Mphasis Tutorials point reading materials Agile

Contents

[References 7](#_Toc47102977)

# AGILE – PRIMER

## What is Agile?

It is a SDLC where the software is built incrementally, with each incrementations happening in short iterations of 1-4 weeks. This means that as the needs of the client changes or becomes clearer or more specific, the development process can keep up with the changes, and the requirement plan can be adjusted accordingly.

Instead of a single-pass development of 6 to 18 months where all the requirements and risks are predicted upfront, Agile adopts a process of frequent feedback where a workable product is delivered after 1 to 4-week iteration.

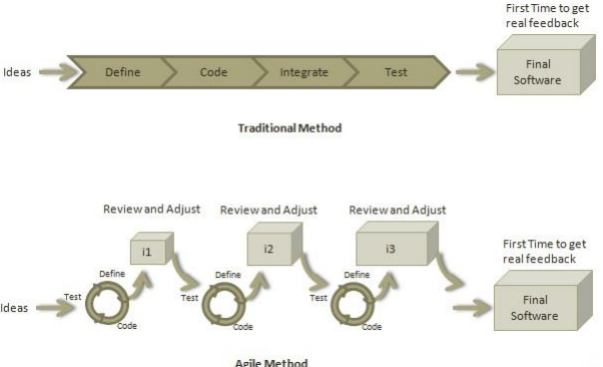


Figure Agile Method

## Roles in Agile

There are 2 main leadership roles in an Agile team

### Scrum Master

Team leader, helps the team to follow scrum practices. Works as an interface between the technical and business aspects of the team. Ensures close cooperation between members, and ensures that agile ‘Inspect and Adapt’ practices are leveraged properly:

1. Daily stand-ups
2. Planned meetings
3. Demo
4. Review
5. Retrospective Meetings

### Product Owner

They drive the product from the business perspective. Their job to ensure the product satisfies all the business requirements. Their responsibilities are:

1. To define the requirements and prioritize their values.
2. To determine the release date and contents.
3. To take an active role in iteration planning and release planning meetings.
4. To ensure that team is working on the most valued requirement.
5. To represent the voice of the customer.
6. To accept the user stories that meet the definition of done and defined acceptance
7. criteria.

## Cross-functional Team

An agile team generally consists of developers, testers, one technical lead, one product owner, and one scrum master. Product Owner and Scrum master are considered to be a part of Team Interface, whereas other members are part of Technical Interface.

## How an Agile Team Plans its Work?

An Agile team works in iterations to deliver user stories where each iteration is of 10 to15 days. Each user story is planned based on its backlog prioritization and size. The team uses its capacity − how many hours are available with team to work on tasks − to decide how much scope they have to plan. In each release, there are 3 stages.

### Iteration planning

Planning what is going to be in each user story.

### User Stories

Cycles of Define, Code, Test, Define, Code, Test, etc.

### Review

Review of the work done, usually culminating in an After-Action Review plan document.

In each release, a number of User Stories are selected, and points given to them in a rough estimate. In iteration planning, User Stories are broken down into tasks and subtasks.



A Point is the amount of time a team can commit to the software user story. A point is usually 8 hours. Each story is estimated in points. The total amount of points adds up to the amount of time a team can work in the fixed time, generally 2 weeks.

Capacity is the amount of time an individual can commit. Total capacity of the team is equal to the total amount of points x (hour/point), and is equivalent to the amount of time a team can work in the fixed time, generally 2 weeks.

## User Story

A user story is a requirement which defines what is required by the user as functionality. A user story can be in two forms:

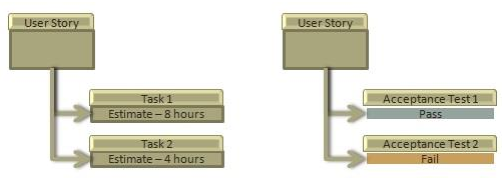
* As a <User Role> I want <Functionality> so that <Business Value>
* In order to <Business value> as a <User Role> I want <Functionality>

User story is a requirement that a particular role wants to realize the need for a particular business value. During release planning, a rough estimate is given to a user story using relative scale as points. During iteration planning, the story is broken down into tasks.

## Relationship of User Stories and Tasks

User story talks about what is to be done. It defines what a user needs. User story is divided into tasks during planning. Task talks about how it is to be done. It defines how a functionality is to be implemented. Stories are implemented by tasks. Each story is a collection of tasks. Tasks are estimated in hours, typically from 2 to 12 hours.

Stories are validated using acceptance tests. Tasks are validated using unit, integration and regression testing.



## When a Story is Done

The team decides what done means. The criteria may be:

* All tasks (development, testing) are completed.
* All acceptance tests are running and are passed.
* No defect is open.
* Product owner has accepted the story.
* Deliverable to the end-user.

## What are Acceptance Criteria?

Criteria defines the functionality, behavior, and performance required by a feature so that it can be accepted by the product owner. It defines what is to be one so that the developer knows when a user story is complete. It is the minimum number of requirements that must be satisfied by a story for the product owner to be ok with it.

## How the Requirements are Defined?

Requirements are defined as a User Story, with acceptance criteria, and tasks to implement the story.

# References

**There are no sources in the current document.**