<https://www.jile.io/agile-basics/scrum-basics>

<https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-agile>

<https://www.simplilearn.com/tutorials/agile-scrum-tutorial/user-stories>

<https://www.guru99.com/jira-epic.html>

<https://www.guru99.com/agile-testing-interview-questions.html>

In today’s modern software development business, the two most important and popular approaches of project management are:

●     Waterfall – This is a traditional software development approach.

●     Agile –  is a type of Rapid Application Development that falls into a certain category.

## What is a Waterfall?

The Waterfall Methodology was the first to be designed and deployed as a Process Model. Another name for this kind of model is a straightforward life cycle model. It is very easy to understand and use. In a waterfall methodology, each phase of work must be completed before moving on to the next phase, and that there is no crossover between the phases in this model.

The waterfall model portrays the software development life cycle as a sequential, linear flow of information. This implies that any stage in the development phase may begin only once the previous phase has been successfully completed. This waterfall model’s stages do not cross in any manner.

#### Pros of Waterfall model

* It is among the most simple types to operate. Because of the project’s nature, each phase contains deliverables and also a review process.
* It is best suitable for smaller tasks with defined requirements that are simple to understand and execute.
* Simple to adopt a method for restructuring teams

#### Cons of Waterfall Model

* This is not the preferred product for a big project.
* This approach is less effective if the requirement is not clearly defined at the start.
* It is difficult to travel ahead and make changes to previous stages.
* The testing step starts after the design phase is finished. As a result, there is a high probability that flaws will be found required during the development process when they will be more costly to repair.

By definition, the word Agile means the “able to move quickly and easily.”

Agile methodologies enable organizations to deliver value to customers faster and with fewer complications by systematically managing projects and developing software in an iterative fashion.

Agile is an iterative approach of software development methodology using short iterations of 1 to 4 weeks. Using Agile methodology, the software is distributed with fastest and fewer changes.

The approach of an agile team is to deliver work in small, but consumable, increments, rather than wagering everything on a "big bang" launch.

The advantages of agile methodology are customer satisfaction by rapid, continuous development and delivery of useful software.

What is Agile Methodology?

An agile methodology is an iterative approach to software development. Each iteration of agile methodology takes a short time interval of 1 to 4 weeks. The agile development process is aligned to deliver the changing business requirement. It distributes the software with faster and fewer changes.

The single-phase software development takes 6 to 18 months. In single-phase development, all the requirement gathering and risks management factors are predicted initially.

**What are the important Agile Model Manifestos?**

Here is the essential manifesto of the Agile Model:

* Individuals and interactions are given priority over processes and tools.
* Adaptive, empowered, self-organizing team.
* Focuses on working software rather than comprehensive documentation.
* Agile Model in software engineering aims to deliver complete customer satisfaction by rapidly delivering valuable software.
* Welcome changes in requirements, even late in the development phase.
* Daily co-operation between businesspeople and developers.
* Priority is customer collaboration over contract negotiation.
* It enables you to satisfy customers through early and frequent delivery.
* A strong emphasis is placed on face-to-face communication.
* Developing working software is the primary indicator of progress.
* Promote sustainable development pace.
* A continuous focus is placed on technical excellence and sound design.
* An improvement review is conducted regularly by the team.

## Phases of Agile Model

Here are the important stages involved in the Agile Model process in the SDLC life cycle:

* **Requirements Gathering:** In this Agile model phase, you must define the requirements. The business opportunities and the time and effort required for the project should also be discussed. By analyzing this information, you can determine a system’s economic and technical feasibility.
* **Design the Requirements:** Following the feasibility study, you can work with stakeholders to define requirements. Using the UFD diagram or high-level UML diagram, you can determine how the new system will be incorporated into your existing software system.
* **Develop/Iteration:**The real work begins at this stage after the software development team defines and designs the requirements. Product, design, and development teams start working, and the product will undergo different stages of improvement using simple and minimal functionality.
* **Test:** This phase of the Agile Model involves the testing team. For example, the Quality Assurance team checks the system’s performance and reports bugs during this phase.
* **Deployment:** In this phase, the initial product is released to the user.
* **Feedback:** After releasing the product, the last step of the Agile Model is feedback. In this phase, the team receives feedback about the product and works on correcting bugs based on the received feedback.

## Types of Agile

**Scrum:**This agile method focuses primarily on managing tasks in team-based development conditions. In the[Scrum Agile model](https://www.guru99.com/agile-scrum-extreme-testing.html#scrum), the team should strictly follow a work plan for each Sprint. Moreover, people involved in this type of project have predefined roles.

**Crystal:**Using Crystal methodology is one of the most straightforward and most flexible approaches to developing software, recognizing that each project has unique characteristics. Therefore, policies and practices need to be tailored to suit them.

Crystal methodologies are categorized as below:

* **CLEAR:**User for small and low critical efforts.
* **ORANGE:**User for moderately larger and critical projects.
* **ORANGE WEB:**Typically, electronic business

**Dynamic Software Development Method (DSDM)**: This Rapid Application Development (RAD) approach involves active user involvement, and the teams are empowered to make decisions with the goal of frequent product delivery.

**Feature Driven Development (FDD):**This Agile method focuses on “designing & building” features. It is divided into several short phases of work that must be completed for each feature separately. It includes domain walkthrough, design inspection, code inspection, etc.

**Lean Software Development:**This methodology is based on the principle of “Just-In-Time Production.” It helps to increase the speed of software development and decrease costs.

As a result of a lean development model, waste is eliminated, learning is amplified, early delivery is achieved, and integrity is built.

**Extreme Programming (XP):**[Extreme Programming](https://www.guru99.com/agile-scrum-extreme-testing.html#extreme-programming) is a useful Agile model when there are constantly changing requirements or demands from clients. It is also used when there is no sure about the system’s functionality.

**When to use the Agile Model?**

Here are the common scenarios where the Agile method is used:

* It is used when there are frequent changes that need to be implemented.
* Low-regulatory-requirement projects
* Projects with not very strict existing process
* Projects where the product owner is highly accessible
* Projects with flexible timelines and budget

**Advantages of the Agile Model**

Here are some common pros/benefits of the Agile Model:

* Communication with clients is on a one-on-one basis.
* Provides a very realistic approach to software development
* Agile Model in software engineering enables you to draft efficient designs and meet the company’s needs.
* Updated versions of functioning software are released every week.
* It delivers early partial working solutions.
* Changes are acceptable at any time.
* You can reduce the overall development time by utilizing this Agile Model.
* It allows concurrent development and delivery within an overall planned context.
* The final product is developed and available for use within a few weeks.

**Limitations of Agile Model**

* It is not useful method for small development projects.
* It requires an expert to take important decisions in the meeting.
* Cost of implementing an agile method is little more compared to other development methodologies.
* The project can easily go off track if the project manager is not clear what outcome he/she wants.

## Key Differences

* Waterfall is a Linear Sequential Life Cycle Model, whereas Agile is a continuous iteration of development and testing in the software development process.
* In Agile vs Waterfall difference, the Agile methodology is known for its flexibility, whereas Waterfall is a structured software development methodology.
* Comparing the Waterfall methodology vs Agile, which follows an incremental approach, whereas the Waterfall is a sequential design process.
* Agile performs testing concurrently with software development, whereas in Waterfall methodology, testing comes after the “Build” phase.
* Agile allows changes in project development requirements, whereas Waterfall has no scope of changing the requirements once the project development starts.

Key Agile Concepts

Here are a few essential Agile concepts.

User Stories: The team divides the work into functional units known as "user stories" in consultation with the client or product owner. Each user story must add something valuable to the final product.

* Daily Meeting: The team meets every day at the same time to update everyone on the information necessary for coordination:
* Personas: When the project requires it, the team creates in-depth, fabricated biographies of hypothetical users of the intended product.
* Team: A small group of individuals assigned to the same project or effort, almost all of whom work full-time, is referred to as a "team" in the Agile context.
* Incremental Development: Agile teams prefer to use an incremental development strategy, which in an Agile setting means that each iteration of the product improves on the one before it by including user-visible functionality.
* Iterative development: Agile projects intentionally permit "repeating" software development activities and the potential for "revisiting" the same work products, known as iterative development.
* Milestone Retrospective: After a project has been running for a while, the team dedicates one to three days to examine the key moments.

## What is Scrum?

Scrum is a simple empirical process that enables teams to build products incrementally in iterations, to keep abreast of the changing market needs and align themselves to the organization's business goals.

Scrum advocates self-organizing teams working towards a common goal through continuous inspection and adaptation. A minimum viable product at the end of each iteration provides an option for the teams to quickly get feedback from end users and respond much faster.

## Scrum Methodology

Scrum is simple light weighted agile project management methodology that enables product teams to build products incrementally in an iterative fashion through effective team collaboration.

he Scrum Methodology is defined by:

* Scrum Roles
* Scrum Events
* Scrum Artifacts
* Scrum Values

## What are Scrum Roles?

The scrum team is made up of three roles: A Product Owner, a Scrum Master and the Development Team.

**1. Product Owner:**

A Product Owner in a scrum team decides what needs to be built. This person has complete knowledge about the market and business needs, has a vision, and owns the return on investment (ROI) or the value delivered by the product.

Unlike traditional delivery, this person is a part of the team that delivers the product.

Following are the key responsibilities of the Product Owner:

* Creates the vision
* Represents business, and is responsible for the ROI
* Cascades the vision to the teams
* Owns the backlog of features
* Prioritizes features by market value
* Is empowered to take decisions
* Negotiates with the team and business to deliver the right product at the right time

**2. Scrum Master:**

The Scrum Master is not a management title and cannot make decisions on behalf of the team. The Scrum Master's major responsibility is to ensure that scrum is understood and practiced by every team member in the true spirit.

The Scrum Master should understand the different skill sets of his or her team and group them by having the right sheep in the right flock. A Scrum Master should guide the team such that the team does not go astray and fall prey to excess time and energy.

Just like a shepherd, a Scrum Master must draw out quiet people during stand-up meetings or when planning poker sessions. Whenever the team loses focus or a team member goes astray, the Scrum Master aka the shepherd should bring the lost one back to the flock and guide appropriately.

The Scrum Master should not enforce agile practices on the team, but should do a 'Servant leadership' role. Scrum Master should lead by example and be a living demonstration of team assets and scrum values.

He or she should create an environment of safety for the team, and guide and facilitate team collaboration. He or she should refrain from solving problems or making decisions by guiding teams to do so.

To summarize, a Scrum Master:

* **Is a servant leader -**mentors and coaches the teams on scrum theory and practices, guides them on how they need to adapt to those, thereby realizing the benefits of scrum both at team level and organization level.
* **Helps remove obstacles/impediments -**supports the Development teams in removing the impediments by reaching out to the right people, thereby ensuring a smooth development progress without disrupting the team.
* **Facilitates collaboration -**enables interactions within the team as well as between the team and the Product Owner.
* **Teaches scrum -**to the team.
* **Protects the teams -**from external disruptions such as changes to stories in the current sprint.
* **Is a change agent -**in growing the organization to deliver early and often, and removing waste.

**3. The Development team:**

The Development team in scrum is the team that has all the skills necessary to execute the backlog items. This team is not a normal team but is committed, dedicated, and motivated to perform the best.

It is a self-organizing team that collaborates, shares their special skills and knowledge and are committed completely to fulfil the objective.

The following are the special characteristics of 'The Development team':

* **Self-organizing -**the Development team will be a self-managing group, who will decide on the tasks that they will work on incrementally. There is no 'Manager', who controls their work.
* **Empowered -**the team should commit to work, determine 'HOW' to deliver and decide on 'HOW MUCH' to deliver in one iteration.
* **Cross-functional -**the team does not segregate members as [agile developers](https://www.jile.io/solutions/product-management-software-for-agile-developers), testers or analysts and each member has the necessary skills to deliver the product increment.
* **Small-sized -**the Development team should ideally have 5 to 9 team members with skills sufficient to deliver the committed work. Smaller teams will not have the bandwidth to complete a considerable work and larger teams will increase the complexity.
* **Co-located -**the agile team is typically co-located to ensure effective collaboration.
* **Committed -**since the team is empowered to take decisions on the scope of work in a sprint, they are committed to delivery, should be transparent on the progress, and highlight the impediments early on.
* **Dedicated -**this team is focused and is 100% dedicated to product delivery.

Unlike traditional methodologies, where the commitment to deliver is made to business by the team that is not involved in the execution, in Agile, the team that does the work commits to how much work can be executed in a sprint.

The Development team decides how much work is to be done in a sprint, and commits to delivering a 'potentially shippable product increment (PSPI)', without sacrificing quality and speed. The team also makes continuous self-improvements.

**What are Scrum Artifacts?**

Scrum focuses more on a working software at the end of every sprint rather than comprehensive documentation. This does not imply that there is no documentation at all. The documentation is to facilitate collaboration and interactions, rather than tracking.

The progress is always measured through a working software. Scrum Documentation has three main artifacts namely: Product backlog, Sprint backlog, Increment.

**1. Product backlog:**

A product backlog is a dynamic list of functionalities the product might include, such that it provides value to users.

The Product Owner maintains this list and is responsible for creating, managing, and prioritizing the backlog by focusing on 'WHAT' brings the highest value to the users. These are a few unique characteristics of a product backlog:

* It is dynamic in nature as it evolves based on changing market needs
* Lists all the features and capabilities that will be taken up in iteration and delivered as a product increment
* It is refined on a continuous basis. The Product Owner and Development team collaborate and update the details, estimate, and prioritize based on business value and size

**2. Sprint backlog:**

Sprint backlog is a subset of the entire product backlog that the scrum team plans to implement in one iteration or sprint.

During the sprint planning, the team selects items from the product backlog that they commit to complete in one sprint and thereby, create the sprint backlog. The Product Owner and Scrum Master should not provide inputs that may impact the team's decision. Sprint backlog has:

* Subset of product backlog items that the teams commit to implement in one sprint
* Items broken into smaller pieces of work as tasks
* A focus on 'HOW' the team does the work and delivers the value in one sprint
* A story or task board that is used by the teams to view backlog and what the individuals sign up for work after [backlog prioritization](https://www.jile.io/blogs/five-factors-influencing-backlog-prioritization)
* Provision for the Development teams to track the sprint progress and check their alignment with sprint goals

**3. Increment:**

An increment is the work delivered at the end of every sprint. Typically, after every iteration there will be a Product Increment (PI) that delivers value and the final product will be a working software.

This increment is a sum of all the capabilities that were delivered in the previous sprints as a part of the PI. At the end of every sprint, the Product Owner decides whether to release the working product increment or wait until the next release.

**What are Scrum Events?**

All scrum activities are time-boxed and allow teams to inspect their current work and implement those learnings in future time-boxes.

**Heart of Scrum - The Sprint**

At the heart of scrum, is the 'Sprint'. The sprint is a time-boxed iteration, typically ranging from 1 to 4 weeks, at the end of which, a potentially shippable product increment is delivered by the Development team.

The sprint has the following characteristics:

* Does not exceed a maximum of one calendar month as this will increase the risk due to changes in requirements and therefore, may not provide the perceived business value at the end of the sprint
* Has a goal or 'definition of done' associated with every sprint that actually measures the success of the sprint
* Can be cancelled by the Product Owner, if the goal or the need for the sprint becomes obsolete due to changing market needs

Scrum advocates specific types of activities or meetings within a sprint to avoid the traditional formal meetings. These events and meetings are conducted at regular intervals and happen at specific periods of the sprint.

Typical Scrum Events are:

* Sprint planning
* Daily scrum
* Sprint review
* Sprint retrospective

# 11 BEST Scrum Tools for Agile Project Management (2022)

### 6) Jira

Jira is one of the popular Scrum Project management tools that offers thousands of functionalities such as Scrum boards, customizable backlogs, reporting options, to name a few. It is one of the best scrum project management tools for IT and software development companies and large business organizations.

**eatures:**

* Offers an advanced reporting solution
* Provide customizable Scrum dashboards
* Flexible Kanban boards
* Provide a single view for all user stories
* Create customer filters using Jira Query Language (JQL).
* Option for Search and filtering
* Integrate with developer tools for end-to-end traceability.
* Roadmap creation tools
* JIRA founded in 2002
* Offers AES-256 encryption
* Export your reports in PDF, Excel, and CSV
* Visualize your project using various views like List, Timeline, Kanban, and Calendar views
* Free plan offers 2 GB storage, and 10 Users
* Provides instant notification via Email, and Desktop
* Offers Drag & Drop, custom reports, centralized workspace, charts, and Time Tracking
* Seamlessly integrates with Figma, Miro, Power BI, Zephyr, GitLab, Ketryx and draw.io
* Offers pre-built templates for Confluence, Page, Project management, Project Plan, Bug tracking, DevOps, Kanban and Scrum
* Provides project management status like New, In Progress and Done
* Offers Scrum boards, Roadmaps, Reports and insights and Project flexibility
* It provides customer support via Phone and Contact form
* **Supported Platforms:** Android, and iOS
* **Price:** Request a Quote from Sales
* **Free Trial:** 14 Days Free Trial

## Epic Vs. Story Vs. Task

Task, Story, and Epic are the key elements of Jira that help to plan, structure, and execute project-related tasks. Some characteristics of these elements are given here:

#### Task:

Task is the smallest element of a project and can be completed in one working day. Task can be assigned to a team member and marked as completed when finished. Some other features of task are:

* Task is a part of sprint or scrum.
* If the task is bigger, it can also be split into sub-tasks.
* Tasks can be linked and can be locked by other tasks.

#### Story:

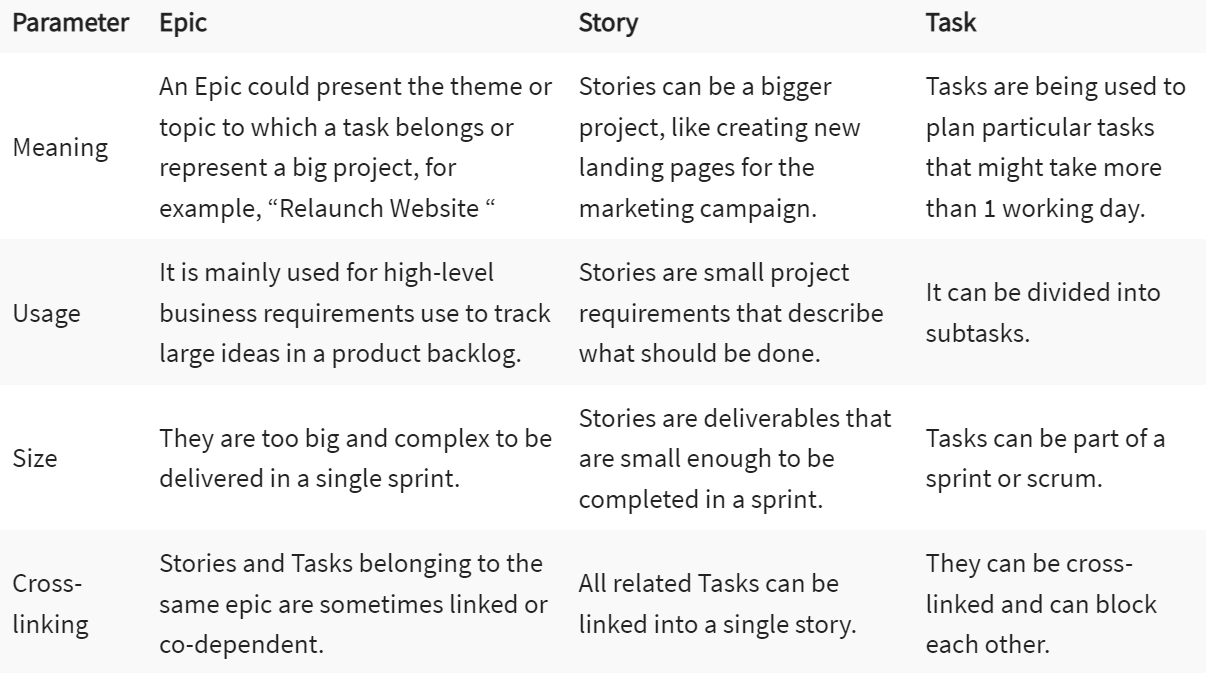
Story is a bigger element as compared to tasks, as stories are made of tasks. Features of Story are:

* The story is usually assigned to the project lead.
* Tasks are linked to stories.
* Stories can be completed in a few sprints.

#### Epic:

Epic is a larger element that is broken down into stories and tasks. Epics have the following characteristics:

* Epics represent the collection of tasks for a bigger project.
* Epics are part of multiple sprints.
* Epic comprises stories and tasks that can be linked.



### 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

### What is story points/efforts/ scales?

It is used to discuss the difficulty of the story without assigning actual hours. The most common scale used is a Fibonacci sequence ( 1,2,3,5,8,13,….100) although some teams use linear scale (1,2,3,4….), Powers of 2 (1,2,4,8……) and cloth size (XS, S ,M,L, XL).

Story points are usually estimated before a sprint planning meeting. This is when a team determines how much work they can complete in an upcoming sprint.

**Here are some story point estimates:**

* 0.5 story points: Anything under 4 hours of work
* 1 story point: ½ day worth of work (4 hours)
* 2 story points: 1 full day (8 hours)
* 3 story points: 2 days (16 hours)

**Teams assign story points relative to:**

* Work complexity
* The amount of work
* Risk or uncertainty

### 21) Explain what is Velocity in Agile?

Velocity is a metric that is calculated by addition of all efforts estimates related with user stories completed in an iteration. It figures out how much work Agile can complete in a sprint and how much time will it need to finish a project.

Story points also provide an internal team metric known as velocity. Velocity is the number of points implemented per sprint

### Mention what is the difference between Scrum and Agile?

* **Scrum**: In the scrum, a sprint is a basic unit of development. Each sprint is followed by a planning meeting, where the tasks for the sprint are identified and estimated. During each sprint, the team creates finished portion of a product
* [**Agile**](https://www.guru99.com/agile-testing-a-beginner-s-guide.html): In Agile, each iteration involves a team working through a full software development cycle, including planning, design, coding, requirement analysis, unit testing, and acceptance testing when a product is demonstrated to stakeholders

In simple words, Agile is the practice and scrum is the process to following this practice.

### 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.