

Agile Project Management 101:

A Beginner's Guide

for Non-Project Managers



Coordinate Anything



- 01 Introduction
- 02 Agile overview
- 03 12 Agile Principles
- 05 Agile Development Cycle
- 06 Advantages & Disadvantages of Agile
- 08 Top Methodologies Used to Implement Agile
- 09 Top Methodologies Used to Implement Agile - Scrum
- 15 Top Methodologies Used to Implement Agile - Kanban
- 20 Other Agile Methodologies
- 21 How to Get Started With Agile
- 23 Finding the Best Agile Tool
- 25 About Smartsheet



Introduction

Everyone manages projects, whether they're a certified project manager or not. Often the people managing the work are simply 'winging it', which can result in a struggle to manage multiple projects, meet deadlines, and adapt to changing requirements.

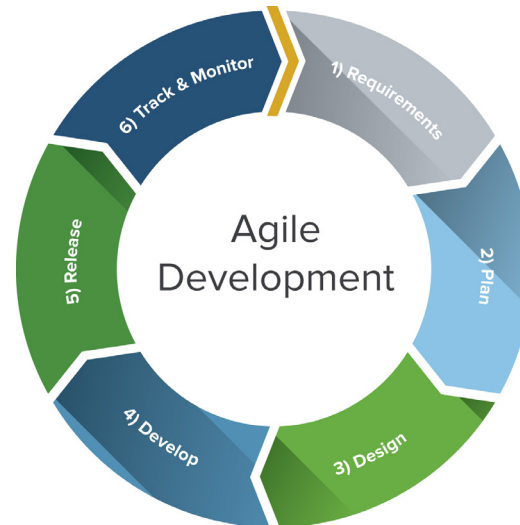
[Studies have found](#) that companies who use a standard project management methodology have had “fewer than half as many project failures than those that did not have one.” With this in mind, anyone who manages work should consider adopting a standard project management method. But [with all the PM methods](#) out there, how are you to know which one is best for your work management needs?

In this e-book, we'll take a look at the Agile project management philosophy. We'll give an overview of the top Agile methods, provide the advantages and disadvantages of each, and tell you how you can get started implementing Agile practices to ensure your next project is a success.

Agile Overview

Agile project management is based on an incremental, iterative approach. Instead of in-depth planning at the beginning of the project, Agile methodologies are open to changing requirements over time and encourages constant feedback from the end users. The goal of each iteration is to produce a working product.

Agile refers to any process that aligns with the concepts of the Agile Manifesto. In 2001, 17 software developers met to discuss lightweight development methods. They published the [Manifesto for Agile Software Development](#), which covered how they found “better ways of developing software by doing it and helping others do it.”



12 Agile Principles

The Agile Manifesto lists [12 principles](#) to guide teams on how to execute with agility:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.



12 Agile Principles

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity -- the art of maximizing the amount of work not done -- is essential
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



Agile Development Cycle

The phases in the Agile development cycle may not happen in succession; they are flexible and always evolving, with many happening in parallel.

Planning: Once an idea is deemed viable, the project team comes together to identify features, prioritize each feature, and assign them to an iteration.

Requirements analysis: Key stakeholders and users meet to identify business requirements that are quantifiable, relevant, and detailed.

Design: The design is prepared from the requirements identified and the team considers what the product or solution will look like, deciding on a test strategy or plan to proceed.

Implementation, coding or development: Coding or developing features, and scheduling iterations for deployment.

Testing: Test the code against the requirements to make sure the product is actually solving customer needs. This phase includes unit testing, integration testing, system testing, and acceptance testing.

Deployment: Deliver the product to customers. Once customers start using the product, they may run into new problems that the project team will need to address in future iterations.



Advantages of Agile

Agile evolved from different development approaches in the 1990s and is a response to some project managers' dislike of the rigid, linear Waterfall methodology. It focuses on flexibility, continuous improvement, and speed.

Here are some of the top advantages of Agile:

Change is embraced: With shorter planning cycles, there's always opportunity to refine and reprioritize the backlog to accommodate changes throughout the project.

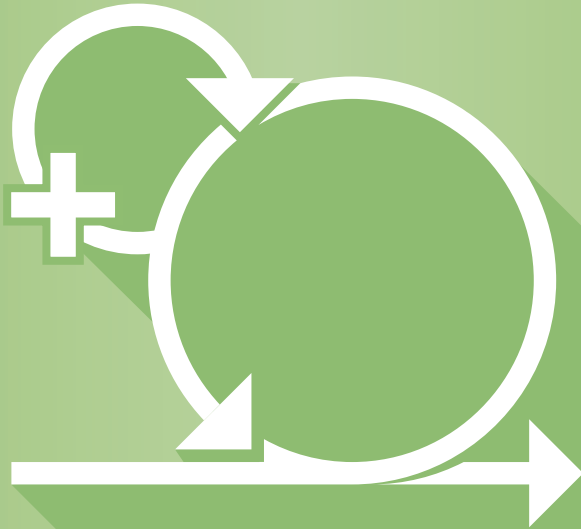
End-goal can be unknown: Agile is beneficial for projects where the end-goal is not clearly defined. As the project progresses, the goals will become evident and the team can adapt.

Faster, high-quality delivery: Breaking down the project into iterations allows the team to focus on high-quality development, testing, and collaboration. Conducting testing during each iteration means that bugs get identified and solved more quickly.

Strong team interaction: Agile embraces frequent communication and face-to-face interactions.

Customers are heard: Customers have many opportunities to see the work being delivered, share their input, and have an impact on the end product.

Continuous improvement: Feedback is encouraged from users and team members throughout the project, so lessons learned are used to improve future iterations.



Disadvantages of Agile

While flexibility in Agile is usually a positive, it also comes with some trade-offs. It can be hard to establish a solid delivery date, documentation can be neglected, or the final product can be very different than originally intended.

Here are some of the disadvantages of Agile:

Planning can be less concrete: Because project managers are often reprioritizing tasks, it's possible some items scheduled for delivery may not be complete in time. And, additional sprints may be added at any time in the project, adding to the overall timeline.

Team must be knowledgeable: Agile teams are usually small, so team members must be highly skilled in a variety of areas and understand Agile methodology.

Time commitment from developers: Active involvement and collaboration is required throughout the Agile process, which is more time consuming than a traditional approach.

Documentation can be neglected: Agile prefers working deliverables over comprehensive documentation. While documentation on its own does not lead to success, teams should find the right balance between documentation and discussion.





Top Methodologies Used to Implement Agile

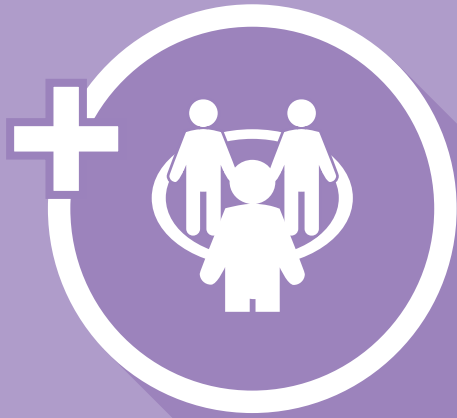
There are a number of specific methods within the Agile movement. We'll take an in-depth look at two of the top Agile Methodologies: Scrum and Kanban.



1 Scrum Methodology

Scrum is a subset of Agile and one of the most popular process frameworks for implementing Agile. It is an iterative development model often used to manage complex software and product development. Fixed-length iterations, called sprints lasting one to two weeks long, allow the team to ship software on a regular cadence. At the end of each sprint, stakeholders and team members meet to plan next steps.





Advantages of Scrum

Scrum is a highly prescriptive framework with specific roles and ceremonies. While it can be a lot to learn, these rules have a lot of advantages:

More transparency and project visibility: With daily stand-up meetings, the whole team knows who is doing what and issues are identified in advance, improving communication and enabling the team to take care of issues right away.

Increased team accountability: There is no project manager. Instead, the team collectively decides what work they can complete in each sprint, working together collaboratively, with accountability.

Easy to accommodate changes: With short sprints and constant feedback, it's easier to accommodate changes..

Increased cost savings: Constant communication ensures the team is aware of all issues and changes sooner, helping to lower expenses and increase quality.



Disadvantages of Scrum

While Scrum offers some concrete benefits, it also has some downsides. Scrum requires a high level of experience and commitment from the team and projects can be at risk of scope creep:

Risk of scope creep: Some Scrum projects can experience scope creep due to a lack of a specific end date, tempting stakeholders to keep requesting additional functionality.

Team requires experience and commitment: The team needs to be familiar with Scrum principles to succeed, as well as needs to commit to the daily meetings and stay on the team for the entire project.

The wrong Scrum Master can ruin everything: The Scrum Master is very different from a project manager. The Scrum Master does not have authority over the team, so he or she must trust the team to complete the work.

Poorly defined tasks can lead to inaccuracies: Project costs and timelines won't be accurate if tasks are not well defined. If the initial goals are unclear, planning becomes difficult and sprints can take more time than originally estimated.

Roles in Scrum

There are three specific roles in Scrum:

Product Owner: The Scrum Product Owner has the vision of what to build and conveys that to the team. He or she focuses on business and market requirements, prioritizing the work that needs to be done, managing the backlog, providing guidance on which features to ship next, and interacting with the team and other stakeholders to make sure everyone understands the items on the product backlog.

Scrum Master: Often considered the coach for the team, the Scrum Master helps the team do their best possible work. This means organizing meetings, dealing with roadblocks and challenges, and working with the Product Owner to ensure the product backlog is ready for the next sprint.

Scrum Team: The Scrum Team is comprised of five to seven members. Unlike traditional development teams, there are not distinct roles like programmer, designer, or tester. Everyone on the project completes the set of work together.



Steps in the Scrum Process

There are a specific, unchanging set of steps in the Scrum flow:

1. **Product backlog:** The product backlog is not a list of things to be completed, but rather it is a list of all the desired features for the product.
Sprint planning: Before each sprint, the Product Owner presents the top items on the backlog in a sprint planning meeting. The team determines the work they can complete during the sprint and moves the work from the product backlog to the sprint backlog.
2. **Backlog refinement/grooming:** At the end of each sprint, the team and Product Owner meet to make sure the backlog is ready for the next sprint. The team may remove user stories that aren't relevant, create new user stories, reassess the priority of stories, or split user stories into smaller tasks.
3. **Daily Scrum meetings:** The Daily Scrum is a 15-minute stand-up meeting that happens at the same time and place every day during the sprint. During the meeting each team member talks about what they worked on the day before, what they'll work on today, and any roadblocks.
4. **Sprint review meeting:** At the end of each sprint, the team presents the work they have completed as a live demo rather than a presentation.
Sprint retrospective meeting: Also at the end of each sprint, the team reflects on how well Scrum is working for them and talks about any changes that need to be made in the next sprint.





Tools and Artifacts in Scrum

In addition to roles and ceremonies, Scrum projects also include certain tools and “artifacts”. For example, the team uses a Scrum board to visualize the backlog or a burndown chart to show outstanding work. The most common artifacts and tools are:

Scrum board: The Scrum board helps to visualize your sprint backlog and traditionally involves index cards or Post-It notes on a whiteboard. The board is usually divided into three categories: to do, work in progress, and done. The team updates the board by moving tasks (written on cards) through the columns on the board.

User stories: A user story describes a software feature from the customer’s perspective. It includes the type of user, what they want, and why they want it.
Burndown chart: A burndown chart represents all outstanding work. The backlog is usually on the vertical axis, with time along the horizontal axis. A burndown chart can warn the team of potential risk and helps to show the impact of decisions.

Timeboxing: A timebox is a set period of time that a team works towards completing a goal. Instead of letting a team work until the goal is reached, the timebox approach stops work when the time limit is reached.

Icebox: Any user stories that are recorded but not moved to development are stored in the icebox.

2 Kanban Methodology

Kanban is Japanese for “visual sign” or “card.” It is a visual framework used to implement Agile and shows what to produce, when to produce it, and how much to produce. It encourages small, incremental changes to your current system and does not require a certain set up or procedure (meaning, you could overlay Kanban on top of other existing workflows).

Kanban was inspired by the Toyota Production System and Lean Manufacturing. In the 1940s, Toyota improved its engineering process by modeling it after how supermarkets stock shelves. Engineer Taiichi Ohno noticed that supermarkets stock just enough product to meet demand. Inventory would only be restocked when there was empty space on the shelf (a visual cue).

These same ideas apply to software teams and IT projects today. In this context, development work-in-progress (WIP) takes the place of inventory, and new work can only be added when there is an “empty space” on the team’s visual Kanban board. Kanban matches the amount of WIP to the team’s capacity, improving flexibility, transparency, and output.





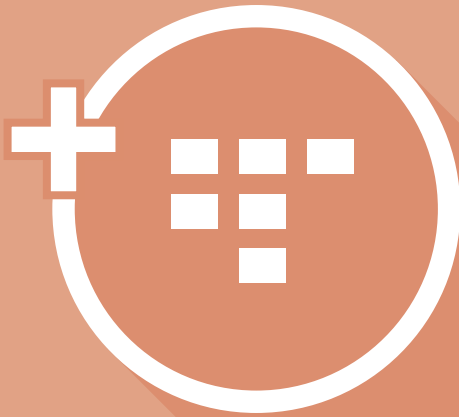
About the Kanban Board

A Kanban board is a tool to implement the Kanban method for projects. Traditionally, this tool has been a physical board, with magnets, plastic chips, or sticky notes on a whiteboard. However, in recent years, more and more project management software tools have created online Kanban boards.

A Kanban board, whether it is physical or online, is made up of different swim lanes or columns. The simplest boards have three columns: to do, in progress, and done. Other projects may consist of backlog, ready, coding, testing, approval, and done columns.

Kanban cards (like sticky notes) represent the work and each card is placed on the board in the lane that represents the status of that work. These cards communicate status at a glance. You could also use different color cards to represent different details. For example, green cards could represent a feature and orange cards could represent a task.





Advantages of Kanban

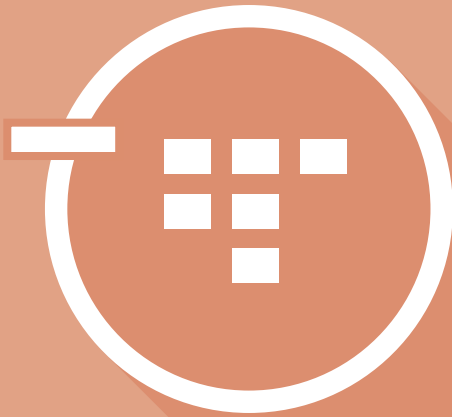
Kanban's visual nature offers a unique advantage when implementing Agile. The Kanban board is easy to learn and understand, it improves flow of work, and minimizes cycle time:

Increases flexibility: Kanban is an evolving, fluid model. There are no set phase durations and priorities are reevaluated with new information.
Reduces waste: Kanban revolves around reducing waste, ensuring that teams don't spend time doing work that isn't needed or doing the wrong kind of work.

Easy to understand: The visual nature of Kanban helps to make it intuitive and easy to learn.

Improves delivery flow: Kanban focuses on the just-in-time delivery of value and delivering work to customers on a regular cadence.

Minimizes cycle time: Cycle time is the amount of time it takes for work to move through the team's workflow. In Kanban projects, the entire team helps to ensure the work is moving quickly and successfully through the process.



Disadvantages of Kanban

Many of the disadvantages associated with Kanban come with misuse or mishandling of the Kanban board. An outdated or overcomplicated board can lead to confusion, inaccuracies, or miscommunication:

Outdated board can lead to issues: The team must be committed to keeping the Kanban board up to date, otherwise they'll be working off inaccurate information.

Teams can overcomplicate the board: The Kanban board should remain clear and easy to read. Adding bells and whistles to the Kanban board just buries the important information.

Lack of timing: The columns on the Kanban board are marked by phase, with no timeframes associated.

Core Practices and Principles of Kanban

Every Kanban project should follow these core principles:

Visualize the workflow: A visual representation of your work allows you to understand the big picture and see how the flow of work progresses. By making all the work visible you can identify issues early on and improve collaboration.

Limit work in progress (WIP): Work in progress limits determine the minimum and maximum amount of work for each column on the board or for each workflow. By putting a limit on WIP, you can increase speed and flexibility, and reduce the need for prioritizing tasks.

Manage and enhance the flow: The flow of work throughout the Kanban board should be monitored for possible improvements. A fast, smooth flow shows the team is creating value quickly.

Make process policies explicit: Everyone needs to understand how things work or what qualifies as “done”. Modify the board to make these processes more clear.

Continuously improve: The Kanban method encourages small, continuous changes that stick. Once the Kanban system is in place, the team will be able to identify and understand issues and suggest improvements.





Other Agile Methodologies

Extreme Programming (XP): This type of software development is intended to improve quality and responsiveness to evolving customer requirements.

Feature-driven development (FDD): There are five basic activities in FDD: develop overall model, build feature list, plan by feature, design by feature, and build by feature.

Adaptive system development (ASD): ASD represents the idea that projects should always be in a state of continuous adaptation, and has a cycle of three repeating series: speculate, collaborate, and learn.

Dynamic Systems Development Method (DSDM): DSDM addresses the common failures of IT projects, like going over budget, missing deadlines, and lack of user involvement. The eight principles of DSDM are: focus on the business need, deliver on time, collaborate, never compromise quality, build incrementally from firm foundations, develop iteratively, communicate continuously and clearly, and demonstrate control.

Lean Software Development (LSD): LSD can be characterized by seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity in, and see the whole.

Crystal Clear: This methodology can be used with teams of six to eight developers and it focuses on the people, not processes or artifacts. Crystal Clear requires the following: frequent delivery of usable code to users, reflective improvement, and osmotic communication preferably by being co-located.



How to Get Started with Agile

Getting started with Agile can be easy. We'll take a look at a few ways to implement Agile practices and how to pick the right Agile tool.

3 Ways to Implement Agile Practices

Whichever route you choose, remember that Agile is flexible in its very nature. There is no wrong or right way to get started with Agile.

Stand up Meetings: A simple way to get started with Agile is to incorporate daily stand-up meetings into your project. Daily stand-up meetings are easy to incorporate into any project methodology you already use (even Waterfall) and don't require any training or knowledge transfer.

Kanban Board: Another way to incorporate Agile practices is to create and use a Kanban board. The Kanban board is a simple tool to help your team visualize the flow of work as it's getting done. Use the board during stand-up meetings to discuss current work in progress or display it where your team can easily access it to make updates to task status.

Changing Team Roles: Some methods of Agile may result in the need to change team roles. For example, working with Scrum, the team may need to take more responsibility and boost speed of delivery. A good place to start with Scrum is to talk about the roles and responsibilities. Every project must have a Scrum Master, Product Owner, and Scrum Team. Clarifying these roles will help teams understand their responsibilities and remain accountable.



Finding the Best Agile Tool

If you're considering a switch to Agile project management, you should determine which tool is best to help track and manage your projects. Here are the top things to consider when choosing a tool:

Familiar and Easy to Use: When switching to a new tool, familiarity is key. You don't want your team spending valuable time learning a new program. Agile project management software should be flexible and intuitive to use.

Collaboration and Communication: Find a tool that facilitates collaboration between internal and external stakeholders. Cloud tools enable people to work in real-time, view and edit projects from anywhere, stay up to date on current status, and communicate with others viewing the project.

Searchable, Central Storage: It's increasingly important to use cloud storage services like Dropbox, Google Drive, Box, or OneDrive for project documentation storage. Whatever service you use, make sure your Agile tool can seamlessly connect with them to store vital assets.

Mobile Ready: The modern worker relies on their smartphone, tablet and other mobile devices to get work done. And many workers want to access parts of their project management software while they are on the go. Be sure to evaluate all the mobile options a tool has to offer.

Work Visualization: Today, work design is infinitely different. The "one-size-fits-all" approach to project management no longer works. Find a tool that includes multiple ways to visualize and manage your work - whether it be waterfall or agile - with Gantt charts, spreadsheet, kanban boards, calendars, reports, and dashboards.

Wrap Up: A More Agile Way to Manage Work

With fewer than one-third of all projects completed on time and on budget, it can sometimes seem impossible to lead new projects to success.

That's why finding the best project management method is important to the success of your projects. Whether you decide to go with Agile, Scrum, Kanban, or even a hybrid of multiple methods, deciding on a standard way to manage projects will help to streamline the process, increase team output, and keep projects on track.

Plus, once you've picked the best method for your team, finding the right agile project management tool can help implement and track the process and ensure no detail is missed.



About Smartsheet

Smartsheet is changing the way that individuals, teams, and organizations deliver great work. Our collaborative work management platform is trusted by over over 8 million registered users in 80,000 customers across more than 190 countries. Smartsheet is used by organizations across all major industries including technology, construction, financial services, education, and more.

By providing a broad range of smart views – Grid, Calendar, Gantt, Sights – Smartsheet works the way you want. Our newest view, Card View, gives teams a more highly-visual way to work, communicate, and collaborate in Smartsheet. Plus, with a similar feel as excel, Smartsheet is easy to start using right away, yet it's cloud-based so you can access your data and information anywhere, anytime.

Want to know how Smartsheet can help individuals and teams implement Agile practices? [Click here](#) to see firsthand how our new Card View can help manage your projects in a more agile way.

