

AGILE PROJECT MANAGEMENT

What is Agile project management?

- Agile project management is an iterative approach to project management that focuses on breaking down large projects into more manageable tasks, which are completed in short iterations throughout the project life cycle.
- Teams that adopt the Agile methodology are able to complete work faster, adapt to changing project requirements, and optimize their workflow.
- As the name suggests, Agile allows teams to be better equipped to quickly change direction and focus. Software companies and marketing agencies are especially aware of the tendency for changes from project stakeholders to happen from week to week.
- The Agile methodology allows teams to re-evaluate the work they are doing and adjust in given increments to make sure that as the work and customer landscape change, the focus also changes for the team.

If you're new to Agile project management, it might look at first like a complex and difficult-to-manage system. But, whether you realize it or not, you're already doing many of the things Agile requires. With a few tweaks, you'll be on your way to shorter development cycles and smaller, more frequent product releases.

What are the 4 core values of Agile?

The Agile Manifesto outlines 4 Core Values and 12 Guiding Principles which serve as a North Star for any team adopting an Agile methodology.

The 4 Core Values of Agile are:

1. Individuals and interactions over processes and tools

As sophisticated as technology gets, the human element will always serve as an important role in any kind of project management. Relying too heavily on processes and tools results in an inability to adapt to changing circumstances.

2. Working software over comprehensive documentation

As important as documentation is, working software is more. This value is all about giving the developers exactly what they need to get the job done, without overloading them.

3. Customer collaboration over contract negotiation

Your customers are one of your most powerful assets. Whether internal or external customers, involving them throughout the process can help to ensure that the end product meets their needs more effectively.

4. Responding to change by following a plan

This value is one of the biggest departures from traditional project management. Historically, change was seen as an expense, and one to be avoided. Agile allows for continuous change throughout the life of any given project. Each sprint provides an opportunity for review and course correction.

What are the 12 principles of Agile?

Agile methodologies can be as diverse and unique as each individual team, but the 12 Principles of Agile should always guide your decisions and product development.

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software (or whatever else you deliver).
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver projects frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
4. Coordinating team members 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. Face-to-face conversation is the most efficient and effective method of conveying information to and within different teams.
7. The final product is the primary measure of progress.
8. Agile processes promote sustainable development. All stakeholders 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.

Key components of Agile project management

A. User stories

Put simply, a user story is a high-level definition of a work request. It contains just enough information so the team can produce a reasonable estimate of the effort required to accomplish the request. This short, simple description is written from the user's perspective and focuses on outlining what your client wants (their goals) and why.

Why create user stories?

Breaking down initiatives and epics {An epic is a high-level requirement, task, or feature set that teams can break down into smaller user stories.} into user stories helps keep developers focused on bite-sized units of work. Each time a team finishes a user story, they can celebrate their accomplishment, which drives momentum and motivation.

User stories are also helpful because they focus explicitly on the customer. With the end-user in mind, teams can collaborate in creative ways to solve real-world problems that affect their clients.

Who creates user stories?

While anyone can write a user story, the product owner usually plays a critical role in ensuring user stories support each epic and its larger initiative. Everyone on the team should be encouraged to suggest user stories that might help drive product development forward.

How to write user stories:

While many teams hold a user story writing session at the beginning of Agile development, product owners and other team members often add new user stories over the course of working on a project. Iterating, adapting, and learning from mistakes is part of the Agile process.

When writing user stories, consider the following questions:

- Who are we serving?
- What does “finish” mean?
- What steps do we need to take to accomplish our goal?
- Whose feedback do we need to incorporate?
- Do we have any hard deadlines?

Your team can use the following user story format to help guide the drafting process: “As a _____, I want to _____ so that _____.”

Keep development focused on user stories

User stories are an excellent way to keep the customer in mind during product development. Project managers should encourage team members to suggest user stories throughout the development process to support epics, initiatives, and the overall vision for the product.

B. Sprints

Sprints: [A sprint in Scrum is a short period of time wherein a development team works to complete specific tasks, milestones, or deliverables. Sprints, also referred to as “iterations,” essentially break the project schedule into digestible blocks of time in which smaller goals can be accomplished] are short iterations, usually between one to three weeks to complete, where teams work on tasks determined in the sprint planning meeting. As you move forward, the idea is to continuously repeat these sprints until your product is feature ready. Once the sprint is over, you review the product see what is and isn't working, make adjustments, and begin another sprint to improve the product or service.

Sprints vs. Scrums

Sometimes, the terms “sprint” and “Scrum” get confused, but they describe different things. Sprints refer to short, repeating blocks of time in which key parts of the project are completed. Scrum, on the other hand, is the name of an Agile project management methodology that uses set processes and protocols, including sprints, to enhance collaboration and continuously improve upon problems. Sprints are often considered the heart of any Scrum approach, as they allow projects to be broken into manageable chunks.

Agile sprint best practices

Now that you understand how an Agile sprint works, here are a few best practices to keep in mind:

- **Check on your tasks or sprint assignments before meetings**—Before each meeting or daily stand-up, be ready to share the status of your tasks as well as any concerns, barriers, or red flags.
- **Don't handle sprint planning alone**—Even seasoned project managers need help estimating how much time tasks will take (Story Points) and which steps should be tackled first. Sprint planning meetings should be a team effort or, at minimum, should allow contributors to sign off on sprint tasks and point out any potential obstacles.
- **Use data to improve**—During your sprint review and retrospective, consult data from your work management platform to help inform future sprint decisions. Perhaps you learned that having two team members collaborate before completing a task saves time and creates a better result. Maybe you've learned that certain task estimates were much too low. Take advantage of the data you have to make better decisions for the next sprint.

Agile project management is all about adapting and updating your project plan as needed, based on new information and insights learned within the project's sprints.

C. Stand-up meetings

A daily stand-up meeting is an opportunity for the project team to discuss a project's progress at a high level. These meetings last 15 minutes and allow each contributor to report on their accomplishments since the last stand-up meeting.

True to its name, all participants in stand-ups usually remain standing to keep the meetings short and on-topic. However, digital stand-ups are also possible. Making sure there's a repeatable agenda is the best way to keep either format of a daily stand-up meeting running smoothly.

In Agile project management, daily stand-up meetings are essential. These meetings allow project members to share critical information, openly discuss issues, and hold themselves and each other accountable. The collaboration and transparent level of communication can also lead to improved team dynamics, which can make completing a project together more productive and enjoyable for all.

Types of Agile stand-up meetings

Depending on whether you're using the Scrum or Kanban approach for Agile, your daily stand-up meetings will look a little different. Here's what to expect in each type of Agile stand-up meeting:

Scrum stand-ups

Scrum stand-ups are 15-minute daily meetings that occur each day of the sprint to discuss progress and quickly note any issues. Each contributor answers three questions:

- What did you accomplish since the last meeting?
- What are you working on before the next meeting?
- What is getting in your way or preventing you from doing your job?

These three questions hold team members accountable to their tasks and commitments by encouraging small, achievable goals that are shared with the team. They also allow team members to discuss problems, challenges, and setbacks openly, so no one is left in the dark.

Kanban stand-ups

If you're using the Kanban approach to project management, the focus of daily stand-up meetings will be slightly different. First, you'll want to pull up the Kanban visual workflow so that the project manager or lead can identify any bottlenecks.

Kanban focuses on eliminating bottlenecks before or as they pop up, so figuring out where problems may be occurring or where capacity challenges might begin to appear is crucial in these meetings.

There are no three questions to answer in the Kanban-style stand-up meeting. Instead, these daily stand-ups should focus on identifying issues and solving them, rather than discussing what everyone is working on. The visual board allows everyone to quickly see where tasks are, sparing team members from needing to detail where they are in the project flow.

Benefits of daily Agile stand-ups

No matter which approaches you use, daily stand-ups allow team members to work collaboratively toward project goals. While Scrum stand-ups focus on completing the sprint's goals, Kanban stand-up meetings work toward correcting bottlenecks before they slow down production.

Daily stand-up meetings are important for keeping Agile teams focused and on-task while providing quick, project-level updates to the rest of the team. These daily check-ins hold all team members accountable for their part in a project by looking at their current workflow and monitoring where they are on certain tasks or items. Because the Agile methodology is all about versatility and flexibility, it's important to make tweaks and improvements to your meetings to fit your team's needs. Your daily stand-up should inform and draw out issues so that you can keep your project on track and get ahead of issues before they pop up.

Common stand-up mistakes

While stand-up meetings can be time-saving meetings, it's easy for them to spin out of control if not managed properly. Here are some common mistakes that cause stand-up meetings to veer off course.

1. Not following the agenda

Since these daily meetings only last 15 minutes, staying on schedule is crucial. For instance, in Scrum stand-ups, team members' answers to the three questions should be quick and concise. If an issue requires further discussion, it should be added to a list (sometimes called a parking lot list) to be discussed in more detail later. Your parking lot list can be written on a whiteboard, typed into a digital document, or added to your work management software, depending on the platform you use. Allowing team members to add items to this parking lot list outside of the daily stand-up can ensure your meeting progresses as planned.

Team members don't need to describe in detail every task they're accomplishing. Instead of going in-depth about each of the 25 webpage templates the developer is working on, he or she can note that the first half of the website templates will be built before the next meeting. This puts the task in context and highlights high-level progress that all participants can easily understand.

Likewise, Kanban stand-up meetings should only address issues or challenges and not focus on what each team member is working on. The visual Kanban board allows team members to quickly see where tasks are and who is working on them, so adding this repetitive step can pull valuable time away from investigating challenges.

2. Not identifying repeating issues

If one team member has the same status update or mentions the same bottleneck every meeting, you have some following up to do.

For instance, in a Scrum stand-up meeting, if your web copywriter reports that they spent yesterday researching, will spend today researching, and has no issues to report, but seems to be making little progress, you may want to meet with them separately after the stand-up. Find out what's being researched, how much more research is left, and if any hurdles are preventing them from moving on to the copywriting phase.

As the project manager or Scrum Master, it's your job to ensure these meetings serve their intended purpose, so addressing any issues as soon as they're spotted will be important.

3. Not attending meetings daily

It's important to hold your daily stand-up meetings at the same time each day, within reason. If there's one day a week where it's impossible to meet at the same time, you can accommodate, but consistency is key to this type of meeting.

Since stand-up meetings are so quick, team members may think it's OK to skip them and provide their updates or issues ahead of time. Allowing team members to miss these meetings and email you their tasks and challenges could cause others to start skipping out as well, if they feel they're no longer mandatory. This can lead to some team members missing valuable project updates and being left out of the loop on changes or challenges.

To avoid this, require daily attendance from everyone. For anyone who works remotely or is traveling, set up a conference line, so they can present their updates wherever they are. Since this is only a 15-minute meeting, everyone should be able to make time for it, even if they're on the road.

4. Not establishing whose turn it is to speak

The silence between team member updates can eat up time and cause meetings to feel slightly awkward and disorganized. If you're meeting in person, you can have team members pass a ball or other object around the table until everyone with tasks has a chance to speak.

5. Not showing your task board

It's important to display the task board for that day, week, or sprint—whether it's on a whiteboard or in your work management system—during the entire meeting. In Scrum meetings, this gives team members the chance to speak up if user stories (also known as tasks) or story points (also known as hours) have been over/underestimated and really focus on the goals for that sprint. In Kanban, the visual flow is important when identifying bottlenecks.

Projecting or screen sharing your project management tool during this meeting will also allow others who are not reporting to better understand the current state of the project and see which tasks are being worked on when.

D. Agile board

An Agile board helps your team track the progress of your project. This can be a whiteboard with sticky notes, a simple Kanban board {What is a Kanban board?

A Kanban board is a visualization tool used in Agile methodologies to document workflows and processes within an Agile team or department. Primarily a function of the Kanban methodology, other variations of Agile boards are used in other project management methodologies like Scrum and Scrumban, though with slightly different elements.

Common elements of a Kanban board include cards, columns, and work in progress (WIP) limits which are used to optimize the flow of work.

Cards

Kanban cards at a basic level are a single item of work that needs to be completed. As work progresses on the task, the card moves across the Kanban board until complete. Kanban cards are key to visualizing the work in progress, work in the backlog, and work that has been completed.

Columns

The columns of a Kanban board represent the different stages of the workflow process. On a simple Kanban board, columns can say "To Do," "In Progress," and "Complete."

WIP limits

Work in progress limits are a key function of the Kanban methodology and are represented on a Kanban board. WIP limits represent the maximum amount of cards allowed in a column. If a WIP limit is hit, no more cards can be added to the column until another card is moved out of the column. This is key in visualizing bottlenecks in a workflow, **or a function within your project management software.**}

Why visualize your workflow?

Workflow visualization isn't just an item on the Agile checklist that teams to complete so they can say that they're Agile. It has enormous benefits, both inside and outside of the Agile team actually creating the Kanban board.

Transparency for the team

First of all, teams gain instant transparency when you visualize your work out onto a Kanban board. People outside of the team can immediately see exactly what its members are doing. That means when they come to the team with an emergency item, they understand what work they would be disrupting by demanding that the team pick it up right away.

Departments should also have far fewer “what are they even working on?” conversations once the team’s workflow is out in the open.

One word of caution about this extreme transparency: the harsh light it shines on output can make some team members uncomfortable. If someone feels they aren’t pulling their weight, they’re likely to be resistant to having their contributions revealed to the world.

Shared understanding within the team

Once everybody’s work is documented and visualized on a Kanban board, it’s very easy to see exactly where the team is devoting its resources. Project managers get insight into how much work is really being done (it’s usually more than you think), and whether it’s actually the right work at the right time.

Maybe everyone is focusing their time on repetitive tasks and never getting to the larger strategic initiatives that will ultimately deliver big results. Sometimes only an accurate Kanban board can reveal this kind of discrepancy.

Identifying and removing bottlenecks

Kanban boards also highlight where work is slowing down through a bottleneck. For example, a team might discover that work is getting stuck with copywriters and slowing down the flow. Armed with this empirical data, you can make a strong case for more writing resources on the team. An accurate workflow visualization takes the conversation from, “we really need more writers” to “we have 65 percent of our projects stalled because we’re short on writing help.” Using a cumulative flow diagram can help identify bottlenecks and improve cycle time and throughput.

At the end of the day you can’t go faster than your bottlenecks will allow, so removing them one by one is key to increasing the team’s effectiveness.

How to create a Kanban board

There are two ways you can create a Kanban board workflow visualization: a physical Kanban board or an online tool with a digital Kanban board feature. Both boards have their pros and cons, so let’s take a look at them one by one.

Option 1: Physical Kanban board

Agile teams have long relied on whiteboards, markers, and sticky notes to track their work, and these tools of the trade are highly valuable.

Pros: Assuming you have the requisite wall space, physical Kanban boards are easy to create. A few strokes of the marker, a couple dozen sticky notes, and you’re ready to go.

It’s also highly satisfying to physically move Kanban cards from one state of work to another, and seeing them stack up in the “done” column is a great feeling.

Cons: A physical Kanban board won't provide you with metrics to track the team's performance unless you're willing to do some manual math. It can also pose a problem for [remote team members](#) who can't update it.

Finally, if external teams need to suggest work for your backlog, a physical Kanban board can make that a cumbersome process.

Option 2: Digital Kanban board

Using a [digital work management tool](#) like Workfront is another solid option for visualizing the workflow, and it instantly takes care of the remote worker and metrics problems cited above. But a digital Kanban board may not be your silver bullet either.

Pros: If you get the right tool, digital Kanban boards automatically deliver metrics around your team's work, giving you hard data about what's going on.

You're also freed from the constraints of a wall, meaning you can add lots of information to a digital card that wouldn't fit on a physical one, like attached documents, extensive checklists, and links to related cards.

Finally, a digital board can simplify the request process so people outside the team can easily add work to the backlog.

Cons: It can be overwhelming to start with a digital visualization tool.

You may not want to take the time to onboard the team into a new system while they're transitioning to Agile. You may also find that people outside the team don't log in to check it, whereas it's hard for them to miss a big whiteboard right outside the team's workspace.

Ideal Option: Physical and digital Kanban boards

If you can, create a basic physical Kanban board near the team and update it every day during the [daily stand-up meeting](#). People teleworking can have a card buddy in the office to move their cards on their behalf.

By adding numbers to each physical card that refer back to their corresponding digital card, you get the power of the physical board with the tracking benefits of a digital tool. Your physical cards can be simple and to the point, while the cards in the digital system can hold more detail.

You can avoid the overhead of updating work in two tools by consolidating responsibility for the digital tool. For example, the team leaders can update the physical Kanban board during stand-up so that it's accurate, while individual contributors record their work in the digital tool.

Kanban board examples

Now that we've looked at some abstract ways to track work, here are three Kanban board examples. Whether you start with one of these examples or do something totally different, make sure your Kanban board accurately reflects how work gets done on your team, not how you wish it got done or how your manager thinks it happens.

Kanban board 1: Nice and simple

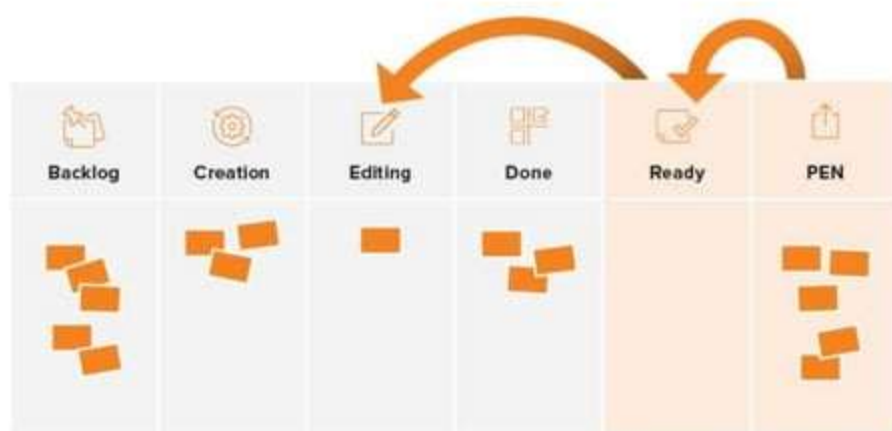
This simple Kanban board example is a great place to start. There's nothing fancy here, just different states that work is likely to enter on its way through the team.



Kanban board 2: Dealing with external review

What can happen with many teams, however, is that cards get hung up in that "review" column. Work may be stalled out in legal, with executives, or with some other [stakeholders](#), but wherever it may be trapped, it's outside of the team's control.

They need a way to keep working without losing sight of work that may be coming back to them. In that case, you can try using a "pen."



In this Kanban board example, work goes into the "pen" whenever it's outside the team. It stays there until it gets feedback, and then it jumps into the "ready" column. Whenever someone on the team prepares to grab a new card, they check the "ready" column and pull work from there before starting on something new from the backlog. If they grab something from "ready" they take it into "editing" to show that it has come back to the team from external review.

Kanban board 3: Using swim lanes

Finally, if your team has several sub-teams that do different kinds of work, you can try incorporating horizontal swim lanes into your board.



This allows you to get a quick look at how each team is doing and creates a simplified look at complex teamwork. You could also create swim lanes for each team member if you have a small team.

E. Backlog

As project requests are added through your intake system, they become outstanding stories in the backlog. During Agile planning sessions, your team will estimate story points (A Story Point is a measurement unit that represents the amount of effort required to complete a task. Essentially, Story Points take the place of hours when estimating tasks in an Agile environment).

It can be hard to look at a task such as “build a wireframe for X webpage” and know the exact amount of time it will take. Rather than come up with a time estimate that might be more of a guess than based on actual effort, you would assign Story Points to denote how much effort the task work requires, in comparison with other tasks in your Sprint or your Backlog.

Story Points typically are listed in a Fibonacci type of sequence (i.e., 0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100), so the numbers are far enough apart from one another to be easily distinguished when making rough estimates. Although 20, 40, and 100 aren’t numbers in the actual Fibonacci sequence, many teams use these for quicker mental calculations. A 20-point task would be four times the effort of a 5-point task, for example. Another option is to use a simple doubling of numbers: 1, 2, 4, 8, 16, 32, etc., to each task. During sprint planning, stories in the backlog are moved into the sprint to be completed during the iteration. Managing your backlog is a vital role for project managers in an Agile environment.

Agile team roles

Different Agile methodologies may require specific team roles to adhere to the framework, or may not require any specified roles. Though individual Agile implementation may not require all of these roles, here are a few common roles that you may find:

- **Scrum Master.** The Scrum Master ensures that each sprint stays on track and helps to remove or resolve any issues or challenges that may come up. They are the team’s advocate.
- **Product owner.** The role of the product owner is to define the goals of each sprint, manage and prioritize the team backlog, and be the voice of the customer or internal stakeholder.
- **Team members.** The people on this team are the ones who execute the work in each sprint. These teams, usually of three to seven people, can be composed of different specialties and strengths, or they can be teams of people with the same job roles.
- **Stakeholders.** This is an informational role only. The stakeholders should be kept up-to-date on the product and sprint goals, have the opportunity to review and approve work during a sprint, and provide feedback during the sprint retrospective.

Each Agile methodology has its own unique list of team members and roles, and while the titles may change, there are a few universal role characteristics that most [Agile team structures](#) should have:

1. **T-shaped:** A valuable Agile team member has a wide breadth of basic knowledge about their subject but also deep knowledge, experience, and ability in one (or more) specific areas.
2. **Cross-functional:** Cross-functional Agile team members have skills outside their traditional areas. They might know some basic graphic design principles and data analysis or even some HTML/CSS.
3. **Adaptable:** If they have a diverse skill set, they know how to use it. No matter the environment, their output remains consistent.
4. **Curious:** Part of optimizing and becoming more efficient is asking the right questions and challenging the way things have always been when it's appropriate.
5. **Entrepreneurial:** An Agile team member is one that doesn't wait to be told what to do. They're ready to fill in and develop campaigns where they see a need.
6. **Team-oriented:** Team players prioritize the success of the team over their own personal glory. If everyone is delivering on time and syncing well together, they see that as a win.
7. **Committed to excellence:** One of the key benefits of Agile projects is delivering quality work, faster. Team members who are committed to excellence don't settle for average. They're not hung up on perfection, but they're dedicated to always producing their best work.

What are the 6 steps in the Agile methodology?

The goal of Agile is to produce shorter development cycles and more frequent product releases than traditional waterfall project management. This shorter time frame enables project teams to react to changes in the client's needs more effectively.

As we said before, you can use a few different Agile frameworks—Scrum and Kanban are two of the most common. But each Agile methodology will follow the same basic process, which includes:

1. Project planning

Like with any project, before beginning your team should understand the end goal, the value to the organization or client, and how it will be achieved.

You can develop a project scope [here](#), [*The project scope details exactly what will be delivered at the end of the project, and the parameters of the work to be done. The purpose of the project scope is to get your [stakeholders](#) and team on the same page. It also provides a concise summary of your meetings, deliverables, and agreements for everyone to refer back to.*

Throughout your project, you'll practice scope management, which begins by writing a thorough project scope document. Having these elements laid out in plain language ensures everyone's expectations are aligned, and prevents you from overcommitting your resources] but remember that the purpose of using Agile project management is to be able to address changes and additions to the project easily, so the project scope shouldn't be seen as unchangeable.

How to Write a Project Scope

Writing a scope of project doesn't need to be a long or complex process. Using a project scope template can expedite the process. And following the steps below will help you create a detailed project scope that helps your team track progress, manage work, and put it together in a simple, useful way.

Step 1: Collect all crucial project information

Writing a project scope begins with collecting information. Project requirements come from many different places. The following information will come from a variety of sources: your stakeholders, your team, and your project sponsors; others you'll create from templates, research, and experience on other projects.

Project deliverables

This is a list of the final products you will deliver at the end of the project. You may be creating a document, website, marketing campaign, video, event, work process, or combination of these deliverables.

Deliverables should be agreed upon and approved by all involved stakeholders, presenting a unified vision of the project and giving you something substantial to measure your final product against. Think about your deliverables as your finish line—the big picture goals that will guide your schedule, budget, and resource allocation.

Pro tip: when you dig into your deliverables and start to look at work packages assigned to individuals, that's a sign you're moving into a different document—your work breakdown structure.

Available resources

Laying out your project's resources as part of your project scope will help you see exactly how you will complete your deliverables. Ongoing resource management will require visibility into team members, budget, and capacity.

When you examine who is available to work on your project, keep in mind that their time may be partially allocated elsewhere—it's common for workers to be assigned to multiple projects at once. Don't assume a team member is at 100% capacity for you. Be sure to check with other managers about how they are allocating overlapping resources to their projects.

In addition to which team members will take on specific work, be sure to articulate resources like specific software, which will impact how the project is carried out.

Inclusions and exclusions

Once deliverables and resources are agreed upon, you need to turn your attention to establishing expectations for how much work will go into the deliverables. If this sounds like an unnecessary level of detail, consider the varying expectations that can come with a project, like creating a website. Such a deliverable may sound simple enough, but will the launch include user testing? How much market and competition research will you be expected to perform? How many design iterations will you create for your stakeholders to choose from?

Stakeholders need to know what you will create and how. Otherwise, unclear expectations can lead to much dreaded scope creep, derail your budget and deadlines, and expand a project to the point of unsustainability.

It's important to list what will and will not be included in the project to avoid misunderstandings based on assumptions. Remember, you can always leave room to agree on how you will handle change orders. Just be sure to weigh any changes against how it might affect other project constraints (see next section).

Project constraints

There is a standard set of project constraints. Regardless of how many constraints your project has, remember that changing any one can impact the rest. Balancing all project constraints together is a crucial project management skill, as is laying out all constraints within your project scope. Project managers should clarify the limitations or parameters of the project so stakeholders and team members are fully aware of what may impact a project's time and budget.

Listing project constraints in your scope can also be a good way to reinforce your list of exclusions. For example, you might list that you're not including animations in a video project due to budget constraints, or that you're excluding any more than two rounds of revision due to time constraints.

You don't yet need to list a detailed project timeline at this point because you will be creating a project schedule next. But if there are time constraints beyond normal protocol, it's important for your team to know up front to plan around them.

Step 2: Build out a project schedule

Once you've collected information about the project itself, you can determine how the project will progress toward its final deadlines. Begin with your list of deliverables and drill down from there.

Major tasks and milestones

Major tasks are one level of detail below your deliverables. These are big team tasks, not individual pieces of work, that lead to completion of deliverables. When it comes to determining how long each major task will take, let quality be your guide. Don't just schedule tasks to completion, but rather to be completed with the kind of quality your scope has promised.

This schedule should then be used by individual team members to plan their own schedules around major deadlines, so be as thorough as possible and include all the tasks your team will be accountable for. And of course, it's always smart to build in some buffer periods for each task whenever you can.

Project phases

For projects that have an especially long timeline or multiple launch dates, it can be helpful to break the project up into phases to help your team with planning. This isn't mandatory, but it can give a sense of accomplishment along the way and help with organization.

Step 3: Tailor the scope to your project and organization

Because projects, stakeholders, teams, and organizations are unique, there might be information you want included in your project scope that hasn't been covered here. For example, you might consider how your scope reflects your organization's preferred methodologies. Or, you might find stating the purpose of your project helpful. Perhaps your stakeholders have a set of acceptance criteria they want met before the project can be completed.

In short, if there's something important you feel should be added, make sure its inclusion helps your document achieve its purpose: encompassing the entire scope of the project to ensure that time and resources aren't spent on unimportant additions.

Step 4: Compile and review

Now that you've done the legwork, you're ready to write your project scope. The project scope should be concise and easy to read. While it can be tempting to create an all-inclusive project scope, save the thorough write-out for your project plan. A project scope doesn't need every conceivable piece of project information. A good rule of thumb is to keep your scope to one or two paragraphs.

- Asking yourself the following questions will help you as you review and finalize your scope:
- Is it clear why this project is necessary and important?
- Do I know exactly who the project stakeholders are?
- Will my stakeholders see a value and benefit in the project?
- Are resources adequate to complete this project?
- Are timelines realistic for the deliverables?
- Does my team have access to the necessary resources?
- Is the list of inclusions and exclusions specific enough?
- Is there too much room for tasks to be added after the project has kicked off?

- Is there any project task or constraint I haven't considered yet?

Step 5: Put your project scope to work

Your project scope will serve as a basic agreement with stakeholders as well as the foundation for several other crucial documents, including your project plan, work breakdown structure, communication plan, budget, and more. You will refer to your scope throughout the duration of your project, and it will serve as a guide when making decisions or changes, making it a great tool for standing firm when people want to make changes that will require time or money you don't have.

A thorough and specific project scope will give you the big picture grasp you need to create and shape a project. Done right, a good scope will ultimately make it easier to plan, manage, and successfully execute your project.

2. Product roadmap creation

A roadmap is a breakdown of the features that will make up the final product. This is a crucial component of the planning stage of Agile, because your team will build these individual features during each sprint.

At this point, you will also develop a product backlog, which is a list of all the features and deliverables that will make up the final product. When you plan sprints later on, your team will pull tasks from this backlog.

3. Release planning

In traditional waterfall project management, there is one implementation date that comes after an entire project has been developed. When using Agile, however, your project uses shorter development cycles (called sprints) with features released at the end of each cycle.

Before kicking off the project, you'll make a high-level plan for feature releases and at the beginning of each sprint, you'll revisit and reassess the release plan for that feature.

4. Sprint planning

Before each sprint begins, the stakeholders need to hold a sprint planning meeting to determine what will be accomplished by each person during that sprint, how it will be achieved, and assess the task load. It's important to share the load evenly among team members so they can accomplish their assigned tasks during the sprint.

You'll also need to visually document your workflow for team transparency, shared understanding within the team, and identifying and removing bottlenecks.

5. Daily stand-ups

To help your team accomplish their tasks during each sprint and assess whether any changes need to be made, hold short daily stand-up meetings. During these meetings, each team member will briefly talk about what they accomplished the day before and what they will be working on that day.

These daily meetings should be only 15 minutes long. They aren't meant to be extended problem-solving sessions or a chance to talk about general news items. Some teams will even hold these meetings standing up to keep it brief.

6. Sprint review and retrospective

After the end of each sprint, your team will hold two meetings: first, you will hold a sprint review with the project stakeholders to show them the finished product. This is an important part of keeping open communication with stakeholders. An in-person or video conference meeting allows both groups to build a relationship and discuss product issues that arise.

Second, you will have a sprint retrospective meeting with your stakeholders to discuss what went well during the sprint, what could have been better, whether the task load was too heavy or too light for each member, and what was accomplished during the sprint.

If your team is new to Agile project management, don't skip this essential meeting. It helps you gauge how much your team can tackle during each sprint and the most efficient sprint length for future projects.

Source: <https://www.workfront.com/project-management/methodologies/agile>