

Agile, Scrum Framework, Benefits and Its Use in Software Projects, Estimation and Risk Management

Seema Kalwani

Security Engineer, IL, USA

seemakalwani@gmail.com,

Abstract

The article provides overview covering Agile framework and discussing scrum a prominent practice of agile methodology. The pillars, principles and benefits of using agile in software projects is covered. The concepts of scrum, different roles, planning, execution along with estimation and risk management is described for overall understanding.

Keywords: Agile, Scrum, Estimation, Scrum Master, Risk Management, Scrum master, collaboration, Product Backlog, Sprint, Retrospection, planning, release management, servant leader

I. INTRODUCTION TO AGILE FRAMEWORK

Agile methodology is a project management framework that breaks projects down into several dynamic phases, commonly known as sprints. In this article, get a high-level overview of Agile project management, plus a few common frameworks to choose the right one for your team.

Agile approaches empower teams to adapt quickly, accelerate delivery, and drive meaningful results. By breaking down silos and fostering collaboration, agile approaches increase efficiency, reduce risk, and ensure continuous innovation.



Fig. 1. Picture depicting agile framework from asana

1) *4 pillars of agile - There are 4 main values of agile Project management*

- Individuals over processes and tools. Agile teams value team collaboration and teamwork over working independently and doing things "by the book."
- Working software over comprehensive documentation. The software that Agile teams develop should work. Additional work, like documentation, is not as important as developing good software.
- Customer collaboration over contract negotiation. Customers are extremely important within the Agile methodology. Agile teams allow customers to guide where the software should go. Therefore, customer collaboration is more important than the finer details of contract negotiation.
- Responding to change over following a plan. One of the major benefits of Agile project management is that it allows teams to be flexible. This framework allows for teams to quickly shift strategies and workflows without derailing an entire project.

2) 12 principles of agile - If the four values of the Agile model are the weight-bearing pillars of a house, then the 12 Agile principles are the rooms you can build within that house. These principles can be easily adapted to fit the needs of your software development process.

- Satisfy customers through early, continuous improvement and delivery. When customers receive new updates regularly, they're more likely to see the changes they want within the product. This leads to happier, more satisfied customers—and more recurring revenue.
- Welcome changing requirements, even late in the project. The Agile framework is all about adaptability. In iterative approaches like Agile, being inflexible causes more harm than good.
- Deliver value frequently. Similar to principle #1, continuous delivery of value to your customers or stakeholders frequently makes it less likely for them to churn.
- Break the silos of your projects. Cross-functional teams and collaboration is a key Agile value. The goal is for people to break out of their individual projects and collaborate more frequently.
- Build projects around motivated individuals. Agile management works best when teams are committed and actively working to achieve a goal.
- The most effective way to communicate is face-to-face. If you're working on a distributed team, spend time communicating in ways that involve face-to-face communication like Zoom calls or daily stand-up meetings.
- Working software is the primary measure of progress. The ultimate goal of software development projects is a working product, and the Agile framework supports this by prioritizing functional software above all.
- Maintain a sustainable working pace. Some aspects of Agile project management can be fast-paced, but it shouldn't be so fast that team members burn out. The goal is to maintain sustainability throughout the development process.

- Continuous excellence enhances agility. If the team develops excellent code in one sprint, they can continue to build off of it the next. Continually creating great work allows teams to move faster in the future.
- Simplicity is essential. Sometimes the simplest solution is the best solution. Agile development aims to not overcomplicate things and find simple answers to complex problems.
- Self-organizing teams generate the most value. Similar to principle #5, proactive teams become valuable assets to the company as they strive to deliver continuous improvement.
- Regularly reflect and adjust your way of work to improve effectiveness. Retrospective meetings are a common Agile practice. It's a dedicated time for Agile teams to look back and reflect on their performance and adapt their behaviors for the future.

II. BENEFITS OF THE AGILE DEVELOPMENT METHODOLOGY

Agile project management is used in application development or other types of software development. This is because software is constantly changing, and the needs of the product have to change with it. Because of this, linear project management methods like the waterfall model are less effective. Here are a few other reasons why teams use Agile:

1) Agile methods are adaptable

There's a reason why they call it the Agile methodology. One of the main benefits of using Agile processes in software development is the ability to shift strategies quickly, without disrupting the flow of a project.

Because phases in the traditional waterfall method flow into one another, shifting strategies is challenging and can disrupt the rest of the project roadmap. Since software development is a much more adaptable field, project managing rapid changes in the traditional sense can be challenging. This is part of the reason why Agile project management is favored in software development.

2) Agile fosters collaborative teamwork

One of the Agile principles states that the most effective way to communicate with your team is face-to-face. Combine this with the principle that encourages teams to break project silos and you have a recipe for collaborative teamwork.

While technology has changed since Agile's inception and work has shifted to welcome more remote-friendly policies, the idea of working face-to-face still hasn't changed.

3) Agile methods focus on customer needs

One of the unique aspects of software development is that teams can focus on customer needs much more closely than other industries. With the rise of cloud-based software, teams can get feedback from their actual customers quickly.

Since customer satisfaction is a key driver for software development, it's easy to see why it was included in the Agile process. By collaborating with customers, Agile teams can prioritize features that focus on

customer needs. When those needs change, teams can take an Agile approach and shift to a different project.

III. INTRODUCTION TO SCRUM

Scrum is an Agile framework that's been used to manage work on complex products since the early 1990s. Scrum is not a process, technique, or definitive method. Rather, it's an outline to refer to while employing various processes and techniques. Teams complete tasks in set periods of time called Sprints, typically less than 30 days each, with daily reassessments of progress and status. This approach is effective in overcoming obstacles often presented by unexpected challenges, including evolving customer input.

A. *Benefits*

- 1) *Higher quality of work at a lower cost (better ROI)*
- 2) *Higher customer satisfaction*
- 3) *Increased team collaboration and morale*

B. *Scrum Versus Agile*

Many organizations use a traditional top-down project management approach, also known as Waterfall. Despite the name, this technique is not always the most fluid way to go. Waterfall is more of a sequential method; the stages of a project are worked on in a consecutive order, from conception, to design, to implementation. The challenge with the Waterfall methodology is that once one step is complete, it's almost impossible to go back (and potentially improve) without scrapping the whole project. This approach costs time and money. Enter the Agile approach. As the most widely used framework within Agile, Scrum is used to break down complex projects into smaller pieces, thereby allowing your team to continuously deliver value on a more frequent basis. It's a more collaborative and flexible approach, so you can respond to your client's evolving needs and changes in the market. It's common for customer needs to evolve throughout the course of any project, and the Scrum approach enables more adaptability.

C. *Product Backlog*

The Product Backlog is the organization of items that guides what's most important to focus on and build next. The Product Backlog is constantly being updated and refined. Sprint At the heart of Scrum is the Sprint, during which a usable and potentially releasable product increment is created. Sprints are usually one week to one month in length, and happen one right after the other to keep projects moving. There are three events (ceremonies) that happen with each

D. *Sprint*

Sprint is usually 30-day period with a goal to accomplish a chunk of work.

- **SPRINT PLANNING** The team decides what to work on for the current time-boxed period.
- **SPRINT REVIEW** The team collaborates about what was done and adapts the Backlog as needed.
- **SPRING RETROSPECTIVE** The team discusses what went right, what went wrong, and how to improve.

- Daily Scrum (think “daily status meeting”) is the heart of the framework. Daily Scrums highlight and promote quick decision-making, and they improve the Scrum Team’s level of transparency and knowledge. The Daily Scrum is short — averaging about 15 minutes — so the team tends to be highly focused.
- Scrum Alliance — the largest, most established, and influential nonprofit professional membership and certification organization in the Agile community — exists to transform the way we tackle complex projects, bringing the Scrum framework and Agile principles beyond software development to the broader world of work. Our mission is to guide and inspire individuals, leaders, and organizations with practices, principles, and values that create workplaces that are joyful, prosperous, and sustainable.

E. Scrum Roles

1) Scrum Master

The ScrumMaster helps the Scrum Team perform at their highest level. They fully understand the underlying principles of Scrum and can effectively apply them in complex, real-world situations. They ensure that goals and scope of projects are clearly understood and work to remove impediments to progress.

- The scrum master is the role responsible for gluing everything together and ensuring that scrum is being done well. In practical terms, that means they help the product owner define value, the development team deliver the value, and the scrum team to get to get better. The scrum master is a servant leader which not only describes a supportive style of leadership but describes what they do on a day-to-day basis.
- They serve the product owner by helping them better understand and communicate value, to manage the backlog, help them plan the work with the team and break down that work to deliver the most effective learning. Serving the development team, the scrum master helps them self-organize, focus on outcomes, get to a “done increment,” and manage blockers. The scrum master also serves the organization at large, helping them understand what scrum is and create an environment that supports scrum.



Fig. 2. A diagram showing the scrum master's responsibilities from Atlassian

2) *Product Owner*

The Product Owner role is typically for those who are closest to the “business side” of projects. They are expected to do the best possible job of satisfying all stakeholders, maintain the Product Backlog, and ensure that everyone knows the priorities.

- Managing the scrum backlog - This does not mean that they are the only one putting in new product backlog Items into the backlog. But ultimately, they are responsible for the backlog that the development team pulls to deliver from. That means the product owner should know about everything that is in the backlog and other people that add items to the product backlog should ensure that they communicate with the product owner. We even offer a free scrum template that makes it easy to get your backlog up and running.
- Release management - The sprint is not a release cycle, but instead a planning cycle. That means that scrum teams can deliver at any time. Ideally, they would deliver frequently throughout the sprint allowing the sprint review to review real customer usage and feedback. However continuous delivery is not always possible and other release models are required. It is important for the product owner to know when things can and should be released.
- Stakeholder management - Any product will have many stakeholders involved ranging from users, customers, governance and organizational leadership. The product owner will have to work with all these people to effectively ensure that the development team is delivering value. That can mean a large amount of stakeholder management and communication.



Fig. 3. A diagram showing the product owner's responsibilities from Atlassian

3) *Development Team*

Development Teams are structured and empowered to organize and manage their own work. The resulting synergy optimizes the Scrum Team’s overall efficiency and effectiveness.



Fig. 4. A diagram showing development team's structure and operation from Atlassian

IV. SPRINT PLANNING & EXECUTION

Sprint planning is an event in Scrum that kicks off the Sprint. It's the first event that happens during a Sprint. The main agenda of Sprint planning is to define the scope of delivery and how to accomplish that work. It sets up a common goal for the team, and everyone's focus is to achieve that goal during the Sprint.

Sprint Planning answers the following:

- What can deliver in the Increment resulting from the Sprint? - The Scope
- How to accomplish the work needed to deliver the Increment? - The Plan
- We can split the Sprint Planning meeting into two parts - The Scope and The Plan. Let's understand each one of them in detail.

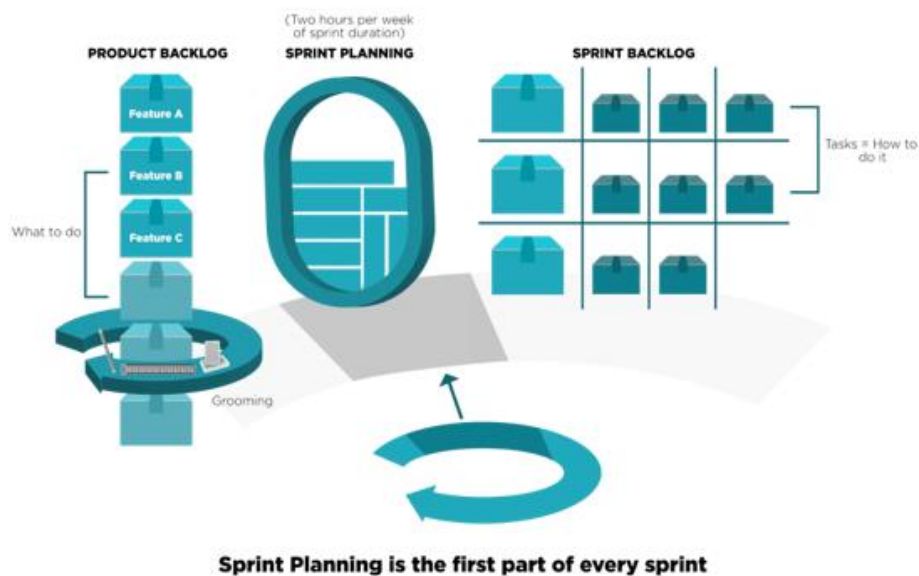


Fig. 5. Sprint planning picture from toolsqa

A. Define the scope of sprint

The First step of Sprint planning is to finalize the scope for the Sprint. The product owner plays an active role in helping out the development team to finalize the scope. Let's understand the steps involved to finalize this scope:

1) Define Sprint Goal -

Each Sprint needs to have a potentially releasable deliverable. How do we achieve that? By having a clearly defined Sprint goal. Imagine if we pick up half of the login, half of the registration, and few stories of the home page in a Sprint. Theoretically, we are still progressing, and work is getting done, but can we release it? Therefore, we must define the Sprint goal first.

Who defines it? The responsibility to determine the Sprint goal is with the product owner. However, the team is equally involved as well, so that they all are on the same page on the Sprint goal. The basis of the Sprint goal is the essential features of the product that needs to build first. One of the critical inputs in defining the Sprint goal is the latest product, increment. In other words, it means, if a product is already under development, then what functionalities we have already created and what makes the best sense to pick up next.

As an example, if we are building Cart functionality of a website, and in Sprint 1, we could complete only half of it, then it would make more sense to complete the rest of it before taking any other features. Examples of Sprint goal could be - "Create login and registration capabilities for customers so they can utilize email and social login features" Sprint goal should not be vague - E.g., Improve performance of the home page. Instead, we should have a clear goal like Improve the page load time of the home page by 10%.

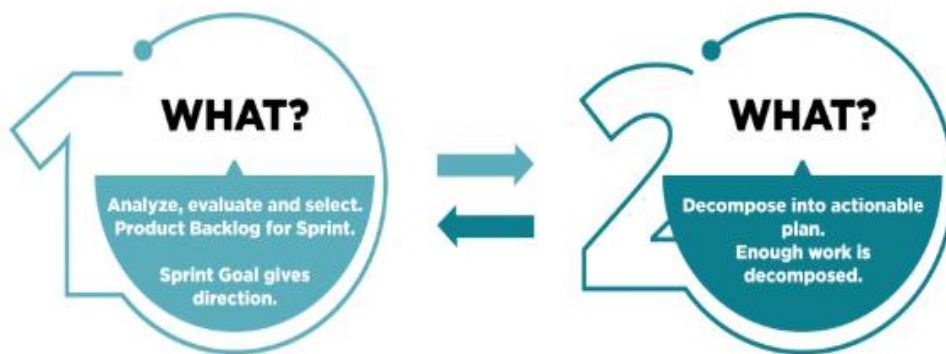


Fig. 6. Daily Stand-ups & Retrospectives – Facilitating team meetings for continuous improvement. From Atlassian

2) Story Selection -

Now that we know the Sprint goal and the team's alignment with it, the next step is to select the stories that correspond to the Sprint goal. One of the critical inputs to the Sprint planning meeting is - the product backlog. It has all the stories that need work, and the team selects the stories from this backlog. Product Owner helps in prioritizing this list. This, in turn, ensures that the correct alignment of the Story chosen to the Sprint goals. Once we select the stories for the Sprint, they are called Sprint Backlog. The

product owner helps in clarifying any doubts that the development team may have on the Story description.

3) *Capacity Planning* -

The sprint goal is defined, and we have got the Sprint stories from the product backlog. Can we do it all in the Sprint? It depends on Sprint Capacity. In simple terms, the stories taken in the Sprint will require some effort - Say 100 Days. So we need to find out whether we have 100 days' worth of effort is available in the team.

Let's assume the team is working on a two weeks Sprint. At the outset, we know that we don't have ten days available. We need to commit some time for Sprint planning, Sprint Review, and Sprint Retrospective meetings. Apart from that, the team might need to reserve some time for supporting production activities or any other work beyond current Sprint goals. We also need to consider that not all 8 hours are available with the team. They would need to attend some organization meetings and participate in account-level activities. Also, we should consider that the team members may plan personal time off.

There are so many dynamics involved daily - Who keeps track of all this? It is the job of the Scrum Master, who has the primary ownership of capacity planning.

4) *Backlog Management*

Product Backlog Management is the act of adjusting and ordering items on the Product Backlog so that the Scrum Team can deliver the most valuable product possible. A Product Owner is accountable for managing the Product Backlog. They do so as often as they see fit and can also delegate the responsibility to others. However, the Product Owner ultimately decides on what work to pursue now, later or not at all.

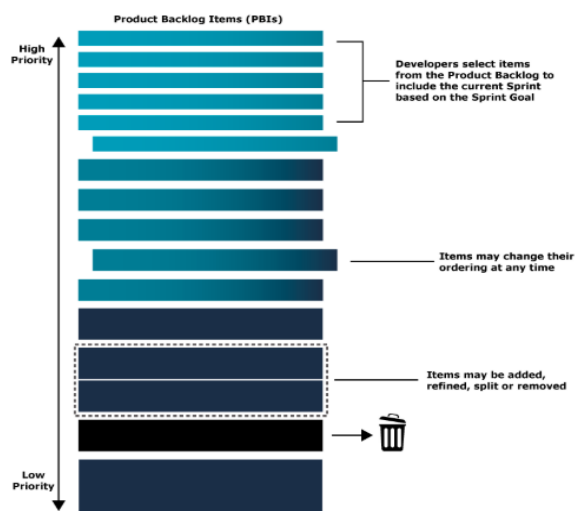


Fig. 7. Scaling Scrum – Applying Scrum in large organizations and multi-team environments from Atlassian

V. AGILE ESTIMATION & RISK MANAGEMENT

Techniques for estimating work and managing risks.

Agile estimation estimates effort to complete a prioritized task in the product backlog. We measure it for the time it would take to complete that task. As a result, you can plan sprints more accurately.

Agile teams also make estimations concerning story points. A story point is used in Agile Software Development projects to estimate the difficulty of implementing a given user story. We measure it in relative units assigned to different user stories that require estimation.

A story point is a number that helps estimate the difficulty of building a user story successfully. This difficulty could be related to the complexities, risks, and efforts involved.

Agile project estimation also helps to build strong coordination. If project X depends on project Y, agile project estimation provides an overview of the wait time.

1) Why run agile estimation

- Making teams accountable for deliverables
- Inducing discipline across the Agile team
- Predicting the approximate time, it will take to finish a project
- Enabling better sprint management
- Improving team productivity

2) Why do Teams Estimate in Agile?

Overestimating and underestimating are both typical for Agile software development companies. It leads to varying development and launch times. Considering Agile estimation in the initial stages can assist in accurate user story estimations. It helps the team stick to the deliverables, and you don't deflect.

3) Benefits of Agile Estimation

Some of the to-the-point benefits of Agile Estimation techniques include:

- Improved Decision-Making

With accurate, agile estimation, the development team can conduct practical backlog grooming sessions, which further helps in precise sprint planning. Their user story delivery time will improve when they make informed decisions and plan well.

- Better Coordination

Let's say the estimated effort for user story A is two weeks. On the other hand, the estimation effort for user story B is four weeks. Now, both user stories depend on each other and are connected. In that case, the team needs to prioritize work so that both user stories get completed simultaneously. It will lead to better coordination among teams.

- Better Risk Management

Software projects often suffer from exceeding budgets and timelines. To lessen this risk, Agile project estimation is an ideal solution. Agile product estimation helps estimate story points and stick to budgets, estimates, and the project's scope—the more accurate the estimates, the better the chances of on-time, quality delivery.

4) *Stages of Agile Estimation: The Short Discovery Phase*

When a project starts, the horizon is limited. Hence, it is wise to implement a short product discovery phase to tide over this problem. The discovery phase establishes the essential tenet of Agile development methodology, breaking down the requirements into small batch sizes. It is an exercise that typically takes two to four weeks, depending on the project's complexity.

VI. SERVANT LEADERSHIP

Coaching teams and fostering collaboration.

Servant leadership is characterized by leaders who put the needs of a group over their own. These leaders foster trust among employees by holding themselves accountable, helping others develop, showing appreciation, sharing power and listening without judging. While serving and leading seem like conflicting activities, these leaders are effective initiators of action.



Fig. 8. Role of servant leadership is to serve the team taken from Managedagile

“The servant-leader is servant first... It begins with the natural feeling that one wants to serve, to serve first.

The difference manifests itself in the care taken by the servant-first to make sure that other people's highest priority needs are being served. The best test, and difficult to administer, is: Do those served grow as persons?

A servant-leader focuses primarily on the growth and well-being of people and the communities to which they belong”

Conclusion

Agile and Scrum have revolutionized the way teams approach software development, project management, and product innovation. Agile methodology prioritizes flexibility, collaboration, and customer feedback, while Scrum offers a structured framework to manage iterative development effectively.

Teams that adopt Agile and Scrum benefit from faster delivery cycles, improved adaptability, enhanced teamwork, and higher customer satisfaction. Scrum's clear roles—Scrum Master, Product Owner, and Development Team—help drive ownership and efficiency, ensuring continuous improvement through sprint reviews and retrospectives.

Despite their advantages, Agile and Scrum require strong leadership, disciplined execution, and commitment to iterative learning. Organizations must foster a culture of agility, promoting openness to change and refining their processes with ongoing feedback loops.

Looking ahead, Agile and Scrum will continue evolving, integrating DevOps, AI-driven project management, and scaled Agile frameworks to enhance efficiency further. Their principles remain timeless, emphasizing collaboration, adaptability, and continuous delivery in dynamic environments.

REFERENCES

- [1] Scrum alliance, “Scrum Framework”, https://assets.scrumalliance.org/scrumredesigndevsite/media/ScrumAllianceMedia/WhitePaper_English_2019.pdf, 2019
- [2] PMI, “Agile Framework”, <https://www.pmi.org/learning/agile/>, last accessed Sep 2023
- [3] Asana, “Agile Methodology”, <https://asana.com/resources/agile-methodology>, last accessed Sep 2023
- [4] Atlassian, “Scrum roles”, <https://www.atlassian.com/agile/scrum/roles>, last accessed Sep 2023
- [5] Toolsqa, “Sprint planning”, https://toolsqa.com/agile/scrum/sprint-planning/#google_vignette, July 2021
- [6] Scrum, “Backlog management”, <https://www.scrum.org/learning-series/product-backlog-management/>, last accessed Oct 2023
- [7] Netsolutions, “Agile Estimation Techniques”, <https://www.netsolutions.com/insights/how-to-estimate-projects-in-agile/>, Dec 2022
- [8] Managedagile, “Servant leadership”, <https://managedagile.com/>, last accessed Nov 2023