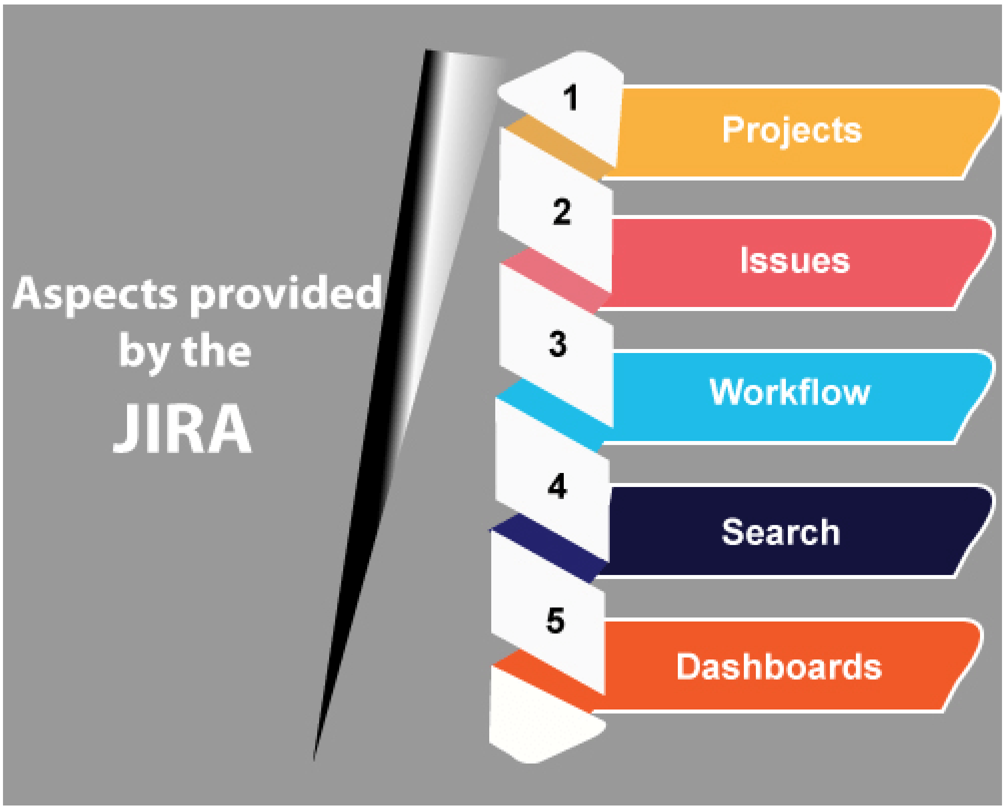
What is JIRA?

JIRA is a software testing tool developed by the Australian Company Atlassian. It is a bug tracking tool that reports all the issues related to your software or mobile apps.

JIRA is based on the Agile methodology and the current version of the Jira is 6.

The following are the useful aspects provided by the Jira:



- **Projects:** It is used to manage the defects very effectively.
- **Issue:** It is used to track and manage the defects/issues.
- **Workflow:** Processes the Issue/Defect life cycle. Suppose we have a business requirement, we create the technical design and from the technical design, we create the test cases. After creating the test cases, coding is done, and then testing is performed on the project. This design workflow is possible by using Jira.
- **Search:** Find with ease. Suppose we have done with a project at the beginning of the December and its version is 1.0. Now, we move to version 1.1 and completed at the end of December. What we are doing is that we are adding new versions. Through Jira, we can get to know that what happened in the

earlier versions, how many defects occurred in the earlier projects and the learning we achieve from the earlier projects.

- **Dashboards:** Dashboard is a display which you see when you log in to the Jira. You can create multiple dashboards for multiple projects. You can create the personal dashboard and can add the gadgets in a dashboard so that you can keep track of the assignments and issues that you are working on.

Why JIRA

JIRA tool is used because of the following reasons:

- **Plan, Track and Work Faster**
  JIRA is a bug-tracking tool mainly used to track, organize, and prioritize the bugs, newly added features, improvements for certain software releases. Projects are subdivided into issues and issues can be of multiple types such as bug, new feature, improvement, and documentation tasks. When the release date of software comes near, then software developers need to focus on the remaining issues which are to be fixed before the specified date. It also becomes difficult for the QA to maintain the status of the documentation, i.e., sometimes it becomes hard to keep track of everything. JIRA is a good choice for handling the above issues. It enables software developers to track issues and improvements. It manages the projects as well as maintain the technical documentation.
- **The main source of information**
  JIRA is the primary source of information for the next software release. On JIRA, the whole team of the software developers can plan for the new features which are to be added and bugs to be fixed in the next release. It also helps the QA team in writing the technical documentation. Through JIRA, the QA team can check the status of each feature that is newly added by the software developers, and according to that, they can plan how to document for the new version.
- **Organize the documentation tasks**
  JIRA tool is used to organize the documentation tasks. It is useful in grouping the multiple tasks by using the component functionality, and even you can create your own documentation. In this way, you can create a structured way of documentation.

- **Track the progress of our documentation** It is a very useful tool in tracking the progress of our documentation. JIRA tool provides a very important feature, i.e., pie chart macro. In the pie chart macro, you can view tasks such as Open tasks, Closed tasks, Resolved tasks.
- **Helps to meet the deadlines of a documentation release.** You can define the specific due date or deadline for the release of documentation, and even you can configure the JIRA tool with the notifications so that you can finish your documentation in time.
- **Measures the time spent on documentation** JIRA tool does not have the default functionality for measuring the time spent on documentation. JIRA tool is bundled with the Tempo Timesheets, which measures how much time has been spent on the documentation.
- **Provides feedback faster** JIRA tool provides the Confluence pages where you can connect to the issues in just a few clicks. If something needs to be updated, then you can create the issues directly from the Confluence page.

JIRA Workflow

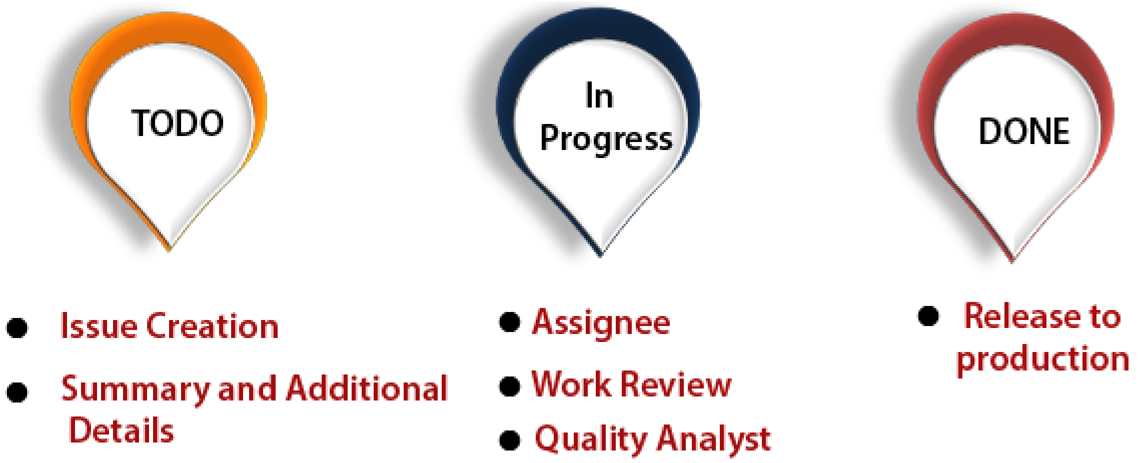
Workflow is a set of activities which are performed to track the status and the transition of an issue during the lifecycle of an issue.

Where **transition** represents some work in the form of link between the two statuses when an issue moves from one status to another.

Status: Status determines the impact of the work on the issue which is filed by the tester.

In Jira tool, following are the phases that occur in the workflow:

- **TODO state**
- **In Progress state**
- **Done state**



There are two activities performed in the TODO state:

- **Issue creation**
  When the tester finds a defect, then they log the defect in Jira tool. Once the defect is logged in a Jira tool, the unique ticket identification number is generated by a Jira tool. This process is known as issue creation.
- **Summary and additional details**
  Issue creation requires some additional information which is to be added or updated to an issue such as issue description, priority, severity, components impacted, subtasks, upload screenshots, email history, etc. After adding all the details to an issue, Jira tool assigns the status as a TODO state.

There are three activities performed in the In Progress state:

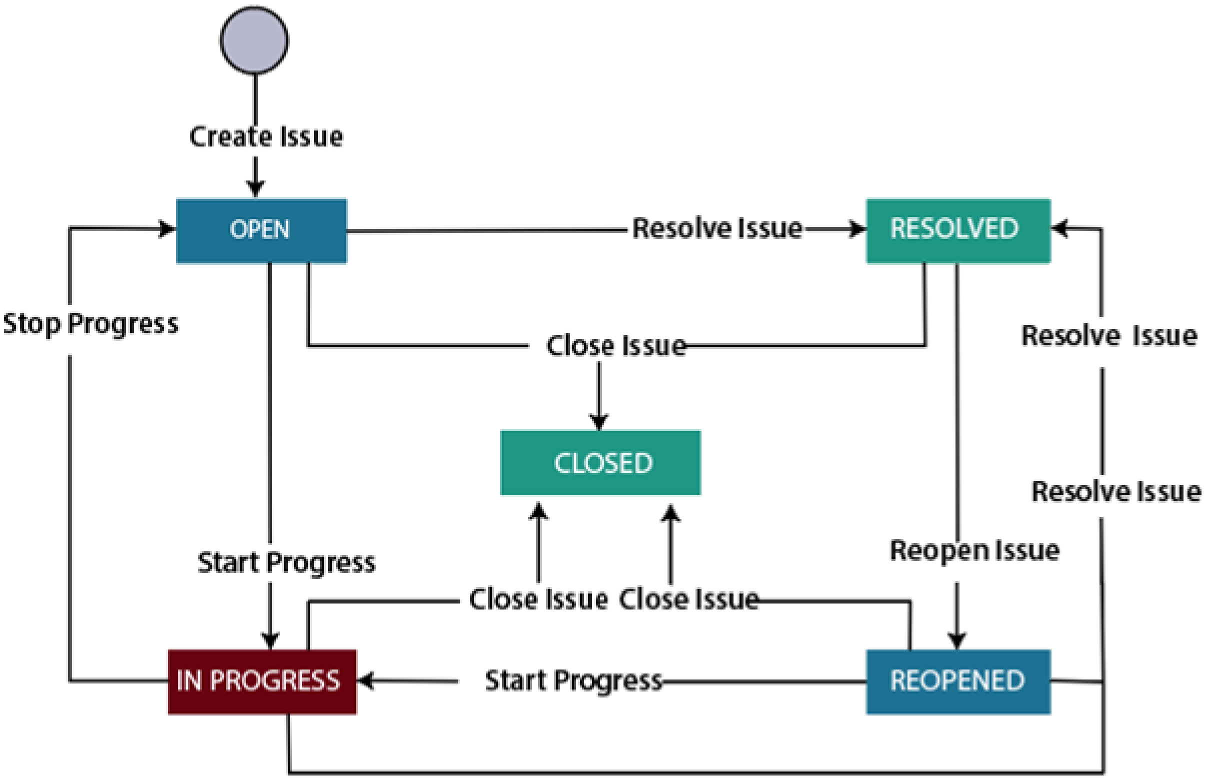
- **Assignee**
  When the issue is created, then it is assigned to a person or a team. After assigning the issue to a person, then the status changes to **In Progress** state.
- **Work Review**
  Work on the issue is first reviewed and monitored by the issue reporter, assignee, and other project's management folks. When the developer removes an issue, then it is first unit tested by the developer, and then the code review team reviews it. During the work review activity, the status remains in the **In Progress** state.
- **Quality Analysis**
  After code review, the changes are verified by the quality analysis team, which includes regression testing. If the QA team finds any problem in the changes,

then the issue is reassigned to the developer otherwise QA team will close the issue which states that the issue has been fixed. During the Quality Analysis, the status remains in the **In Progress** state.

One activity is performed in the **Done** state:

- **Release to production** Once the product has been developed and tested, then the product is released in the market. When the product is released in the market, then the status is changed to **Done** state.

JIRA Workflow can also be referred to as a defect lifecycle. The defect lifecycle is shown below:



- **Open issue:** When the issue is created, then the issue is assigned to the software developer, and they start working on it.
- **In Progress issue:** The software developers start working on the issue.
- **Resolved issue:** Issue is resolved by the software developer but waiting for the verification by the tester. If the verification is successful, then the issue is closed; otherwise, the issue gets reopened.

- **Reopened issue:** When the software developer did not resolve the issue as per the requirements, then the issue comes in a reopened state.
- **Close issue:** When the software developer correctly resolves the issue and verified by the software tester, then the issue will be closed.

Jira Scrum Board

Jira Scrum Board is a tool used to unite the teams to achieve a single goal and incremental-iterative delivery.

Functions of Scrum Board

- **Increase communication and transparency**
  Jira Scrum board is the single source through which all the work of a team can be accessed by a team member at any time. This increases communication among the team members and transparency.
- **Promote sprint planning and iterative development**
  The heart of the Scrum framework is a sprint, which is a fixed amount of time for teams to build a releasable product increment. Jira scrum board is mainly designed so that the teams can organize their work within the sprint timeframe.
- **Improve team focus and organization**
  Sometimes teams forget the project deadlines, and they are over-committed on their workload. Jira Scrum Boards provide transparency to the team's work by dividing the work into multiple stages and providing the burndown and velocity reports.

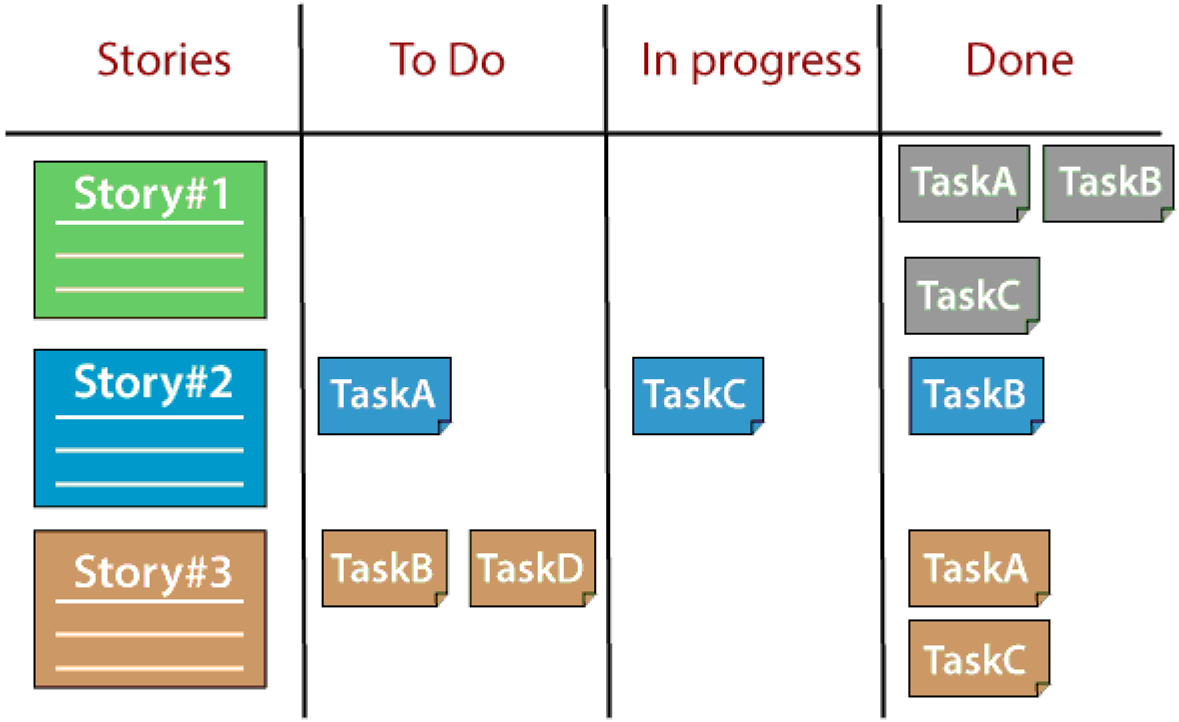
Following are the important terms related to a Scrum Board:

- **Sprint**
  A sprint is a time-boxed iteration of time which is mainly of 2 weeks. When the sprint is completed, then it produces the most valuable, useable, and releasable product in the market.
- **Backlog**
  The Backlog is owned by the Product owner. Product Backlog is a list of features, defects, enhancements, experiments that are to be considered in the product.
- **User story**
  - A user story is the smallest unit of work in an agile framework. A User story is a goal, not a feature which is expressed in terms of the user's perspective.

  - The purpose of a user story is to show how a piece of work will deliver a particular value back to the customer.
  - User stories are the sentences written in simple language to provide the desired outcome.
  - It should be in a granular format so that it can be delivered within a sprint. Sprint is nothing but a time-bound iteration to deliver the working software.
  - User story comes under the epic, and epic is a large chunk of work, user stories come under the epic. Some user stories are taken from the epic and placed in a particular sprint. This cycle goes on until and unless all the user stories within a sprint are completed.
- **Issue**
  An issue is also known as user story. In scrum board, an issue contains all the tasks, dependencies, and other relevant information required to perform a single piece of work.
- **Epic**
  Epic is a large chunk of work. It is basically a large user story which can be broken into a number of smaller stories. All the user stories in the epic might take several sprints to complete.
- **Swimlane**
  Swimlane is a means of categorizing of issues in an active sprint of a scrum board or a kanban board. It helps you to distinguish the tasks of different categories, such as users, workstreams, etc.

Layout of the Scrum Board

The Scrum Board is a physical board on which the user stories present in the current sprint backlog are displayed.



Stories	To Do	In progress	Done
Story#1			TaskA TaskB TaskC
Story#2	TaskA	TaskC	TaskB
Story#3	TaskB TaskD		TaskA TaskC

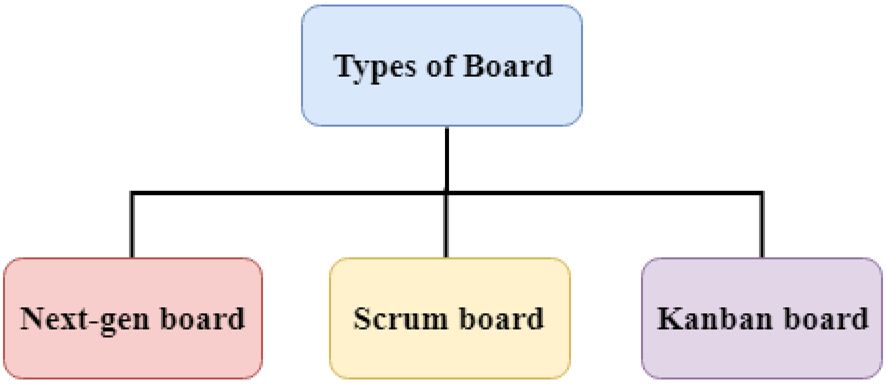
Scrum board is divided into columns such as:

- **Stories:** This column contains all the user stories available in the current sprint backlog.
- **TODO:** This state contains the subtasks of the stories on which the work has not started on.
- **In Progress:** This state contains all the tasks on which the work has started.
- **Done:** This state contains all the tasks on which the work have been completed.

What is board?

A board displays all the issues that occurs within the project, providing you a flexible way of viewing, managing, and reporting the progress on work.

There are three types of boards that exist in Jira:



- **Next-gen board**
  This board is useful for those who are new to the agile. It is a very simplified, straight-forward, and streamlined board.
- **Scrum board**
  This board is useful when teams work on sprints that includes a backlog.
- **Kanban board**
  Kanban board is an agile project management tool designed for the visualization of work, limit work-in-progress, and maximize efficiency.

Issue types

Teams work together to break down the pieces of work into issues. Issues can represent software bugs, a project task, subtasks, or any other project work. Jira project comes with the default issue type, and you can also create your own issue type.

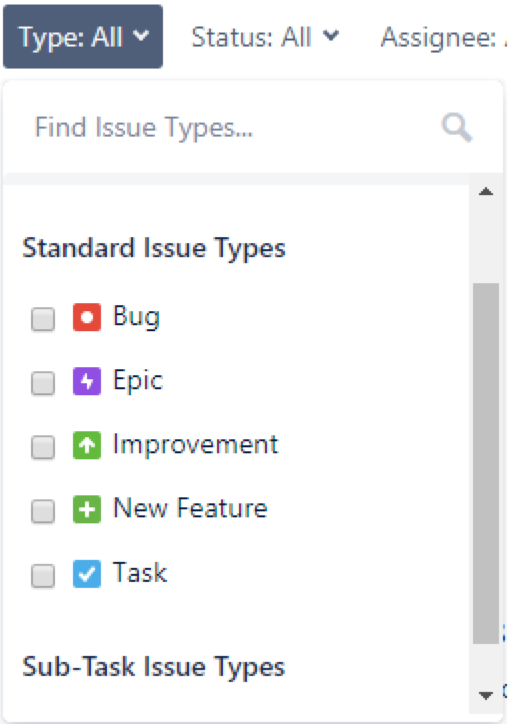
An issue type is a generic name for the unit of work. In Jira, we have different units of work in a project, and issue types field differentiate this.

When you create an issue, then you need to specify the issue type. You can even change the issue type after creating the issue.

Each issue has an icon associated with it to identify the issue type.

Custom Issue type: It is also used to create the customization issue type that provides the flexibility to the team to create their own issue types according to their projects

The below screen shows the list of Issue types:



Standard Issue Types

□ Bug

□ Epic

□ Improvement

□ New Feature

□ Task

Sub-Task Issue Types

There are three types of Issue types:

- **JIRA Core Default Issue types**
  - **Task:** The task is a work item done by the team but not connected directly to the user's requirements. For example, to upgrade the version of a product used by the teams.
  - **Subtask:** It is a part of another issue. It is used to break an issue into different pieces of work. While creating an issue, sub task issue is not given in the issue type field drop-down as it contains some parent issue, so we can say that subtask issue cannot be created independently.
- **JIRA Software default issue types**
  - **Story:** It is a requirement from the user's perspective.
  - **Bug:** It is a flaw in a product that needs to be fixed by the developers. It can be tracked with its own issue type to differentiate from other types of work.
  - **Epic:** An epic is a big issue that contains other issues.
- **JIRA Service desk default issue types**
  - Incident
  - Service Request

- Change
- Problem

Issue types are used because of the following reasons:

- They support multiple work items. Usually, teams contain multiple work items, and issue types are used to differentiate these work items.
- Each Issue type can have different fields, screens, and workflows. For example, the bug appears on the top of the project board.
- You can report issue types separately. Issues are categorized by issue types; for example, you want to report the progress of your work of the previous week.

Subtask

- Subtask is an issue type that must have a parent issue.
- To create a subtask, click on the create subtask icon for the parent issue.

Scrum

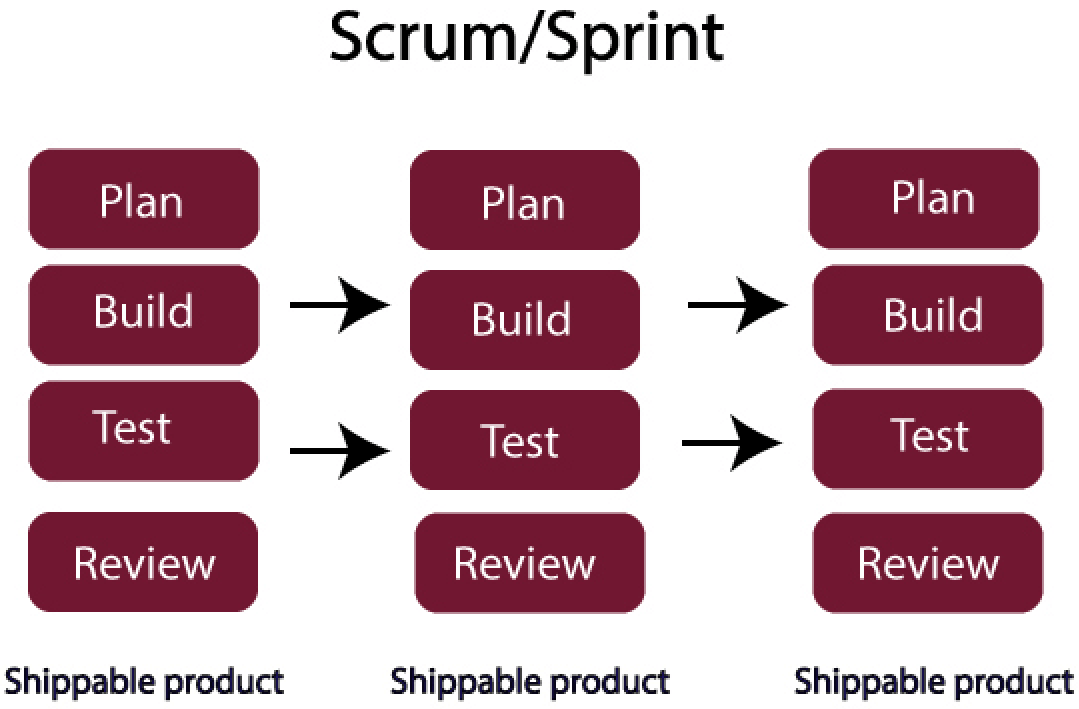
We have studied the Agile methodology where Agile is a set of beliefs which should be followed to develop the software development project. On these beliefs or values, there is many models have developed, and in which one of the models is a **scrum**.

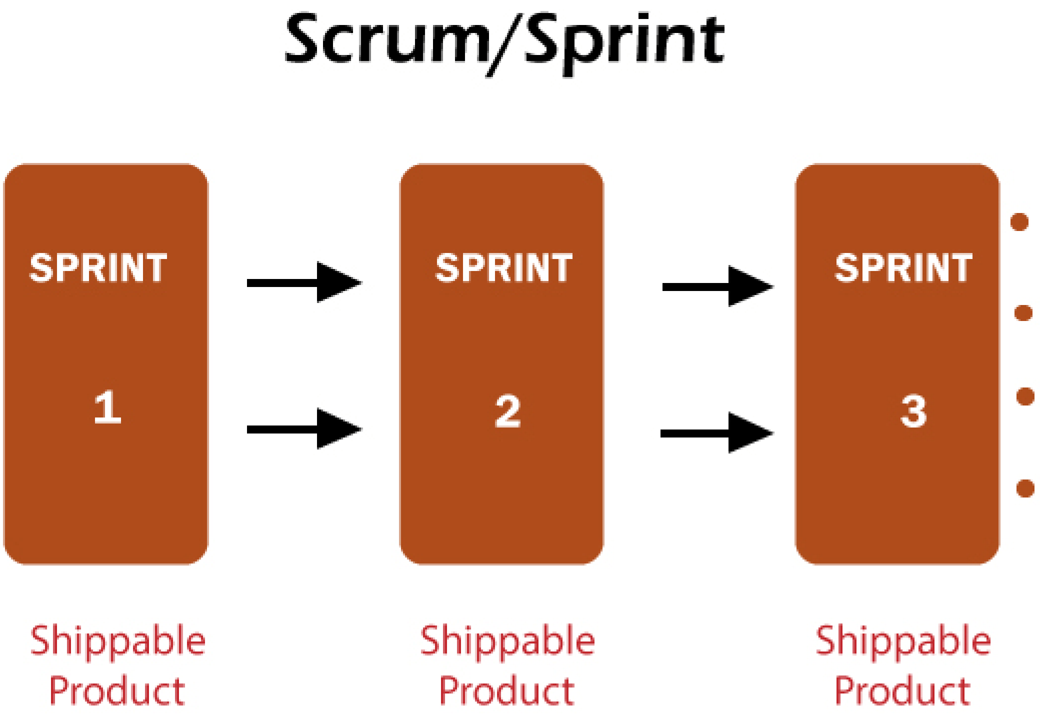
Before going to deep in Scrum, you should know the meanings of some basic terms:

- **Scrum:** Scrum is an agile framework that helps you to organize, iterate, and continue the same project that you are working on. In scrum, a product is built in the series of iterations known as sprints or parts.
- **Sprint:** Sprint is a time-boxed period in which the scrum team needs to finish the set amount of work. Each sprint has a specified timeline, i.e., 2 weeks to 1 month. The scrum team agrees with this timeline during the sprint planning meeting.
- **Scrum Master:** Scrum Master is defined as a facilitator or servant-leader to the Scrum development team. Scrum Master must ensure that scrum principles are followed.

- **Scrum development team:** A scrum development team is a collection of individual members that includes developers, QA, and scrum master. It decides and provides the effort estimate. The recommended size of the scrum team is between 5 and 9 members.

How does Scrum work

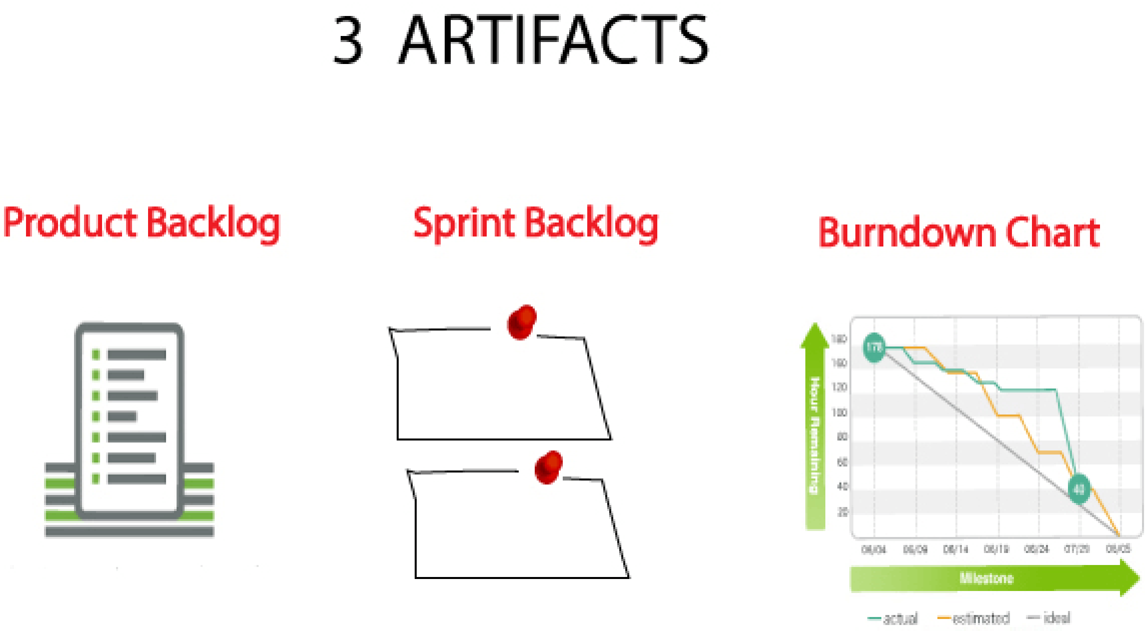




Artifacts of Scrum

The documentation and stuff which are prepared in scrum are known as Artifacts.

Following are the artifacts of Scrum:



- **Product Backlog**
  Product Backlog is a collection of activities that need to be done within the project. When we want to develop software, then we need to perform the 'n' number of activities. For example, we need to develop the e-commerce website then we have to do the 'n' number of activities such we need to create the login page, payment system, cart system, etc. and these 'n' number of activities which are needed to develop the software is known as the product backlog.
- **Sprint Backlog**
  We know that in a scrum, we break the scrum into 'n' number of sprints and the objective of a sprint is to bring the small functionality of the software and ship it to the client for demo. In Product backlog, we have to do all the activities which are required to develop the software while in the sprint backlog, a small set of product backlog activities are performed within that sprint. The 'n' number of sprint backlogs is equal to the 1 product backlog.
- **Burndown chart**
  Burndown chart is the outcome of the sprint, which shows the progress in a sprint. After each sprint, we need to examine the progress of each sprint. The burndown chart tells how you are working on the sprint. In the burndown chart, the graph starts from some time, i.e., where the activity gets started, and at the end of the sprint, the graph reaches to zero where the activity ends. It is generally an inclined line from top to bottom.

Scrum Roles

There are three scrum roles:

- **Product Owner**
  There is a client who wants to develop his software, so he approaches to the company who can develop his software. What does the company do? The Company assigns a role, i.e., Product Owner. Product Owner is the person who communicates with the clients understands their requirements. Product Owner is the responsible person from the company for software development.
- **Scrum Master**
  During the sprint, Agile says that the team should meet together once daily.

When the team is following scrum means that they are conducting meetings daily for 10 to 15 minutes. This meeting is known as a scrum meeting. Scrum Master is the person who handles the scrum meeting.

- **Team**
  The team comprises of persons who work on the project. It can be developers, testers or designers. When we talk about Agile or Scrum then we talk about the team, we do not talk about developers, or testers as an individual. Agile says that developers can work as a tester or testers can work as a developer when the need arises.

Scrum Ceremonies

Let's look at the following Scrum Ceremonies:

- **Sprint Planning**
  Scrum consists of a number of sprints which have a different set of modules used to deliver the software. Before starting the sprint planning, we have a meeting known as sprint planning, and in sprint planning, we discuss what we are going to do in a sprint. In sprint planning, product owner discusses about each feature of a product and estimates the effort involved by the development team.
- **Daily Scrum**
  In Scrum, meetings are conducted daily for 15 minutes by Scrum Master, where Scrum Master is the person who manages the meeting. Meeting consists of scrum master, developers, testers, designers, product owner, the client where product owner and client are optional.
- **Sprint Review**
  After the completion of each sprint, the meeting is conducted with a client in which a product is shown to the client for demo and team discuss the features they added in the project.

Kanban Methodology

Kanban is the most popular agile framework after Scrum for software development. It provides the real-time and transparency of work. In Kanban board, all the tasks are visible that allows the team members to see the state of every task at any time.

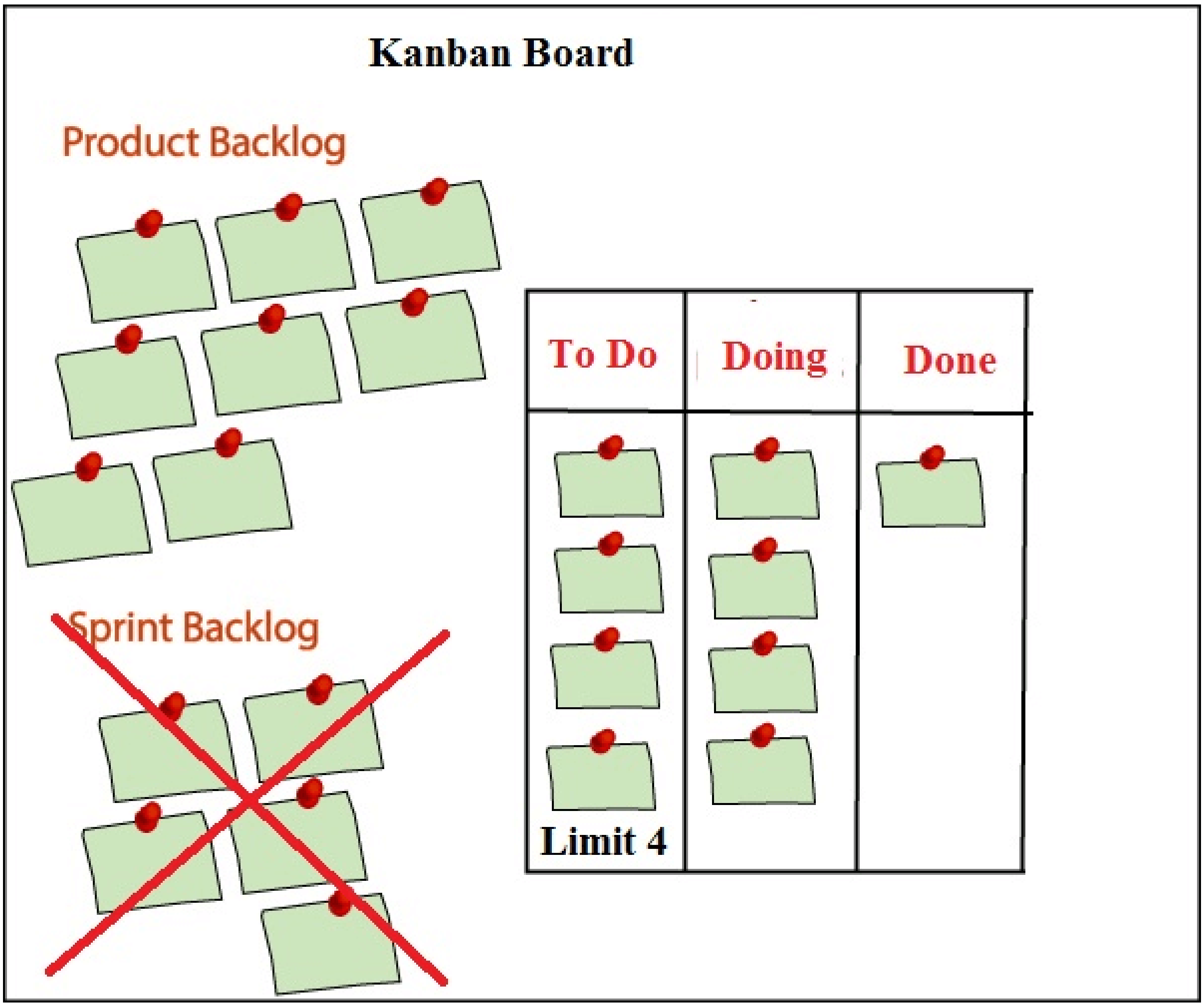
Characteristics of Kanban methodology:

- **Flexibility**
  In Kanban methodology, a team focusses on the work that is '**in progress**' state. Once the team completed its task, then it pulls the next first task of the product backlog. A product owner reprioritizes the tasks or made changes in a product backlog outside the team, so this will not disrupt or impact the team. A product owner keeps the most important task on the top of the product backlog, so the development team assures that they will produce the most valuable output. In Kanban, we do not need to do the fixed-length iteration as we did in the scrum.
- **Minimize time cycles**
  The cycle time is the amount of time taken by the work to travel from the moment it gets started to the moment it gets shipped to the customers. An overlapping skill set can minimize the cycle time. In this, developers not only write the code but can also test the code whenever required. This type of sharing skills means the team members can take the heterogeneous work which optimizes the cycle time.
- **Visual metrics**
  Visual metrics is a way of improving team efficiency and team effectiveness. A visual metric is shown through charts, and team members can view the data in charts, and can spot the issues arises in their process. The main goal of the visual metric is to reduce the amount of time taken by the issues to move through the entire process.
  There are two types of charts used by the kanban team:
  Control charts: It shows the cycle time taken by each issue.
  Cumulative flow diagrams: It shows the number of issues present in each state.
- **Continuous delivery**
  The main aim of continuous delivery to deliver the product with low risk rapidly. The transition from the agile methods to continuous delivery move the two-three weeks sprint to the Kanban methodology. Both Kanban methodology and continuous delivery complement each other by delivering the product to the customers faster. Software development teams are used to develop, test, and review the new features in a continuous manner. Therefore, we can say that Kanban is a continuous flow methodology.

Kanban Board

Kanban board is a tool used to visualize the work and limit work-in-progress.

As in scrum, we are taking some activities from a product backlog and adding in a sprint backlog. However, in Kanban, we do not have sprint, so sprint backlog activity will not be performed. This is the main difference between scrum and Kanban that scrum contains sprint backlog while kanban does not contain the sprint backlog.



Kanban board is a tool used to visualize the work and limit work-in-progress.

As in scrum, we are taking some activities from a product backlog and adding in a sprint backlog. However, in Kanban, we do not have sprint, so sprint backlog activity will not be performed. This is the main difference between scrum and Kanban that scrum contains sprint backlog while kanban does not contain the sprint backlog.

Kanban board consists of three states:

- To Do
- Doing
- Done

When the project is started, then we put all the activities from the product backlog to the '**To Do**' state. When the team member starts working on an activity, then that activity is put in a '**Doing**' state, and when the activity is placed, then it is placed in a '**Done**' state.

From the Kanban board, one can get to know which activities have been done and which activities they need to develop.

One of the most important features of the Kanban board is a **Limit** option. In the above figure, we have eight tasks in a product backlog and limit set is 4. At a time, it will take only four tasks in a '**To Do**' state, and if any of the tasks come in a '**Doing**' state, then one more task from the product backlog will be placed in a '**To Do**' state. In this way, we can set the **limit** depending on the availability of the resources.