**Agile Software Development**

[Agile methods](https://www.versionone.com/agile-101/agile-methodologies/) grew out of the real-life project experiences of leading software professionals who had experienced the challenges and limitations of traditional waterfall development on project after project. The approach promoted by agile development is in direct response to the issue associated with traditional software development both in terms of overall philosophy as well as specific processes. The overall goal of each Agile method is to adapt to change and deliver working software as quickly as possible.  full Agile software development lifecycle includes the concept, inception, construction, release, production, and retirement phases.

Examples of Agile tools are Rally, Version One, Jira

What is Scrum Project Management?

[Scrum](http://info.versionone.com/sutherland-power-of-scrum-agile-video.html) is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks. It is one of the approaches that influenced the [Agile Manifesto](http://agilemanifesto.org/), which articulates a set of values and principles to guide decisions on how to develop higher-quality software faster.

Who Uses Agile Scrum Methodology?

Scrum is widely used by software development teams. In fact it’s the [most popular agile methodology](https://www.versionone.com/agile-101/agile-methodologies/). According to the [10th Annual State of Agile](http://stateofagile.versionone.com/) report, 75% of software teams use Scrum or a Scrum hybrid. However, Scrum has spread to other business functions including IT and marketing where there are projects that must move forward in the presence of complexity and ambiguity. Leadership teams are also basing their agile management practices on Scrum, often combining it with lean and Kanban practices (subgroups of agile project management).

What is Scrum in Relation to Agile Project Management?

Scrum is a sub-group of agile:

* Agile is a set of values and principles that describe a group’s day-to-day interactions and activities. Agile itself is not prescriptive or specific.
* The Scrum methodology follows the values and principles of agile, but includes further definitions and specifications, especially regarding certain software development practices.

Although developed for agile software development, agile Scrum became the preferred framework for agile project management in general and is sometimes simply referred to as Scrum project management or Scrum development.

What are the Benefits Received from the Scrum Methodology?

Organizations that have adopted agile Scrum have experienced:

* Higher productivity
* Better-quality products
* Reduced time to market
* Improved stakeholder satisfaction
* Better team dynamics
* Happier employees

What is So Special About Scrum Project Management?

Scrum addresses complexity in work by making information transparent, so that people can inspect and adapt based on current conditions, rather than predicted conditions. This allows teams to address the common pitfalls of a waterfall development process: chaos resulting from constantly changing requirements; underestimation of time, resources and cost; compromises on software quality; and inaccurate progress reporting. Transparency of common terms and standards is required in Scrum development to ensure that what is being delivered is what was expected. Frequent inspection ensures progress and detects variances early on so that adjustments can be made quickly. The most common Scrum events for inspection and adaptation are: Sprint Planning, Daily Scrum or “Stand Up”, Sprint Review, and Sprint Retrospective (see Scrum Events below).

What is Scrum Methodology Compared to Other Agile Approaches?

Most enterprises first transition individual teams to agile before they “scale” to the rest of the organization. Scaling agile is not easy, which has recently prompted new frameworks to emerge, such as the Scaled Agile Framework® and Disciplined Agile Delivery (DAD) This popularity has made Scrum a significant piece of many agile application lifecycle management (agile ALM) initiatives.

What Are the Components of Agile Scrum Development?

The Scrum methodology is defined by team roles, events (ceremonies), artifacts, and rules.

The Scrum Team

Scrum teams are typically composed of 7 +/- 2 members and have no team leader to delegate tasks or decide how a problem is solved. The team as a unit decides how to address issues and solve problems. Each member of the Scrum team is an integral part of the solution and is expected to carry a product from inception to completion.

There are three key roles in a Scrum team:

***The Product Owner***  
The Product Owner is the project’s key stakeholder – usually an internal or external customer, or a spokesperson for the customer. There is only one Product Owner who conveys the overall mission and vision of the product which the team is building. The Product Owner is ultimately accountable for managing the product backlog and accepting completed increments of work.

* This person owns the Product backlog and writes user stories and acceptance criteria.
* Product Backlog is prioritized by him and he decides the release date and the content.
* He accepts or rejects user stories.
* He has the power to cancel the Sprint, if he thinks the Sprint goal is redundant.
* He is the one who is responsible for the Return on Investment of the product.

***The ScrumMaster***  
The [ScrumMaster](http://versionone.com/training/agile_training/?d=public" \t "_blank) is the servant leader to the Product Owner, Development Team and Organization. With no hierarchial authority over the team but rather more of a facilitator, the ScrumMaster ensures that the team adheres to Scrum theory, practices, and rules. The ScrumMaster protects the team by doing anything possible to help the team perform at the highest level. This may include removing impediments, facilitating meetings, and helping the Product Owner groom the backlog.

* He is a facilitator and Servant Leader who encourages and demands self-organization from the development team.
* He enables close cooperation across all roles and functions, addresses resource issue and disobedience of scrum practices.
* He protects the team from external and internal distractions.
* He removes impediments so the team can focus on the work at hand and follow scrum practices.
* He is not typically a manager or lead, but he is an influential leader and coach who does not do direct command and control.

***The Development Team***  
The Development Team is a self-organizing, cross-functional group armed with all of the skills to deliver shippable increments at the completion of each sprint. Scrum broadens the definition of the term “developer” beyond programmers to include anyone who participates in the creation of the delivered increment. There are no titles in the Development Team and no one, including the ScrumMaster, tells the Development Team how to turn product backlog items into potentially shippable increments

**Scrum Events (Ceremonies)**

***The Sprint***  
A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprints are usually 2-4 weeks long but can be as short as one week.

***Sprint Planning***  
Sprint Planning team meetings are time-boxed events that determine which product backlog items will be delivered and how the work will be achieved.

***The Daily Stand-up***  
The Daily Stand-up is a short communication meeting (no more than 15 minutes) in which each team member quickly and transparently covers progress since the last stand-up, planned work before the next meeting, and any impediments that may be blocking his or her progress.

***The Sprint Review***  
The Sprint Review is the “show-and-tell” or demonstration event for the team to present the work completed during the sprint. The Product Owner checks the work against pre-defined acceptance criteria and either accepts or rejects the work. The stakeholders or clients give feedback to ensure that the delivered increment met the business need.

***The Retrospective***  
The Retrospective, or Retro, is the final team meeting in the Sprint to determine what went well, what didn’t go well, and how the team can improve in the next Sprint. Attended by the team and the ScrumMaster, the Retrospective is an important opportunity for the team to focus on its overall performance and identify strategies for continuous improvement on its processes.

**Scrum Artifacts**

***Product Backlog***  
The product backlog is the single most important document that outlines every requirement for a system, project or product. The product backlog can be thought of as a to-do list consisting of work items, each of which produces a deliverable with business value. Backlog items are ordered in terms of business value by the Product Owner.

***Sprint Backlog***  
A sprint backlog is the specific list of items taken from the product backlog which are to be completed in a sprint.

***Increment***  
An Increment is the sum of all product backlog items that have been completed since the last software release. While it is up to the Product Owner to decide on when an increment is released, it is the team’s responsibility to make sure everything that is included in an increment is ready to be released. This is also referred to as the Potentially Shippable Increment (PSI).

**Scrum Rules**

The rules of agile Scrum should be completely up to the team and governed by what works best for their processes. The best agile coaches will tell teams to start with the basic scrum events listed above and then inspect and adapt based on your team’s unique needs so there is continuous improvement in the way teams work together.

# **What is Sprint planning in Agile project management?**

Sprint Planning meeting is one of the key activities in Agile project management.

Generally, Agile teams need to deliver a working product at the end of  each Sprint. However some organizations also use a **Sprint Zero**. Sprint Zero is used exclusively for planning the first Sprint and **no work product is delivered** at the end of Sprint Zero. Usage of Sprint Zero is decided by the team and complexity of the project.

Not all projects and organizations use Sprint zero. Many teams directly start with the first Sprint and the activities listed below are incorporated as part of Sprint planning meeting.

The developers and testers collaborate with the team on the following activities during **Sprint Planning** meeting:

* The Product Owner explains the release goal and walk through the release backlog.
* The Product Owner also explains the big picture and vision.
* The team with the help of the Product Owner breaks down requirements into user stories, details the acceptance criteria and sizes them.
* The team also creates a high level architecture design.
* The team plans for necessary infrastructure items and tools for the release (e.g., for test management, defect management, test automation, and continuous integration)
* Team creates the initial test strategy for various test levels to understand the test scope; technical risks and initial risks analysis (see [Testing Quadrants, Test Levels and Testing Types](http://istqbexamcertification.com/what-are-test-pyramid-and-testing-quadrants-in-agile-testing-methodology/)).
* The team establishes various metrics to measure the progress with the help of Scrum Master.
* The brainstorms and creates the Definition of done.
* The team creates the big visible task board

**Name as many characteristics of agile development as you can?**

like cross-functional team composition, face-to face communication, solving problems immediately after these are identified, working solution as a primary metric of progress.

Agile can be characterized by many ways. It is probably the most operational software development method, it is the one when the interaction in between the team members is the highest and typically it helps to shorten the product development cycle and avoid problems to occur once the team is finalizing, or testing the project.

But what I like the most about agile is the time efficiency. All the projects I worked on using agile methodology were completed in a given time frame and actually even earlier. The results were as expected, team members learned something from each other and at the end of the project, everyone was satisfied.

**Mention the key difference between sprint backlog and product backlog?**

Product backlog: It contains a list of all desired features and is owned by the product owner

Sprint backlog: It is a subset of the product backlog owned by development team and commits to deliver it in a sprint. It is created in Sprint Planning Meeting.

**Mention what are the Tools that can be useful for screenshots while working on Agile projects?**

While working on Agile projects you can use tools like

BugDigger, BugShooting, qTrace, Snagit, Bonfire, Usersnap

**Explain how can you implement scrum in an easy way to your project?**

These are the tips which can be helpful to implement scrum in your project

* Get your backlog in order
* Get an idea of the size of your product backlog items
* Clarify sprint requirement and duration to complete the sprint backlog
* Calculate the team sprint budget and then break requirements into tasks
* Collaborate workspace- a center of all team discussion, which includes plans, roadmaps, key dates, sketches of functionality, issues, log, status reports, etc.
* Sprint- Make sure you complete one feature at a time before moving on to the next. A sprint should not be abort unless if there is no other option
* Attend a daily stand-up meeting: In meeting you need to mention, what have been achieved since the last meeting, what will they achieve before the next meeting and is anything holding up their progress
* Use burndown chart to track daily progress. From the burndown chart, you can estimate whether you are on track, or you are running behind
* Complete each features well before moving on to the next
* At the end of the sprint- hold a sprint review meeting, mention what is achieved or delivered in the sprint.

**Explain what does it mean by product roadmap?**

A product roadmap is referred for the holistic view of product features that create the product vision.

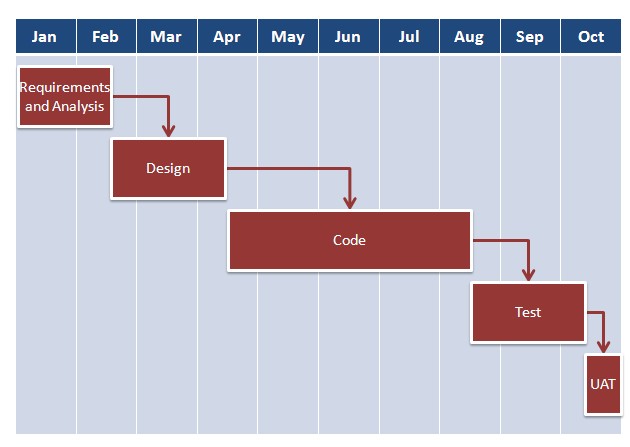
## Example of Agile software development

Example: Google is working on project to come up with a competing product for MS Word, that provides all the features provided by MS Word and any other features requested by the marketing team. The final product needs to be ready in 10 months of time. Let us see how this project is executed in traditional and Agile methodologies.

**In traditional Waterfall model –**

* At a high level, the project teams would spend 15% of their time on gathering requirements and analysis (1.5 months)
* 20% of their time on design (2 months)
* 40% on coding (4 months) and unit testing
* 20% on System and Integration testing (2 months).
* At the end of this cycle, the project may also have 2 weeks of User Acceptance testing by marketing teams.
* In this approach, the customer does not get to see the end product until the end of the project, when it becomes too late to make significant changes.

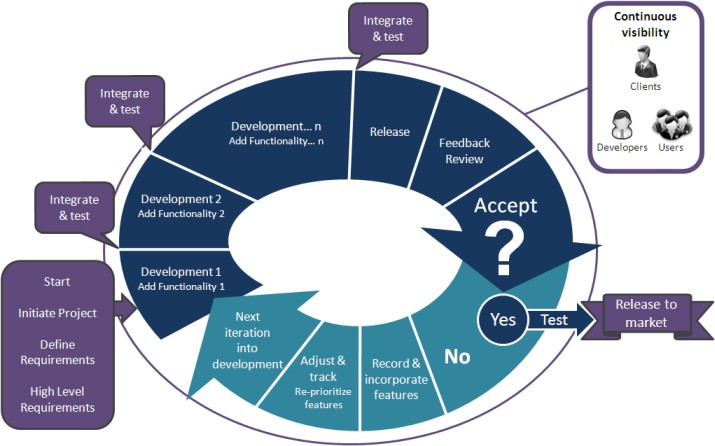
The image below shows how these activities align with the project schedule in traditional software development.

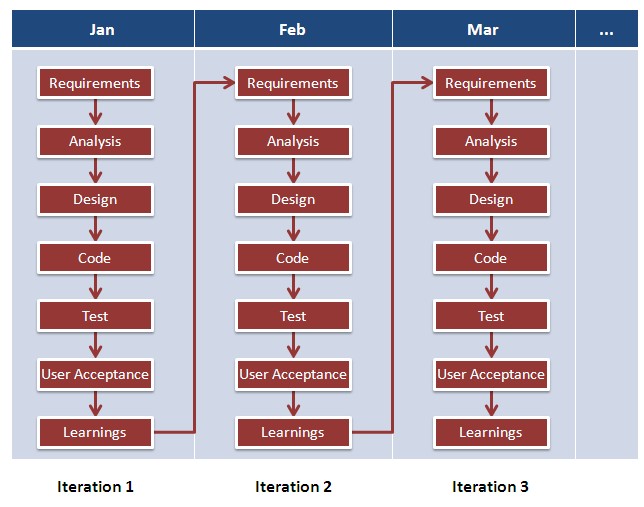


With **Agile development** methodology –

* In the [Agile methodology](http://istqbexamcertification.com/what-is-agile-methodology-examples-when-to-use-it-advantages-and-disadvantages/), each project is broken up into several ‘Iterations’.
* All Iterations should be of the same time duration (between 2 to 8 weeks).
* At the end of each iteration, a working product should be delivered.
* In simple terms, in the Agile approach the project will be broken up into 10 releases (assuming each iteration is set to last 4 weeks).
* Rather than spending 1.5 months on requirements gathering, in Agile software development, the team will decide the basic core features that are required in the product and decide which of these features can be developed in the first iteration.
* Any remaining features that cannot be delivered in the first iteration will be taken up in the next iteration or subsequent iterations, based on priority.
* At the end of the first iterations, the team will deliver a working software with the features that were finalized for that iteration.
* There will be 10 iterations and at the end of each iteration the customer is delivered a working software that is incrementally enhanced and updated with the features that were shortlisted for that iteration.

The iteration cycle of an Agile project is shown in the image below.





This approach allows the customer to interact and work with functioning software at the end of each iteration and provide feedback on it. This approach allows teams to take up changes more easily and make course corrections if needed. In the Agile approach, software is developed and released incrementally in the iterations. An example of how software may evolve through iterations is shown in the image below.

**Waterfall Model Advantages and disadvantages:**



**Advantages of waterfall model:**

* This model is simple and easy to understand and use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model phases are processed and completed one at a time. Phases do not overlap.
* Waterfall model works well for smaller projects where requirements are very well understood.

**Disadvantages of waterfall model:**

* Once an application is in the [**testing**](http://istqbexamcertification.com/what-is-a-software-testing/) stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

**When to use the waterfall model:**

* This model is used only when the requirements are very well known, clear and fixed.
* Product definition is stable.
* Technology is understood.
* There are no ambiguous requirements
* Ample resources with required expertise are available freely
* The project is short.

Very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demoed to the end users. Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everywhere from document till the logic.

**More Information on agile:**

<http://reqtest.com/testing-blog/agile-methodology-interview-questions/>

<http://career.guru99.com/top-20-agile-testing-interview-questions/>

**Agile Manifesto principles**

<https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>