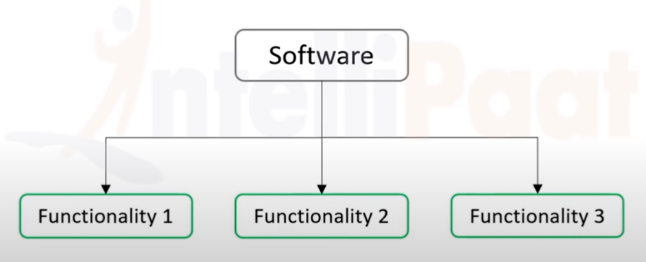
**Jira Tutorial for Beginners**

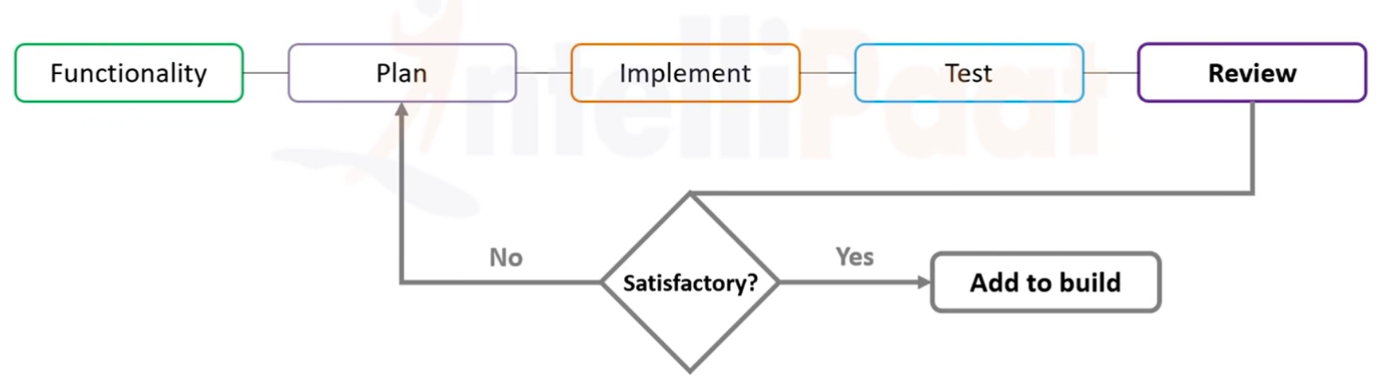
* **The Agile Process:** It is a ***time-boxed***, ***iterative approach*** to software development, ***to deliver*** the ***product incrementally***, instead of all at once.
* A software consists of several functionalities.



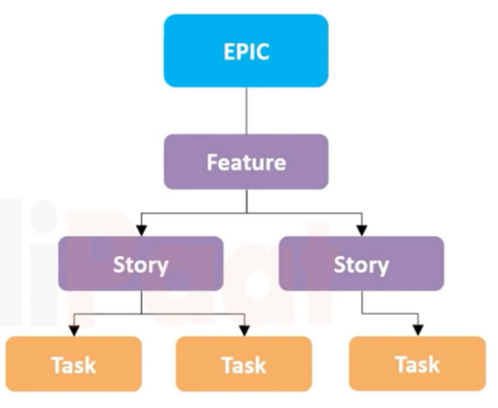
* In agile process, each functionality is considered as an individual project.



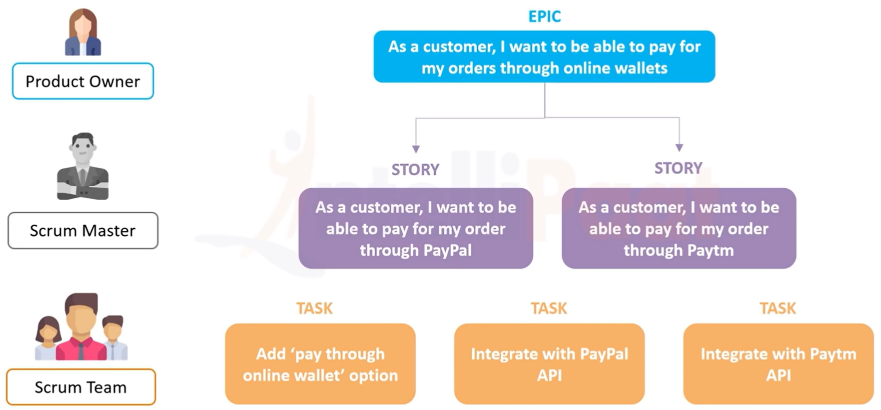
* For each functionality, the following steps are followed –
* Plan
* Implement
* Test
* Review



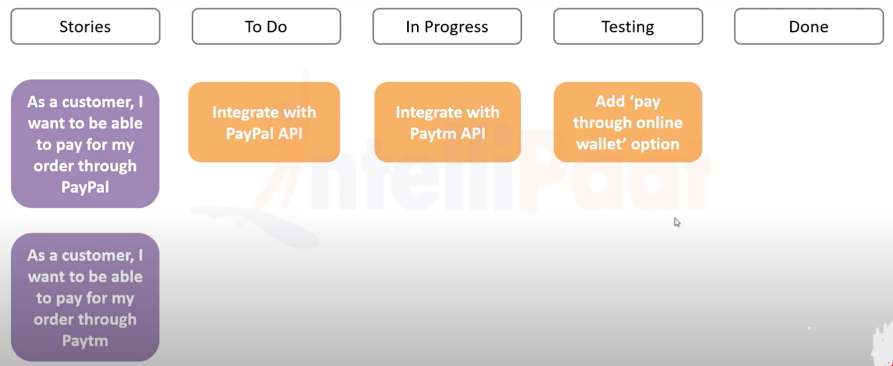
* Advantages of Agile process –
* An ***unsatisfactory functionality*** ***cannot stop*** the ***development*** of the software.
* ***Each*** of the ***functionalities*** can be ***developed in parallel***, or sequentially, based on requirements. Hence, it ***saves time***.
* **Agile Terminologies Used in Jira –**
* ***Epic***: Abstract view of what we exactly need.
* ***Features***: Epic is broken down into features that are required to fulfill that epic.
* ***Stories***: Non-technical components that are required to fulfill the features or epic.
* ***Tasks***: Technical sub-components that describe the technical aspect and constraints for developers to work on the problem in hand.



* **Scrum:** It ***describes*** the ***hierarchy of employees*** working for a software development company ***and how they operate*** to achieve agile development.
* **Agile Scrum Roles –**
* **Product Owner:** The product owner is responsible for owning and the successful completion of the project. The responsibilities include –
* Understanding what the customer wants.
* Convey the message to the development team through epic.
* Creates and maintains product backlog.
* Release management.
* **Scrum Master:** Intermediary between product owner and the development team. The responsibilities include –
* Ensures that dev team follows agile methodology.
* Makes sure that stories or epics are broken down into appropriate tasks.
* Manages product backlog.
* Ensures project completion.
* **Scrum (Development) Team:** Consists of software developers and quality assurance people. The responsibilities include –
* Break down work into subtasks.
* Follow agile methodology.
* Ensure timely delivery with quality.
* Communicate their progress with scrum master.



* **Scrum Board:** It has many columns, such as –
* **Stories:** It consists of the stories required to be worked on.
* **To Do:** It consists of the tasks that have not been started as yet.
* **In Progress:** It consists of the tasks that are currently being worked on.
* **Testing:** It consists of tasks that are currently being tested.
* **Done:** It consists of tasks that are developed and tested fully.





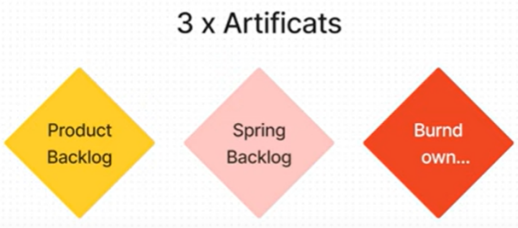
* **Sprint:** It is between 1 to 4 weeks, allotted to complete the assigned tasks.
* **Scrum Meetings:** 15-minute meetings daily to discuss the previous day’s complete/ incomplete tasks and the current day’s expected tasks.
* **Jira:** Project Management tool, developed by Atlassian. Two of its most important features are –
* Bug tracking
* Agile project management
* **Why use Jira?**
* Creating backlogs
* Creating sprints
* Creating tasks
* Updating status on scrum board
* Managing releases
* Code integration
* Managing workflows

**Introduction to Agile Methodology**

* **Downtime:** A ***specific time frame*** allocated ***to deploy or update changes*** for a software product in a real-time environment. It is a part of the traditional waterfall model. ***Used*** to ***avoid disturbance*** in the ***workflow*** of an organization. ***Applying changes*** in a waterfall model product ***without downtime*** may ***produce*** irrelevant results or ***product failure***.
* ***In Agile*** methodology, this ***downtime*** can be ***drastically reduced***.

**What is Scrum?**

* 3 different artifacts of the Scrum workflow –
* **Product Backlog:** Features of the product, e.g. login, admin dashboard, etc.
* **Sprint Backlog:** Turning these features into user stories that can then be developed. Preferably, add 100% ± 15% of the tasks possible to complete in one sprint.
* **Burndown Chart:** Scrum board.



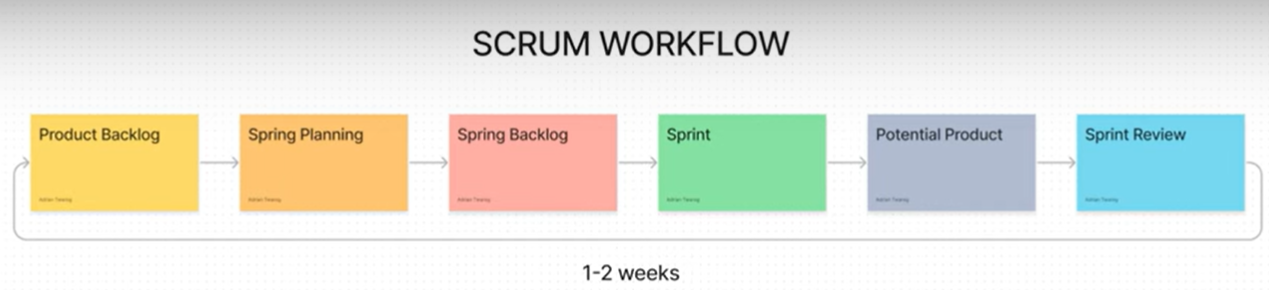
* 3 different ceremonies in Scrum workflow –
* **Sprint Planning:** The phase of planning the project sprint of 1-4 weeks. Higher priority tasks are added first.
* **Daily Scrum:** Scrum meeting.
* **Sprint Review:** To see if everything that was planned has been completed.



* A user story needs to have the following format –

“As a \_\_\_\_\_\_\_, I need \_\_\_\_\_\_, so that \_\_\_\_\_\_.”

For example, “As a user, I need to reset my password, so that I can still login even when I forget my credentials.”



**Sprint Planning with Jira Part 1**

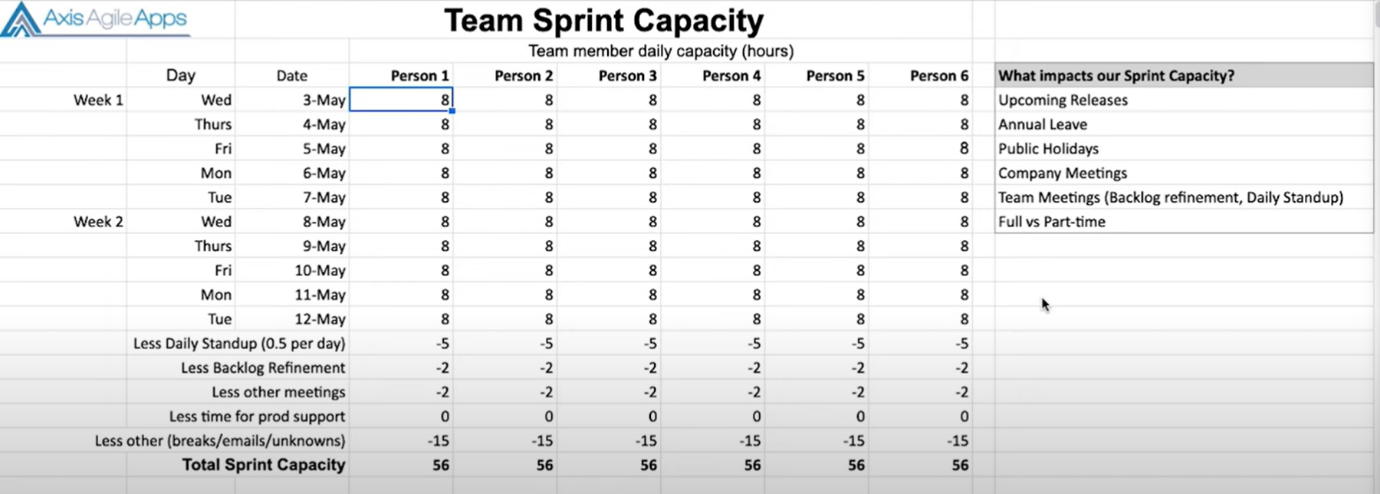
* 3 questions that need answering –
* ***Why*** is the sprint valuable?
* ***What*** can be done in this sprint?
* ***How*** will the work get done?
* Prerequisites for sprint planning –
* Sprint-ready product backlog items, which are –
* well-defined
* well-understood
* right size
* Team capacity determined: How much workload can the team handle during a single sprint?

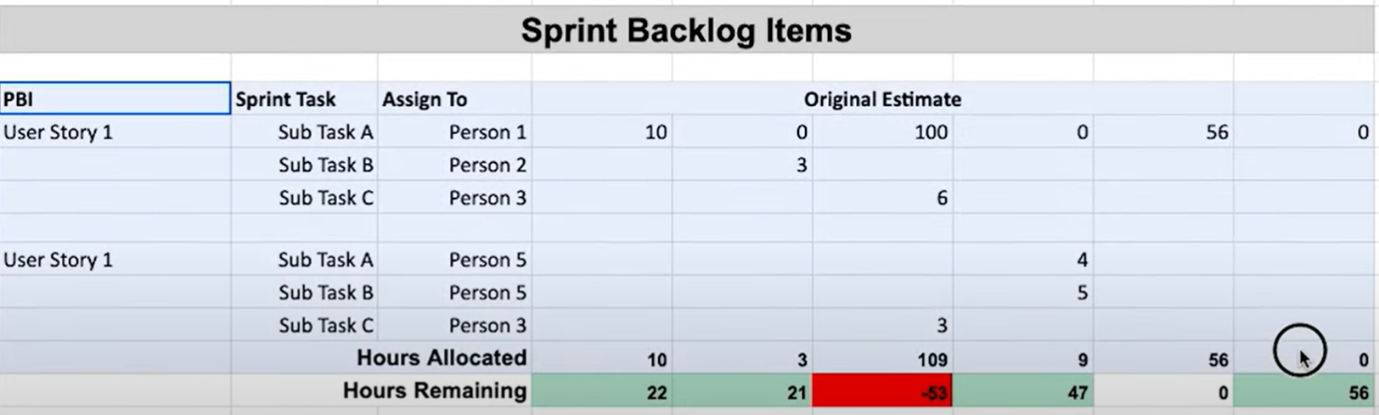
**Sprint Planning with Jira Part 2**

* Two methods to perform sprint planning –
* ***Velocity or story points***
* ***Time***
* For calculating velocity in Jira, keep the following things in mind –
* ***More sprints the better*** estimated velocity
* ***Consistent sprint length***
* ***Stable teams*** (team members are same and not changing frequently)
* ***Factors*** that may ***affect*** the ***sprint***, like –
* team member on ***leave***
* public ***holidays***
* company ***events***
* There are ***two parts*** in the ***sprint planning*** process –
* Ask product owner the final questions to ***clarify priorities***.
* ***Decide*** ***what can get done*** during this sprint.
* For every product backlog item, decide –
* **Technical implementation:** What’s involved and who’s doing what work.
* **Dependencies:** How the item may be dependent or associated with other items. Is third-party integration required? Is outside help required?
* **Risks:** E.g. working with a new technology, a team member remaining absent, etc.
* Let’s say the team’s capacity is 22 points, but the total points added to the sprint is 21 only. The spared time can be used for a number of activities, such as –
* Pay down ***technical debt***, like code-cleanups.
* Work on ***process improvements***, like automating some parts of the implementation or testing phase.
* Perform ***product backlog refinements***.
* Have an ‘***Innovation Day***’. An entire day where team members build whatever they want.
* ***Pros*** of sprint planning using velocity –
* ***Quicker*** than time-based approach.
* ***Greater comfort with team*** members.
* ***Cons*** of sprint planning using velocity –
* ***Requires*** a ***few sprints*** initially to ***attain*** a ***reliable velocity***.
* ***Not precise***. Junior developer will take more time to do the work when compared to a senior developer, but that is not evident in the story points.
* ***Harder to understand***.

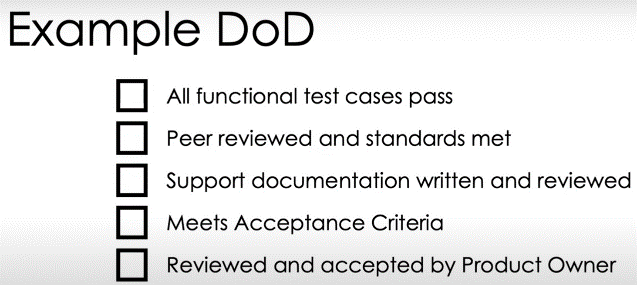
**Sprint Planning with Jira Part 3**

* ***Some teams never complete their sprint plan***. This is ***due to bad estimates*** on the team ***capacity***.
* ***Incomplete sprint plans*** may result in –
* ***Demotivated employees***
* From ***stakeholders’ point of view***, when the team never complete their said tasks sprint after sprint, it ***reflects poorly on the team***.





* Tips for ***creating sub-tasks*** in time-based planning –
* Sub-tasks ***shouldn’t be too granular***, e.g. don’t create 15 minutes sub-tasks.
* ***Shouldn’t take longer*** than a day, e.g. stick a task to less than 8 hours.
* Follow the ***definition of done*** (DOD)



* ***Pros*** of time-based planning –
* ***Greater precision***
* More ***likely to complete sprint*** plan
* ***Improve estimation*** by tracking original estimate
* ***Cons*** of time-based planning –
* ***Takes more time*** to do
* Might be ***perceived as micro-managing***
* ***Discomfort with time estimation***
* Tips for teams that are uncomfortable with time-estimation –
* Compare to previous estimates
* Ensure Product Backlog items are sprint ready
* It’s for planning, not for punishment
* Erase all time estimates at the end of sprint planning