Understanding Agile

What is Agile?

- Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches.
- Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments.
- Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

What is agile project management?

- Agile project management is an iterative approach to managing software development projects that focuses on continuous releases and incorporating customer feedback with every iteration.
- Software teams that embrace agile project management methodologies increase their development speed, expand collaboration, and foster the ability to better respond to market trends.
- Here is everything you need to know to get started or refine your agile project management practices.

Sprint

- Fixed time period during continuous development cycle where team completes their task from product backlog.

```
https://python10sep.atlassian.net/jira/software/projects/AQUA/boards/1
```

What are stories, epics, and initiatives?

- **Stories**, also called “user stories,” are short requirements or requests written from the perspective of an end user.
- **Epics** are large bodies of work that can be broken down into a number of smaller tasks (called stories).
- **Initiatives** are collections of epics that drive toward a common goal.

What are agile user stories?

In Agile a user story is a short, informal, plain language description of what a user wants to do within a software product to gain something they find valuable.

User stories typically follow the role-feature-benefit pattern (or template):

- As a [type of user],
- I want [an action]
- so that [a benefit/value]

As the smallest unit of work in an Agile setting, user stories are a key tool in incremental development.

A user story is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

User stories are a few sentences in simple language that outline the desired outcome. They don't go into detail. Requirements are added later, once agreed upon by the team.

What is an agile epic?

An epic is a large body of work that can be broken down into a number of smaller stories, or sometimes called “Issues” in Jira. Epics often encompass multiple teams, on multiple projects, and can even be tracked on multiple boards.

Epics are almost always delivered over a set of sprints. As a team learns more about an epic through development and customer feedback, user stories will be added and removed as necessary. That’s the key with agile epics: Scope is flexible, based on customer feedback and team cadence.

Agile epic vs. story

An epic is **a large body of work that can be broken down into a number of smaller stories, or sometimes called “Issues” in Jira**.

Epics often encompass multiple teams, on multiple projects, and can even be tracked on multiple boards. Epics are almost always delivered over a set of sprints.

In a sense, stories and epics in agile are similar to stories and epics in film or literature. A story is one simple narrative; a series of related and interdependent stories makes up an epic.

The same is true for your work management, where the completion of related stories leads to the completion of an epic.

The stories tell the arc of the work completed while the epic shares a high-level view of the unifying objective.

On an agile team, stories are something the team can commit to finish within a one- or two-week sprint. Oftentimes, developers would work on dozens of stories a month. Epics, in contrast, are few in number and take longer to complete.

If your company was launching rockets into space, and wanted to improve the streaming service for your launches, you might structure your stories like the ones below.

Examples of an agile story:

- iPhone users need access to a vertical view of the live feed when using the mobile app.
- Desktop users need a “view fullscreen” button in the lower right hand corner of the video player.
- Android users need to be linked to apple store.

Story points vs. hours

Story points are **units of measure for expressing an estimate of the overall effort required to fully implement a product backlog item or any other piece of work.**

Teams assign story points relative to work complexity, the amount of work, and risk or uncertainty.

Traditional software teams give estimates in a time format: days, weeks, months. Many agile teams, however, have transitioned to story points. Story points are units of measure for expressing an estimate of the overall effort required to fully implement a product backlog item or any other piece of work.

Teams assign story points relative to work complexity, the amount of work, and risk or uncertainty. Values are assigned to more effectively break down work into smaller pieces, so they can address uncertainty.

Typically story points are assigned in fibonacci series formats (1, 2, 3, 5, 8, 13).

1	1 day
2	2 days
3	3-4 days
5	1 week
8	little more than week
13	2 weeks

5 Agile ceremonies

1. Sprint planning: These sessions are what initiate each sprint. Your team decides what work from the product backlog can be completed in the time period, who will complete what tasks, and how you'll be successful.
2. Daily scrum (or daily stand-up): These daily check-ups help teams stay on track and mark progress. Each morning, team members discuss what they worked on yesterday, what they're doing today, and what's blocking them from moving forward.
3. Sprint review: At the end of each sprint cycle, teams meet to demo what they've shipped and get early feedback from stakeholders. These review sessions can be informal 'show and tell' sessions or more formal meetings.
4. Sprint retrospective: At the end of each sprint cycle, the team also meets to discuss what's working (or not) on a process level, including blockers, communication issues, or a lack of resources.
5. Product backlog refinement: While not always listed as an 'official' ceremony, product backlog refinement is a continuous process of cleaning up, adjusting, and updating your task and feature list. Some teams set specific times to go through and clean up the backlog, while others do it continuously throughout each sprint.