Agile Software Development: Main Use Cases

According to the annual PMI’s Pulse of the Profession report, 48 percent of projects miss deadlines, 43 percent exceed budget, and 31 percent fail to meet the project's initial goals.

Modern project managers are obviously struggling to find a road to success, which is why an increasing number of them are implementing Agile methodology for their software development projects.

Agile is one of the simplest yet most effective methodologies for delivering high-quality software development services and solutions. When used correctly, Agile allows for efficient, flexible, and error-proof software development teams management. However, somewhere along the way, people begin to overcomplicate things.

Below are the most frequently asked questions about Agile methodology, answered by our key experts.

What Is Agile Software Development?

Agile methodology differs from the once-dominant Waterfall approach. Software development teams that use the Waterfall process create highly detailed specifications (SRS) and functionality requirements. After that, the software development process is divided into milestones and is conducted until a "finished" product is released.

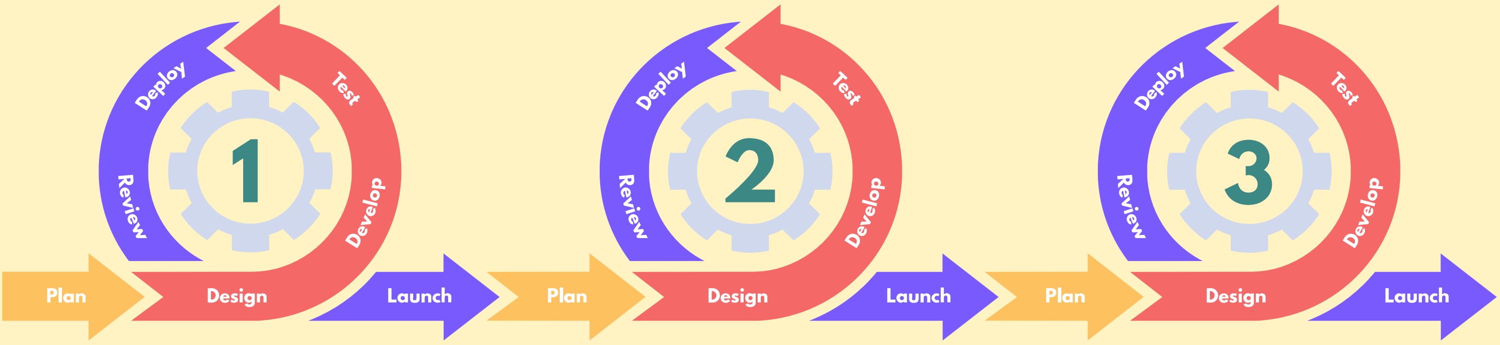
The more sophisticated the software is, the more difficult it is to create upfront specifications that cover all the details and foresee what features users will really need. As software is becoming more and more complicated over time, Agile software development methodology is becoming more prevailing than a Waterfall approach. With the constantly changing requirements, it is difficult to predict what features users might need in the end. Thus most businesses prefer to create an MVP version to test drive their solution and get user feedback at the very beginning. That’s where the Agile approach comes into place, being an iterative methodology that allows for developing software that focuses on delivering a minimal viable product (MVP) and then modifying and adding features in phases based on user feedback. This methodology helps to reduce the risk of wasting the budget and producing a digital product based on erroneous assumptions about the functionality that users need. Besides, cross-functional team collaboration is an often practice in the Agile methodology.

In a non-Agile methodology like Waterfall, there can be minimal collaboration across development teams. It's a much more linear process, similar to what you would find in a manufacturing production line, where the development team conceptualizes, designs, and develops comprehensive specifications for a product. The specifications are then forwarded to software developers and designers, who create their own pieces, which are then assembled. After that, the software is passed on to QA and testing teams, followed by an operations team that deploys the software as a live product.

Agile vs Waterfall: which project management methodology is better for software development?

Check out

Agile facilitates a disciplined project management process that enables regular checks and modifications, a leadership philosophy that encourages teamwork, and a business approach that aligns development with customer needs and company goals. In Agile development, there is constant back-and-forth interaction and frequent cross-over between teams. It is a cyclical process, where first the team creates an MVP, then revisions it and plans future changes.



What Is Agile Used For?

Many companies are utilizing Agile methodology to improve team performance, customer satisfaction and increase project flexibility. Companies that have adopted Agile are able to adjust more quickly to market changes and complete more projects effectively. Priorities and requirements can easily be adjusted throughout the project to meet the needs of the stakeholders.

Agile is a great approach to clear up any misconceptions and misunderstandings and speed up the development process. Agile projects place a greater emphasis on completing tasks rather than planning and documenting them. With each iteration or sprint, the team's efforts are focused on producing and delivering working software.

When the product concept or features aren't clearly defined, Agile is the right approach to use. It allows business owners to change requirements and objectives as the project progresses, allowing them to leverage opportunities and ultimately deliver a better product.

12 Agile Principles

Agile's goal is to align development with business needs, and it has proven to be successful for many complex projects.

Agile has evolved into an overarching philosophy of software development where clients are at the center of agile development, which encourages client feedback and participation.

1.Satisfy Customers Through Early & Continuous Delivery

The best way to ensure you make customers satisfied is to deliver the product early, iterate frequently, and listen to customers continually. Unlike traditional product development methodologies, which are known for their lengthy development cycles, agile principles are aimed at reducing the time between ideation and launch. The goal is to deliver a working product as quickly as possible to swiftly deploy an MVP and get feedback from customers. This information is then handed back to the development team to help shape future releases.

2.Welcome Changing Requirements Even Late in the Project

In a world that’s constantly evolving, nobody can really predict what customers’ needs will be in a year or even some months. Instead of skipping these changes and moving forward, agile principles encourage you to respond to them. Previously, product development was often changed adversely which means that the development team had to create elaborate, well-documented plans before the development process actually began and set them in stone regardless of changes. Agile principles encourage keeping pace with ever-changing markets and customer needs and adjusting the development process as needed.

3.Deliver Value Frequently

The Agile philosophy endorses cutting down the development of a product into smaller iterations and releasing them continuously. This means smaller releases, and smaller releases mean fewer chances for bugs to arise.

Agile's frequent-release approach also creates more opportunities for your team to validate your product ideas and strategies. More frequent releases also provide more opportunities for customers’ feedback. If you wait for a long time to get feedback on all of your improvements, you will have a lot more work to do to address any criticism.

4.Business Stakeholders and Developers Should Collaborate Throughout the Project

A successful product requires insights from both the client and the development team, which can only be achieved if they collaborate on a regular basis.

A major problem associated with traditional project management methodologies is that the project stakeholders are typically unaware of the project's development stages. To ensure ongoing feedback and a valuable end product, Agile principles urge all stakeholders to participate in all stages of the product development.

Early detection of misunderstandings and regular feedback from each other helps in producing successful outcomes.

5.Build Projects Around Motivated Individuals

Your project is unlikely to succeed if it is built around people who aren't motivated or are demotivated due to the lack of trust or support. In Agile, it is the project manager's role to provide a motivating environment where responsibilities are clearly defined prior to the start of a project and include the right specialists and skill sets to complete the task. Giving team members confidence and inspiring them will result in considerable improvements in their overall performance, which will eventually bring benefit to the project.

6.Favor Face-To-Face Conversation to Other Modes of Communication

Some forms of communication, especially e-mails, can turn what would otherwise be a 10-minute discussion into 30 minutes, causing people to avoid contacting remote staff to avoid the hassle. All of this slows down the developing process.

Again, there are occasions when circumstances force us to choose one form of communication over another. Agile's main goal is to encourage product managers and developers to communicate in real-time about product-related issues.

7.Working Software is the Primary Measure of Progress

Software development is a lengthy process so it is normal for companies to track the progress. According to the Agile approach, working software is the key way to measure progress.

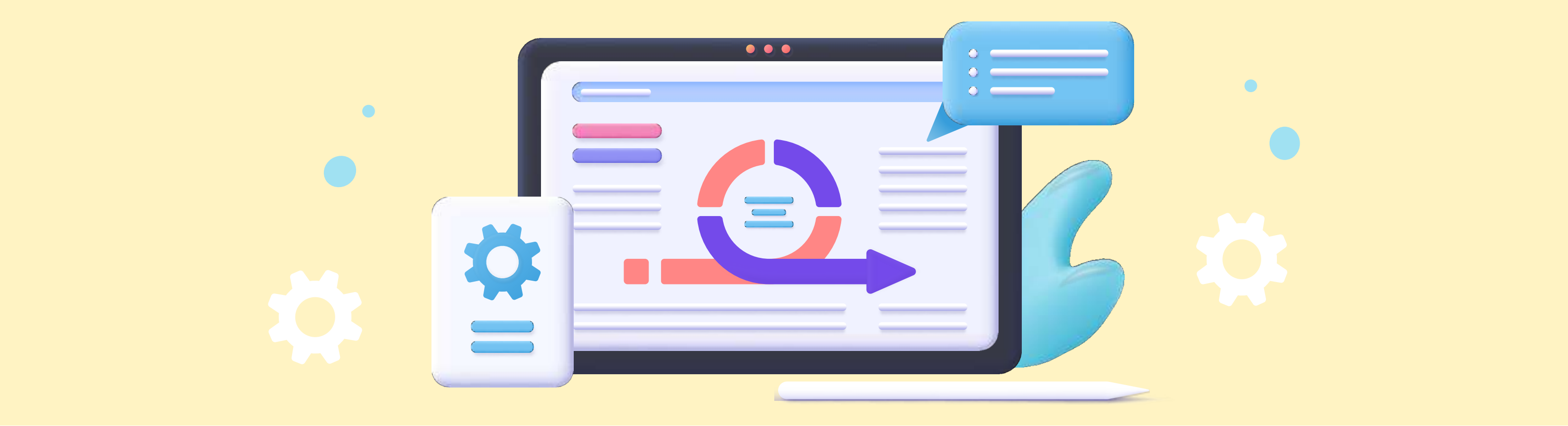
Working software comes first, followed by perfect mock-ups detailed documentation, etc. But if you haven't built at least a fraction of a working product, you haven't created value for your client.

A working product is the ultimate measure of success in Agile. As a result, this practice focuses on getting products to market as quickly as possible rather than having documentation become a bottleneck.

8.Maintain a Sustainable Working Pace

Keeping up with a rigorous, rapid-release schedule can be stressful for a team. Particularly if the bar is set too high.

Agile principles encourage product managers to be aware of this and to set realistic, measurable goals. The idea is to keep morale high and improve work-life balance among team members to avoid burnout. An iterative approach can keep the team members from feeling overwhelmed and overstressed, while the project continues to move forward at a steady pace.



9.Attention to Technical Detail and Design Enhances Agility

While the Agile methodology favors shorter development cycles and more frequent releases, it also emphasizes the need of keeping things neat and tidy to avoid future difficulties. Along with the development, there should be a focus on reducing technical problems and redundant code.
The development team should not wait to clean up the code but should do so on a regular basis.

10.Simplicity is Essential

Teams should also optimize the amount of work that can be done by eliminating procedures that are no longer relevant, automating manual tasks, and utilizing existing libraries rather than creating their own. It all saves us time and money, allowing teams to focus on providing more value to clients.

11.Self-Organizing Teams Generate Most Value

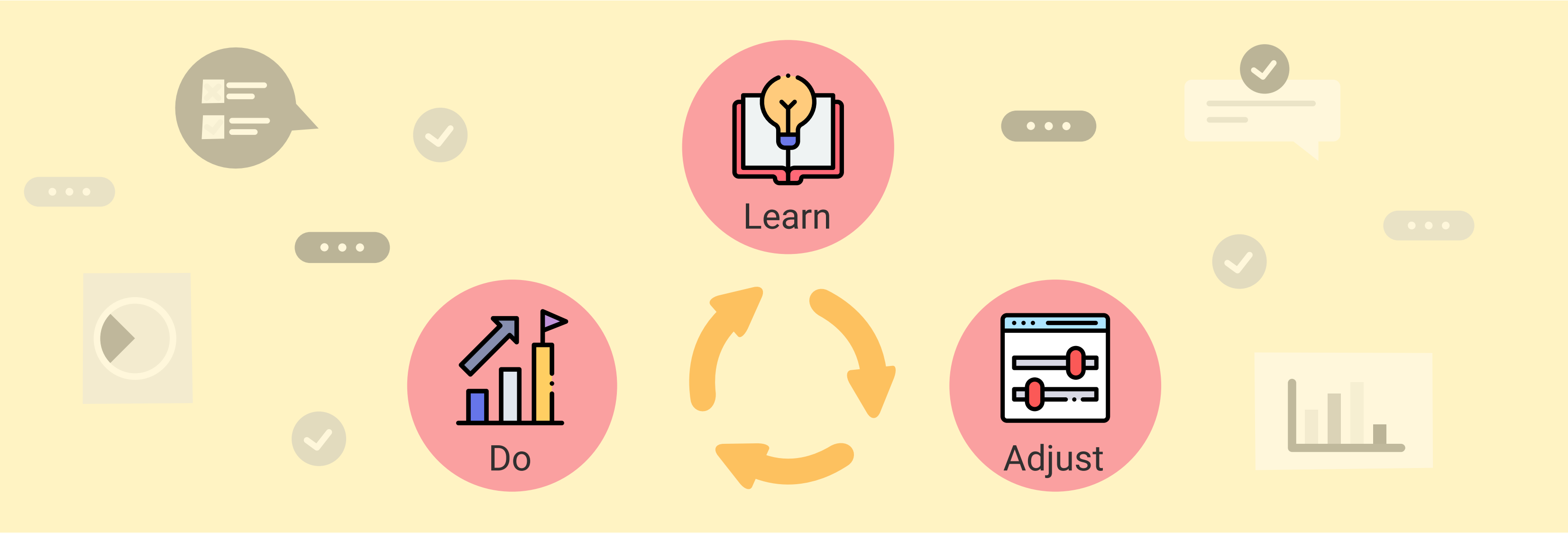
In traditional software development methodologies, key decisions are made by C-level executives.

Agile principles advocate for establishing self-organizing teams that follow a "flat" management style in which decisions are made collectively rather than by a single manager or management team. The concept is based on Agile's value of teams and interactions over processes and tools, with the goal of empowering teams to collaborate as needed. Still, developers are still considered employees who can be assigned tasks.

12.Regularly Reflect and Adjust Your Way of Work to Boost Effectiveness

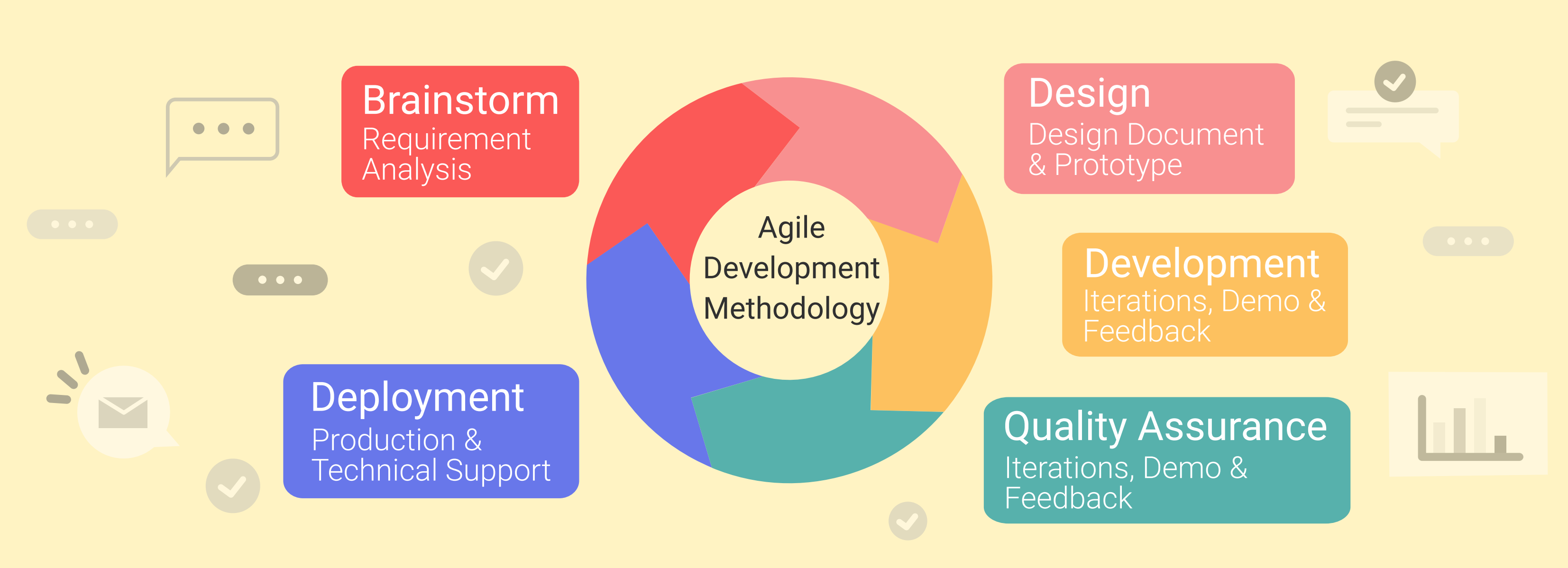
This principle will always be relevant. It's what makes teams and businesses successful by refusing to accept the status quo and continually seeking ways to improve things.

Agile methodologies are built on the concept of iteration, where teams learn from their mistakes and improve their performance. Project managers should encourage team members to reflect on their performance and find ways to enhance their technical and managerial skills.



Key Agile Software Development Phases

The Agile software development life cycle is a set of steps that a product goes through as it progresses from conception to completion: concept, inception, iteration, release, maintenance, and retirement.


Brainstorm
Requirement Analysis
Design
Design Document & Prototype
Agile Development Methodology
Development
Iterations, Demo & Feedback
Deployment
Production & Technical Support
Quality Assurance
Iterations, Demo & Feedback

Requirements

Stakeholders examine the whole project to determine the amount of time and resources needed to complete the development process. At the same time, the owner evaluates the risks and prioritizes the various functions based on their importance to the company.

Design

The business owner meets with the software development team and walks them through the requirements outlined in the first step. The team then determines the sequence of development steps and chooses the tech stack. During this stage, the software development team can prototype the desired user interface.

Development and Coding

After the client and the team have agreed on the plan, the team starts building the product. The product is delivered in stages, with each sprint aimed at improving the current version of the product. The initial release is likely to undergo many changes to provide maximum functionality. Each cycle includes testing, and the final product should undergo final testing.

Integration and Testing

At this stage, the team must run a number of tests to confirm that the product is fully functional. If any potential bugs or flaws are discovered, the developers will resolve problems right away. The development team also gathers user feedback at this phase.

Implementation and Deployment

Customers can now access the software because it has been fully deployed. During this stage, the software development team provides ongoing support to ensure that the software functions properly and doesn't have any bugs in it. Over time, the development team may need to update an existing product or add other functionality.

Retirement

During the retirement phase, the software release is removed from production. This happens either because the business owner wants to replace it with a newer version or because the software becomes redundant, obsolete, or incompatible with the company's current strategy.

SCAND Experience

SCAND team mostly uses Agile methodology (including Scrum and Kanban) for customer projects to provide predictability and full transparency of the software development progress.

More information on the benefits of using Scrum and how to use it is here.

Kanban is a less rigid methodology that focuses on a single board with all of the team's tasks. It's great when the project is already delivering new features on a regular basis. It's primarily useful for feature-oriented continuous delivery.

Daily meetings and regular demonstrations of the results keep the development process on track. We work with only estimated tasks and stories. We also actively manage the number of tasks/stories on the board for a clearer product roadmap and better team's cognitive relief. Besides, we also use a backlog of planned tasks and stories for "feeding" the main boards.

In addition, to be efficient we use the following principle when helping customers: working with expectations and consequences of all actions. To help with their upkeep, we maintain Project Management Office practices that allow for the development of key PM skills and the exchange of knowledge across all departments.

Conclusion

Software development is a fast-paced industry, so companies need to be flexible and proactive in all aspects of project development. Agile methodologies allow building cutting-edge solutions and cultivating innovative experience while maintaining products aligned with market trends and customer requirements.

However, there is always a place for diversity. The choice of the right software development methodology depends largely on your team size, goals, and other factors. Here is an overview of the most widely utilized software development methodologies to help you decide which is right for your team.

Feel free to contact us and ask any questions: info@scand.com