.
- ***Execute the sprint:*** Daily meetings are held to check on progress and resolve any issues that may arise as the team works to complete the sprint's intended tasks.
- ***Review and demo:*** At the end of the sprint, the team shows the customer the finished product and asks for feedback.
- ***Retrospect:*** The group analyses the sprint's results and plans to enhance future sprints based on what worked and what did not.
- ***Repeat:*** This procedure is repeated after every sprint for the project's duration. The product is built and shipped to the customer in manageable portions.
- ***Continuously improve:*** Agile approaches emphasize constant refinement. After reviewing their work, the team makes any necessary changes to the upcoming sprint's workflow, resources, and communication channels.

What are the Benefits of Agile?

The following (as shown in Fig. 1.1) are the key benefits of agile (S 2023):

- ***Customer:*** The manufacturer seems more receptive to customers' suggestions for new features. Shorter cycles to develop and deliver high-value features are more efficient than the lengthy cycles favored by traditional "waterfall" techniques.
- ***Benefits to Vendors:*** By focusing development efforts on features with the highest potential value, vendors can reduce waste and shorten time to market compared to waterfall procedures. If you can increase your customers' happiness, you can keep more of them and get more recommendations from them.
- ***Benefits to Development Teams:*** Team members take pride in their development efforts and appreciate feedback on their progress. Scrum helps team members by freeing them from unproductive tasks (such as creating specifications or other artifacts

that no one uses) and allowing them to focus on what they enjoy doing. Team members feel confident in their contributions because requirements are selected to optimize customer value.

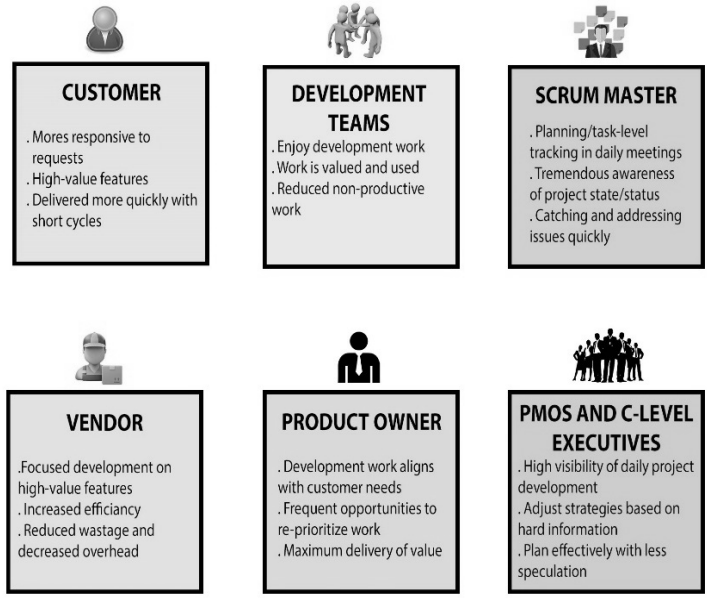

Fig. 1.1. Key benefits of agile.

- ***Benefits to Product Managers:*** Product Managers, who often take on the responsibilities of the Product Owner, are tasked with maximizing customer satisfaction by guaranteeing that all development efforts are directed towards satisfying those needs. Scrum facilitates this convergence by giving regular reprioritization opportunities to ensure the highest possible value is delivered to customers.
- ***Transparency without additional overhead:*** Scrum Masters, or others tasked with similar responsibilities, often report that planning and monitoring are more approachable and concrete than waterfall procedures. Every team member has a high level of situational awareness thanks to the Daily Standups, the emphasis on task-level tracking, and the usage of Burndown Charts to highlight daily progress. Leaders have a clear view of progress and can spot and fix problems as soon as possible due to this level of consciousness.

- ***Benefits to PMOs and C-Level Executives:*** With Scrum, you can always know where your development project stands. This insight is helpful for external stakeholders, such as C-Level executives and PMO staff, because it allows them to make better-informed decisions and refine their strategy.

What is Agility?

Agility is derived from the word Agile, which means responsive to change (Agile Alliance 2023) (Scaled Agile Inc. 2023). Let us see what some prominent business leaders say about (as shown in Fig. 1.2) what agility means to them.

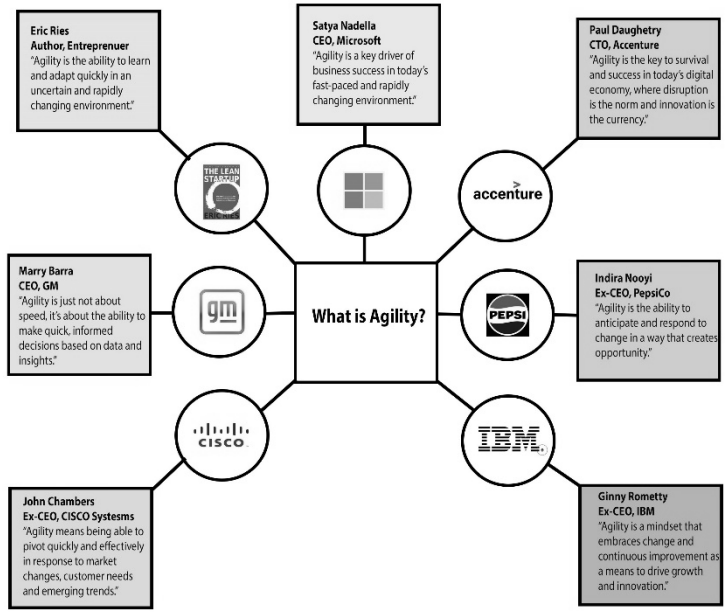

Fig. 1.2. What is Agility?

Agility is a concept that originated in software development but has since been adopted by many other industries and fields. At its core, agility refers to responding quickly and effectively to changing circumstances or requirements (S 2023).

In software development, agility is typically associated with the Agile methodology, which emphasizes iterative development, continuous delivery,

and a focus on customer needs and feedback (Cprime 2023). Agile teams work in short sprints, typically two to four weeks, and prioritize delivering working software that meets customer needs over following a rigid plan (Peek 2023).

However, agility can also apply to other areas, such as project management, business strategy, and organizational management. In these contexts, agility often focuses on adaptability, flexibility, and responsiveness to changing conditions (Pieper 2023).

Agility is Built on Five Pillars

Here, we will outline five key pillars (as shown in Fig. 1.3) on which agility is built.

1. ***Customer focus:*** Prioritizing the customer's or end-user's needs and providing them value.
2. ***Iterative development involves breaking*** problems into manageable chunks and continuously delivering working products or solutions.
3. ***Flexibility and adaptability:*** Pivoting quickly to changing circumstances or requirements.
4. ***Collaboration and teamwork:*** Working closely with others to achieve shared goals.
5. ***Continuous improvement:*** A commitment to ongoing learning, feedback, and progress.

The foundation of building agility is a mindset or approach that emphasizes responsiveness, adaptability, and customer focus. It can be applied in various contexts to help organizations be more effective and successful.

Agile relies on cross-functional, self-organizing teams to refine requirements progressively and develop solutions. Agile methodologies are iterative, meaning they focus on delivering small portions of a project and continuously improving the work as it progresses. This allows the project to adapt to changing requirements and enables the customer to provide feedback throughout the process.

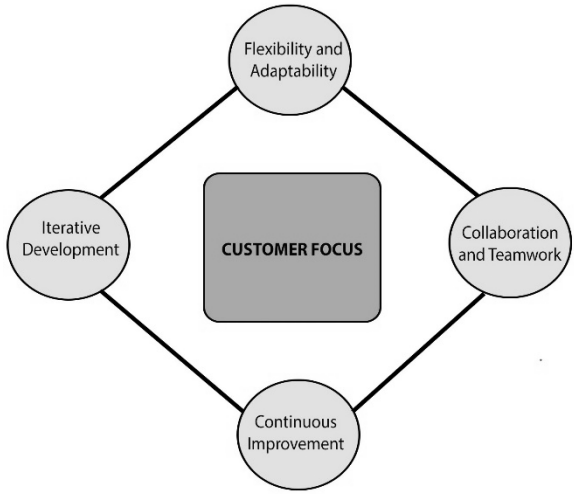

Fig. 1.3. Five Key Pillars of Agility.

Why Agility is Important?

Organizations and leaders have realized that agility enables them to respond to changing circumstances and customer needs quickly and effectively. In today's rapidly changing business environment, organizations that can adapt and innovate are more likely to succeed and remain competitive. Here are some of the key reasons for its importance (Leadership Tribe 2023) (Salesforce 2023):

1. ***Faster time-to-market:*** Agile development enables organizations to deliver working software and other products more quickly, allowing them to respond promptly to changing market conditions and customer needs.
2. ***Increased customer satisfaction:*** Agile development values customer collaboration and feedback, enabling organizations to build products that better meet customer needs and expectations.
3. ***Improved quality:*** Agile development focuses on delivering working software frequently and continuously improving through feedback and testing, resulting in higher quality products.
4. ***Greater flexibility:*** Agile development enables organizations to adapt to changing requirements and circumstances more efficiently, reducing the risk of project failure or delays.

5. ***Better team collaboration:*** Agile development emphasizes teamwork, collaboration, and communication, creating a more positive and productive team culture that leads to better outcomes.
6. ***Enhanced innovation:*** Agile development encourages experimentation and creativity, enabling organizations to explore new ideas and solutions more quickly.
7. ***Better decision-making:*** As decision-making is moved closer to the people working on the solution, it results in more effective decisions, better solutions, and more significant business outcomes.

Agility is essential because it helps organizations stay responsive, competitive, and innovative in a rapidly changing business environment. Most Fortune 500 companies mention Agility or Agile in their annual reports and strategic roadmaps.

How is Agility Changing the Way of Work?

Agility has profoundly impacted our work today by promoting a more flexible, collaborative, and customer-focused approach to software development and other business areas. Some of the critical ways that agility is changing the course of work:

1. ***Cross-functional teams:*** Agile development promotes cross-functional teams that include developers, testers, designers, and other roles. This enables teams to work more collaboratively and reduces silos between different departments.
2. ***Iterative development:*** Agile development emphasizes an iterative, incremental approach to development, with regular feedback and testing. Because of this, teams are better equipped to adapt to the demands of their customers and meet their evolving expectations.
3. ***Continuous delivery:*** Agile development promotes a continuous delivery approach, delivering software frequently and consistently. This reduces the time and risk associated with significant, infrequent releases.
4. ***Customer collaboration:*** Agile development emphasizes close collaboration with customers and stakeholders, ensuring that products meet customer needs and expectations.
5. ***Emphasis on people:*** Agile development strongly focuses on people and their interactions, recognizing that motivated and

engaged teams are critical to project success.

6. ***Continuous improvement:*** Agile development promotes a culture of constant improvement, with regular reflection and adaptation based on feedback and data. This enables teams to learn from their experiences and continuously improve their processes and outcomes.

By causing a shift in the mindset with a more customer-focused, flexible, and collaborative approach to delivering business value, companies are reaping the benefits of more favorable outcomes, more significant innovation, and increased organizational success.

Agile Values, Principles, and Methodologies

The Agile Values are a set of guiding principles that underpin the Agile methodology and its various frameworks. The Agile Values were first articulated in the Agile Manifesto, a document created by software developers in 2001. The four Agile Values (as shown in Fig. 1.4) are (Eby 2016):

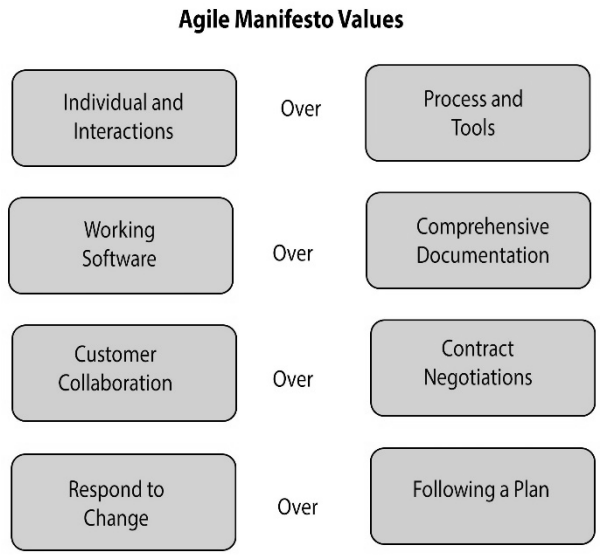

Fig. 1.4. Agile Manifesto Values (Quick 2023).

1. ***Individuals and interactions over processes and tools:*** Agile values collaboration, communication, and teamwork and emphasizes the importance of human interaction and relationships in achieving success.
2. ***Working software over comprehensive documentation:*** Agile places a premium on producing usable software that satisfies client expectations rather than documenting every process piece.
3. ***Prioritizing client cooperation above contract negotiation:*** Agile values customer and end-user participation in development and emphasizes customer communication and collaboration to fulfill their needs.
4. ***Instead of sticking to a plan, adapting to changes: Agile highly values*** flexibility, adaptability, and responsiveness to changing requirements or circumstances. Agile teams prioritize delivering working software quickly and adapting to evolving needs or feedback rather than following a rigid plan.

These four Agile Values are supported by 12 Agile Principles, which provide more detailed guidance for implementing the Agile methodology. The 12 Agile Principles (as shown in Fig. 1.5) are (Eby 2016):

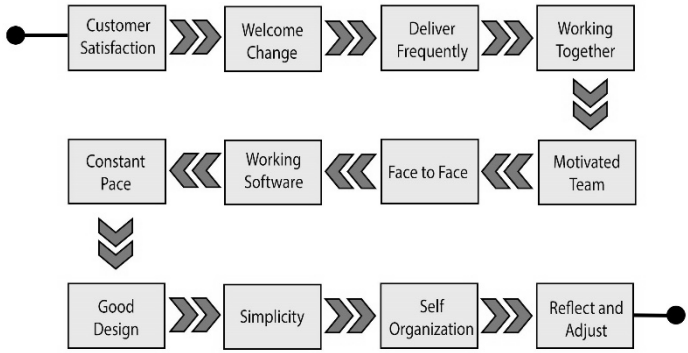

Fig. 1.5. Agile Principles (Quick 2023).

1. Software must be delivered quickly and reliably to ensure customer satisfaction.
2. The ability to adapt to new requirements during development.
3. Maintain a focus on rapid iteration and frequent delivery of working software.

4. Maintain close contact with clients and interested parties as the project progresses.
5. Collaborate on projects with driven individuals and ensure their success by giving them the necessary trust and support.
6. Face-to-face meetings should be used whenever possible to facilitate understanding and teamwork.
7. Working software is the fundamental indicator of progress.
8. Encourage a consistent pace and concentration on technical excellence to further sustainable development.
9. Prioritize ease of use and minimize effort expended.
10. Allow teams to self-organize and have faith in their ability to make wise choices.
11. Team members should continually assess their actions and adjust to maximize efficiency.
12. Feedback, experimentation, and adjustments should be made continuously to enhance the process.

Together, the Agile Values and Principles provide a framework for developing software and other products that are responsive to customer needs, flexible, and adaptable to changing requirements and circumstances.

Some of the popular Agile frameworks and methodologies adopted in the industry are (S 2023) (AltexSoft 2023)**:**

- ***Scrum:*** A popular Agile framework emphasizing collaboration, self-organization, and iterative development. Scrum teams work in sprints of two to four weeks and are guided by a set of defined roles, ceremonies, and artifacts.
- ***Kanban:*** An Agile framework that emphasizes visualizing work, limiting work in progress, and delivering small, frequent releases. Kanban teams use a Kanban board to track work items and their progress through different stages of development.
- ***Lean:*** An Agile framework that emphasizes maximizing customer value while minimizing waste. Lean teams focus on continuous improvement and the elimination of non-value-adding activities.
- ***Extreme Programming (XP):*** Continuous integration, pair programming, and test-driven development are some of the key principles of an Agile methodology. Quick iterations are a hallmark of the XP methodology, which helps teams concentrate on providing customers with high-quality software.
- ***Crystal:*** An Agile framework that emphasizes the importance of communication, simplicity, and reflection. Crystal teams are highly

adaptable and tailor their practices to the specific needs of the project and team.

- ***Feature-Driven Development (FDD):*** An Agile framework that emphasizes a structured, iterative approach to feature development. FDD teams focus on delivering high-quality features that meet customer needs and use defined practices and processes to guide development.

Each Agile framework has its specific practices, roles, and processes, but all share the core values and principles of the Agile methodology. Organizations can choose the framework that best fits their needs and adapt it to their specific circumstances and goals.

The Agile Software Development Life Cycle

A product passes through a defined sequence of steps known as the Agile software development life cycle (Wrike 2023) (as shown in Fig. 1.6) as it evolves from inception to completion. There are six stages: concept, inception, iteration, release, maintenance, and retirement.

1. ***Concept:*** This process begins with the concept phase. Here, a product owner will lay out the parameters for their project. Priority will be given to the most important initiatives if there are numerous. After meeting with the customer to go over the most important needs, the product owner will describe those specifications in great detail, including all of the features and the results that may be anticipated. Reduce the list of needs as much as possible; more can always be added. While brainstorming potential projects, the product owner often calculates rough costs and timelines. This analysis will help them decide if the project is worth continuing.
2. ***Inception:*** Putting together the software development team can begin once the concept has been defined. A product owner will find out who is available to work on the project, select the best candidates, and then give them all the tools they need to finish. At that point, we can start working on the designs. A user interface mockup will be made, and the project's infrastructure will be built. The inception phase solicits more feedback from stakeholders to define the product's features and flesh out the requirements diagram. Reviews should be conducted frequently to ensure all needs are considered during the design process.

3. ***Iteration:*** The next step is the iterative phase, often called the construction phase. In this stage, most of the effort is expended. The UX designers and developers will include all product requirements and user feedback when working on the final code. The product is expected to have little functionality by the conclusion of the first sprint. Potentially, more features and tweaks will appear in later updates. Agile software development relies heavily on this phase, which allows for rapid prototyping and iterative refinement in service of a happy client.

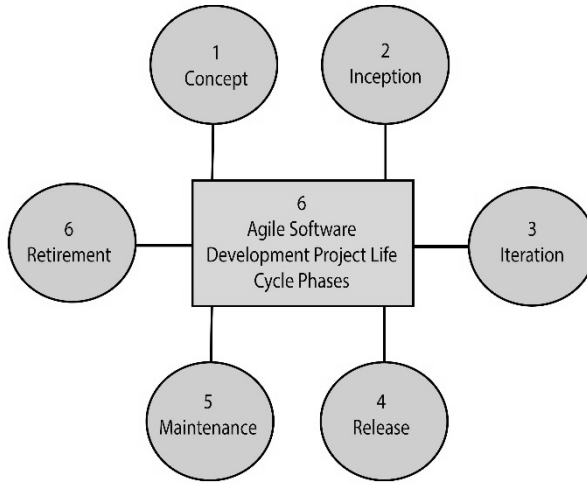

Fig. 1.6. Agile-Software-Development-Project-Life-Cycle-Phases.

4. ***Release:*** The product is getting close to its release time. However, the quality assurance department must thoroughly test the product before release. This Agile team will perform system tests to ensure the code is error-free; if any issues are discovered, they will be resolved promptly. More records will also be needed at this level for user training. Doing so will allow the final product to be released to production.
5. ***Maintenance:*** Everything is now in place for the software to be released to customers. With this action, we go into the upkeep stage. The software development team continuously strives to ensure the system's stability and the absence of new flaws. These people will also provide extra training and help with using the device. New product versions might be released periodically to fix

bugs and include new functionality.

6. ***Retirement:*** The two leading causes of a product's retirement are introducing newer software that can do the same or the system's age and subsequent incompatibility with the company's needs. As soon as the development team discontinues the software, they notify users. Substitutes will be switched if they are available. When this project is ready, the developers will stop adding new features to the outdated program and finish all remaining end-of-life chores.

Numerous iterations are performed during each stage of the Agile life cycle to improve deliveries and provide superior outcomes. Let us examine the inner workings of this iterative process in each stage.

The Agile Iteration Workflow

Agile iterations have a fixed duration, typically between two and four weeks. Agile iterations usually follow a five-step process flow (Wrike 2023):

- Plan requirements
- Develop product
- Test Software
- Deliver iteration
- Incorporate feedback

Developers will go through several iterations during each Agile phase to create the greatest software possible. Iterations are essentially mini-cycles within the larger Agile development life cycle.

The Agile life cycle, which keeps teams on track from conception through retirement, is an essential structural paradigm for software development teams. Agile project management software is one resource and tool that should be available to all team members to facilitate all activities in the Agile cycle.

The Advantages and Disadvantages of Agile

As with any methodology, agile has advantages and disadvantages (as shown in Fig. 1.7).

Advantages of Agile Methodology

The advantages of Agile Methodology, as outlined by the Agile Alliance (University of Minnesota 2022):

- Early and consistent delivery of high-quality software is one of the top priorities to ensure customer satisfaction. The priority is to provide excellent service and products to the customers.
- Allow for requirements to evolve even late in the design process. Agile methodologies are open to change to give the client a competitive advantage. Stop resisting change and figure out how to work with it.
- Release software regularly, ideally every few weeks or months at most. Do not deliver results only at the end of a project; do so continuously.

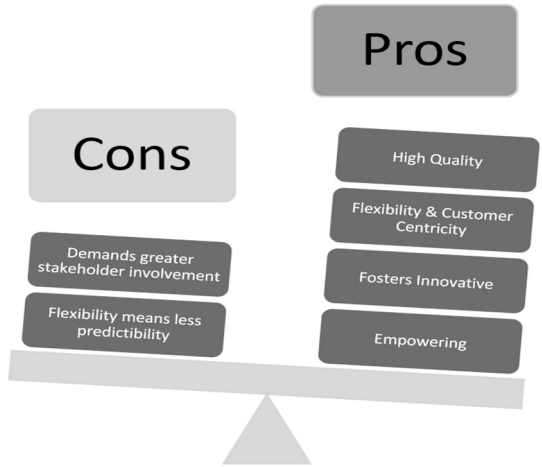

Fig. 1.7. Pros and Cons of Agile Methodology.

- Throughout the project, business leaders and developers must collaborate every day. Working together is crucial.
- Develop your projects around motivated individuals. Provide the resources they need to succeed, and have faith in their abilities to complete the task. Acquire people of skill and diligence, then step aside.

- Direct, in-person communication is the best way to deliver knowledge to and within a development team. Reduce the number of probable misunderstandings.
- Successfully functioning software is the fundamental indicator of success. Efficacy is more important than perfection here.
- Agile methods aid in long-term growth and prosperity. Sponsors, developers, and users should be able to keep going at the same rate forever. When you take things slowly and steadily, you end up ahead.
- A constant focus improves agility on technical superiority and sound design. Keep in mind the importance of paying close attention to detail.
- The ability to keep things straightforward is known as simplicity. Trim the fat.
- Self-organizing teams produce superior architectures, requirements, and designs. To get the most out of your team, you must give them the freedom to define their roles.
- Each team member regularly considers how they might improve the team's performance and then takes action to do so. Solicit and offer feedback, reflect on it, and make adjustments as necessary.

Disadvantages of Agile Methodology

Just like any other project management approach, Agile has its flaws. Agile's inherent strengths can be a team's downfall if the methodology is improper. Some of Agile Methodology's more common criticisms are outlined in a recent issue of Inc. magazine (University of Minnesota 2022):

- ***Less predictable.*** The Agile methodology's foundational flexibility also leads to far less predictability. It is significantly more difficult to accurately predict how long and what materials will be required to complete a project. Many groups dread this unpredictability, which can cause stress and poor decision-making.
- ***More time and commitment.*** While talking to one another and working together is beneficial, it does require more effort and time from everyone concerned.
- ***Greater demands on developers and clients.*** Active engagement from all stakeholders is crucial for the success of the Agile Methodology. The output could suffer due to the input of individuals who are not fully committed.

- ***Lack of necessary documentation.*** In the Agile development process, tasks are typically finished just before they are developed, which can lead to sloppy documentation and confusion.
- ***Projects easily fall off track.*** Due to its less organized nature, Agile Methodology can cause projects to veer off track or swiftly expand beyond their initial scope.

Beyond Software Development – Agile at Scale

Along with software development, the Agile mindset can be applied in several industries. The practices, methods, and tools used in Agile need to be adapted to each industry. For instance, iterations in the Software industry can be 2-weeks-long. This is not feasible in the automotive or aeronautics industry, where the iterations will be instead of several months period before delivering an MVP. Note that most industries today have embedded systems with embedded software. The iteration cycle of an embedded software MVP can be shorter (Koteich 2020).

Agile Development Example

Agile software development uses iterative approaches and continual improvement. This implies repeatedly attempting something until it works.

The plan is to create the first working model of a product component. This release is not supposed to be flawless, but it is usable for users to test real-world scenarios. This is the most efficient method for finding and fixing problems or unfulfilled requirements.

On the other hand, incremental development is prompted by small steps. The work is divided into horizontal sections or sites using a waterfall approach. The purpose of the agile vertical incremental technique is to construct functional parts progressively.

Consider the production of a car (as shown in Fig. 1.8) as an illustration. (Singh 2020):

- In a horizontal approach, the wheels come first, followed by the body, the engine, the headlights, etc.
- A basic car that runs on essential components might be constructed vertically. Secondary amenities (comfortable seats, paint, bright lights, a GPS, etc.) are added to each component to enhance it further.

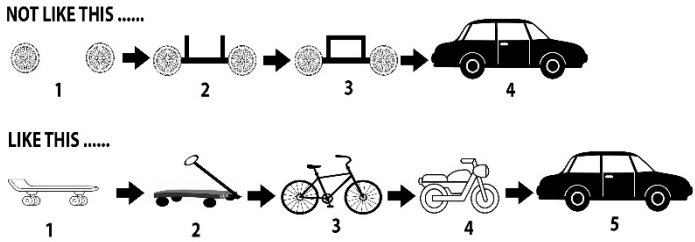

Fig. 1.8. Agile development example.

Misconceptions about Agile

Some things about Agility need to be clarified. Here are some of those myths (Koteich 2020):

- ***Do not create documentation:*** This may lead to wasted time. Agile means focusing on the product and taking your time with the documents. The working product measures progress.
- ***You do not need a plan:*** Planning is crucial in Agile practices; we need somewhat realistic iterative methods. Every iteration should have a good plan.
- ***No management, no discipline:*** The Agile approach needs project management and discipline, but it requires a different mindset than the old-school approach. Instead of using top-down management, leaders must delegate at the right hierarchy level to make decisions.
- ***Agile = Scrum:*** Scrum is only one practice that embraces the Agile mindset. People are often uncomfortable with abstract concepts and prefer more concrete structures. Agile is often mistaken for some practices and methods (Scrum, Kanban, etc.). Be careful; the team spirit and mindset matter, not the methods and tools they use.
- ***It works only for Software development:*** as a mindset, it can be applied to any product development, including heavy and complex industries.
- ***It works only for young employees:*** wrong.
- ***Agile is meant to save money:*** Agile is about adapting to change and satisfying customer needs, which may reduce development costs. However, Agile might result in rework in some contexts and require more money to provide an excellent product rather than saving money to produce a mediocre output.

Conclusion

The Agile Method is famous for managing projects in and outside software development. Due to its flexibility and adaptability, it is the most suitable method of project management for the present day. The proper use of Agile Methodology principles can lead to several beneficial consequences. The COVID-19 epidemic has made it evident how important it is to be flexible in the face of unpredictability. Agile methodology is not always the best option, but its values may be judiciously applied to any project to keep the team agile and ready for anything.

Bibliography

Agile Alliance. 2023. "Agile 101." Agile Alliance. 2023. https://www.agilealliance.org/agile101/.

AltexSoft. 2023. "Agile Project Management: Best Practices and Methodologies." AltexSoft. 2023. https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/.

Cprime, Inc. 2023. "What Is Agile? What Is Scrum?" Cprime, Inc. 2023. https://www.cprime.com/resources/what-is-agile-what-is-scrum/.

Eby, Kate. 2016. "Comprehensive Guide to the Agile Manifesto." Smartsheet Inc. 2016. https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto#the-four-values-of-the-agile-manifesto.

Koteich, Mohamad. 2020. "Agility – Beyond Software Development." LinkedIn. 2020. https://www.linkedin.com/pulse/agility-beyond-software-development-mohamad-koteich/.

Leadership Tribe. 2023. "5 Reasons Why Agility Is More Important Than Ever in the Enterprise." Leadership Tribe. 2023. https://leadershiptribe.in/blog/5-reasons-why-agility-is-more-important-than-ever-in-the-enterprise.

Peek, Sean. 2023. "What Is Agile Scrum Methodology?" Business News Dailyaily. 2023. https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html.

Pieper, Robert. 2023. "Agile Transformation." Responsive Advisors, Inc. 2023. https://responsiveadvisors.com/agile-transformation/.

Quick, Lindy. 2023. “The Ultimate Guide to the Agile Manifesto: 12 Agile Manifesto Principles.” Knowledgehut Solutions Private Limited. 2023. https://www.knowledgehut.com/blog/agile/agile-manifesto-principles-values.

S, Bhaskar. 2023. “What Is Agile Methodology?” NimbleWork, Inc. 2023. https://www.nimblework.com/agile/agile-methodology/.

Salesforce, Inc. 2023. “Why Agility Is an Important Part of Your Company Culture.” Salesforce, Inc. 2023. https://www.salesforce.com/ca/blog/2021/07/why-agility-is-an-important-part-of-your-company-culture.html.

Scaled Agile Inc. 2023. “Organizational Agility.” Scaled Agile Inc. 2023. https://scaledagileframework.com/organizational-agility/.

Singh, Ricardo. 2020. “5 Advantages & Disadvantages of Using an Agile Methodology.” Appvizer. 2020. https://www.appvizer.co.uk/magazine/operations/project-management/advantages-of-agile-methodology.

University of Minnesota. 2022. “Agile Methodology: Advantages and Disadvantages.” University of Minnesota. 2022. https://ccaps.umn.edu/story/agile-methodology-advantages-and-disadvantages.

Wrike, Inc. 2023. “The Agile Software Development Life Cycle.” Wrike, Inc. 2023. https://www.wrike.com/agile-guide/agile-development-life-cycle/.

CHAPTER 2

WHAT BROUGHT US TO AGILITY?

Abstract

This chapter explores the reasons behind the shift towards agile frameworks in product development and project management. The traditional, rigid approaches to project management, such as the Waterfall model, were designed for a stable work environment where requirements were well-defined and unlikely to change. However, the fast pace of technological change, the globalization of markets, and the increasing complexity of projects have all created a need for a more flexible and adaptable approach. This chapter highlights the limitations of traditional approaches and the benefits of adopting an agile approach, including improved customer satisfaction, increased collaboration and flexibility, and the ability to respond to change effectively. The chapter provides a comprehensive overview of the drivers that have led us to adopt Agility in project management, making it a valuable resource for individuals and teams seeking to understand the reasons behind the shift towards agile approaches. This chapter also provides some illustrations.

The Origins - The Waterfall Struggle

During the infancy of software development in the 1960s, its framework, standards, and best practices had to evolve from scratch. With a lack of any precedents, there was a need to borrow some successful processes from manufacturing. This brought about the Waterfall model - a linear sequential development approach first proposed in 1970 by Winston Royce (Haham 2019). This model was heavily influenced by the manufacturing and construction industries, which followed a sequential and structured process for project execution. The idea was to apply similar principles to software development to ensure that projects were completed systematically and predictably in a series of sequential phases, with each phase building on the output of the previous phase. It gained popularity in the 1970s and 1980s because it provided a structured approach to software development, which

was crucial when the field was still relatively new (Haham 2019). However, the model's limitations became apparent over time, particularly its inflexibility and inability to adapt to changing requirements.

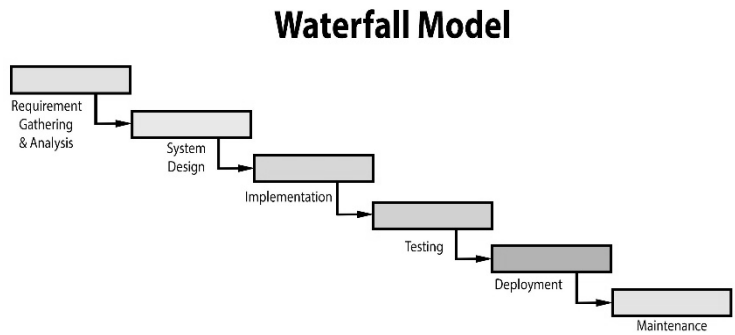

Fig. 2.1. The Waterfall Model.

The Waterfall model (as shown in Fig. 2.1) has been criticized for several reasons, including (Mashimo 2009) (Albani 2016) (Lucidchart 2023):

1. It needs to be more flexible, and it can be challenging to adapt to changes in requirements.
2. It can be slow and inefficient, requiring all phases to be completed before product release.
3. It can be challenging to manage, as it requires a clear understanding of the requirements and the ability to track progress across all phases.
4. Unlike manufacturing, the underlying technology, tools, languages, and hardware in software engineering change ever-increasingly. This warranted constant technology and feature upgrades, and the waterfall model proved to be a bottleneck.
5. The biggest drawback is that validation and feedback should be done on time in the game, tremendously increasing the risk of project failure.

From a business point of view, in these ever-changing market dynamics and customer preferences, timing the big bang launch of a finished product with the market demand became a pipe dream for most organizations. Several high-profile failures of projects have used the Waterfall model (Sutherland 2012). Some examples include:

- The development of the F-35 fighter jet was delayed and over budget.
- The development of the Millennium Dome, a major tourist attraction, failed to attract the expected number of visitors.
- The development of the Segway was a self-balancing scooter that was hyped as a major innovation but failed to find a large market.

Some more Examples

The FBI's Virtual Case File (VCF) project was launched in 2001, with an initial budget of $380 million and an expected completion date in 2003. The VCF project followed a Waterfall-like approach, focusing on defining all the requirements upfront and moving through a sequential process.

The VCF project encountered numerous issues that are common in the waterfall model. These include lack of clarity because of poor communication, inability to accommodate changes to adapt to the evolving needs of the FBI, and testing only in the end, which led to the discovery of a multitude of issues too late in the process.

The VCF project was eventually deemed a failure and was abandoned in 2005 after spending nearly $170 million without delivering a functional system. It was replaced by a new project called Sentinel, which adopted an Agile approach to development, and the plan was successfully deployed in 2012.

While the Waterfall Model was not solely responsible for the VCF project's failure, its rigidity and inability to adapt to changing requirements significantly influenced its challenges. This example highlights the risks of using the Waterfall Model for complex, high-profile projects with evolving needs.

The market moves swiftly to where there is value, and the waterfall approach needs to be more flexible to allow organizations to quickly adapt to new market dynamics, customer demands, and technological advances. Hence, there was a need for a radical change in the value discovery and delivery approach. At the same time, with the ever-increasing importance of technology in organizations' success, it was natural that the next level of organizational evolution originated from the technology side – the onset of agile frameworks.

With its emphasis on the customer first, collaboration, fast feedback cycle, and economic consideration, this agile approach proved refreshing, fulfilling, and successful. As a result, many businesses are now turning to agile frameworks such as Scrum, Kanban, and Lean Startup.

Contrast these with the advantages of Agility (The CRM Team 2018)

(Ciuła 2021) (Hoory, Bottorff, and Watts 2022):

- Agile frameworks are more flexible and adaptable. They allow teams to respond to changes quickly and easily, which is essential in today's fast-paced world.
- Collaboration is more prevalent in agile frameworks. They promote teamwork and idea-sharing, which in turn can improve decision-making and provide more creative outcomes.
- One advantage of agile frameworks is their increased emphasis on the client. A higher level of customer satisfaction and loyalty may result from teams' increased concentration on meeting customer needs through the delivery of products.
- Agility brings work to the persistent, cross-functional, self-organized team instead of creating a new team for every project and getting the job to them.
- Enterprise agility brings alignment between the company strategy and execution at the team level.

As any product manager will agree, predicting user behavior and preferences is challenging. According to a study by the Standish Group in their 2002 "CHAOS Report," around 64% of software features were rarely or never used, while only about 20% were used often or always.

According to Bruce Tuckman's (1965) theory of team dynamics, every newly formed team goes through the stages of forming, storming, norming, and performance (as shown in Fig. 2.2). As the team navigates through the forming, storming, and forming stages, it finally moves to the performing stage to derive the most significant benefits of a collaborative team.

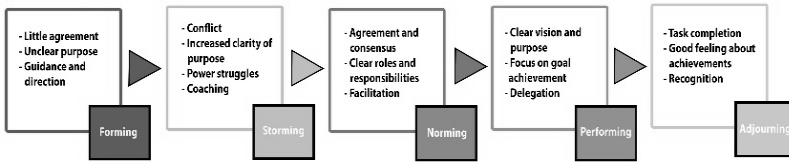

Fig. 2.2. Team Development Stages (Okpalad 2015).

In a waterfall model of project-based team creation, every project goes through the first three stages for most of the project duration before they can start performing. Moreover, as some (not all) teams reach the performing stage, the project is often closed. With the successful or otherwise completion of the project, the team is disbanded, the high productivity