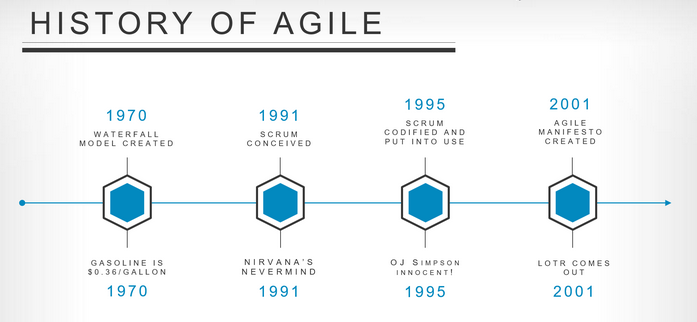
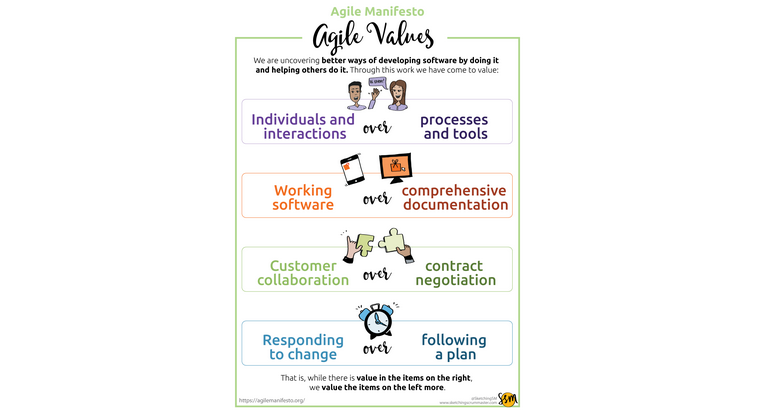
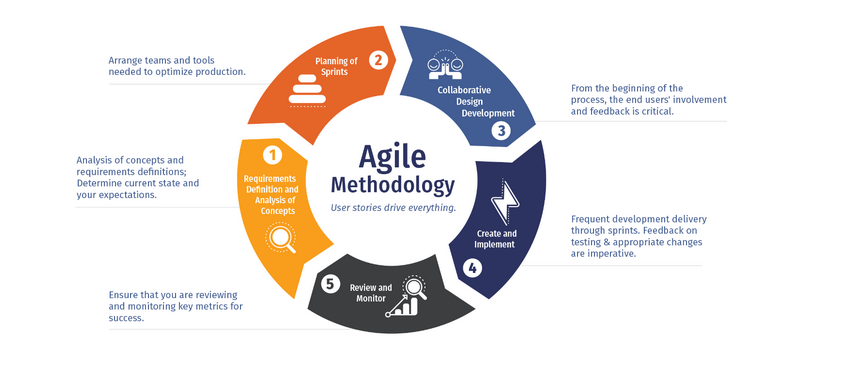
**What is the definition of a sprint in agile?**

The definition of a sprint is **a dedicated period of time in which a set amount of work will be completed on a project**. It's part of the agile methodology, and an Agile project will be broken down into a number of sprints, each sprint taking the project closer to completion.

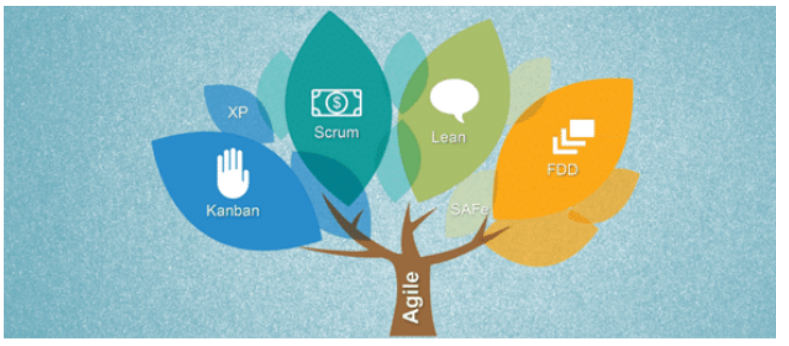


Lean:

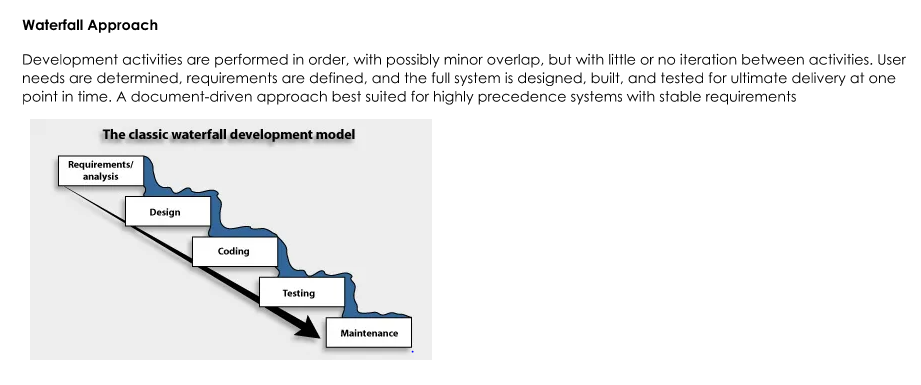




Way to implement Agile:



Waterfall Approach:





**The 12 Agile Principles**

The writers of the [agile manifesto](https://www.projectmanager.com/blog/agile-manifesto-explained?) agreed on 12 principles that define how to run an agile workflow. Let’s look at each of these 12 principles to learn what they are and how they can help you manage your projects.

**1. Satisfy the Customer Through Early and Continuous Delivery of Valuable Software**

By shortening the time between documenting the project, reporting to your customer and then getting feedback, you can focus on the real goal of the project, which is delivering what the customer wants, not what you planned.

**2. Welcome Changing Requirements, Even Late in Development**

Embrace change. Even when the customer [requests a change](https://www.projectmanager.com/templates/change-request-form?) late in the project phase, implement it. Why wait for another project to explore another iteration when you can do it now and get the results immediately? Agile wants you to stay nimble and on your feet so you can pivot without having to constantly reinvent the wheel.

**3. Deliver Working Software Frequently**

If you’re going to embrace change, then you’re going to have to give up on your etched-in-stone schedule, or at least create a shorter range to run your tasks. One way agile does this is by cutting out a lot of the documentation that is required with [traditional project management](https://www.projectmanager.com/project-management?) when planning your schedule before you ever start a task. The trouble is a lot of that paperwork isn’t necessary. It only slows things down.

You need to reach an agreement with your team and stakeholders to come up with an [agile release planning](https://www.projectmanager.com/blog/agile-release-planning-tips?) that satisfies both parties.

**4. Business People and Developers Must Work Together**

It’s like they’re talking two different languages, and in a sense, they are, but both the business and developer sides of the project are crucial to its success. You must build a bridge between [the stakeholders](https://www.projectmanager.com/stakeholder-management?) so they can understand each other and, as important, work together. Use the same tools you would manage [remote teams](https://www.projectmanager.com/training/3-new-tips-managing-remote-teams?) to facilitate an exchange of ideas that both sides understand and are on board with.

**5. Build Projects Around Motivated Individuals**

In other words, don’t micromanage. It doesn’t work. It takes you away from what you should be focusing on. It erodes morale and sends talent packing. You assembled the best, now let them do what they’re good at. If you did the due diligence beforehand, then you can trust them to do the work. Of course, you’ll [monitor that work,](https://www.projectmanager.com/project-tracker?) and step in as needed, but stay out of their way.

**6. Promote Face-to-Face Conversations**

Documenting conversations, creating email narrative streams and even using [collaboration software](https://www.projectmanager.com/software/collaboration?) like Slack, are all well and good. But when you’re trying to move swiftly, you don’t have time to wait for a reply. You need immediate answers, and the only way to achieve that speed of response is by talking to your team member or team in person. You can do this by working in the same physical space or having distributed teams. But if it’s the latter, you want to try and keep the schedules to the same hours, so you can at least video conference. That creates a more collaborative environment.

**7. Working Software Is the Primary Measure of Progress**

That means, is the software (or whatever product or process you’re working on in the project) working correctly? You’re not measuring progress by checking off tasks and moving across your [scheduled timeline,](https://www.projectmanager.com/project-timeline?) but by the success of the software (or whatever) is the subject of your project. Basically, it’s staying focused on what’s important. The process is what gets you to achieve the goal of the project, but the goal of the project isn’t the process.

**8. Agile Processes Promote Sustainable Development**

One reason for short sprints of activity is not only that they lend themselves to accepting change more readily, but they also help to keep your teams motivated. If you’re working on a project for an extended period, there’s going to be burnout. It’s unavoidable. Don’t overtax your team with too much overtime. It’s going to impact the quality of your project. So, get the right team for the job, one that will work hard but not overextend themselves and put the project quality in jeopardy.

**9. Continuous Attention to Technical Excellence and Good Design Enhances Agility**

Whether you’re working on code or something more concrete, you want to make sure that after each iteration it’s improving. You don’t want to have to come back and fix things later. Fix them now. Better still, make sure they’re getting better. Use [scrum](https://www.projectmanager.com/blog/scrum-methodology?), an agile framework for completing complex projects, to help review and keep the project evolving.

**10. Simplicity—the Art of Maximizing the Amount of Work Not Being Done—is Essential**

If you’re looking to move quickly through a project, then you’re going to want to cut out unnecessary complexities. Keeping things as simple as possible is a great ethic to streamline your process. You can do this many ways, including the use of [agile tools](https://www.projectmanager.com/software/agile?) that cut out the busy work and give you more control over every aspect of the project.

**11. The Best Architectures, Requirements and Designs Emerge from Self-organizing Teams**

When you have a strong team, you want to give that team the autonomy to act independently. This means they can adapt to change quicker. In fact, they can do everything with greater agility because you’ve given them the trust to act without second-guessing them. If you’ve done your job in collecting the right people, then they’ll do their job addressing issues and resolving them before they become problems.

**12. Have Regular Intervals feedback meeting**

Another benefit of creating a well-rounded team is that they will stop, reflect and tweak the way they do things throughout the course of the project. They don’t act by rote or just blindly follow protocol, but think through their relationship to the project and adjust when necessary. The last thing you want is a complacent team, one that stands on their laurels. What you need is an ever-evolving group that is constantly engaged and looking for ways to improve productivity.