**Agile Cheat Sheet**

**Nilav Patel**

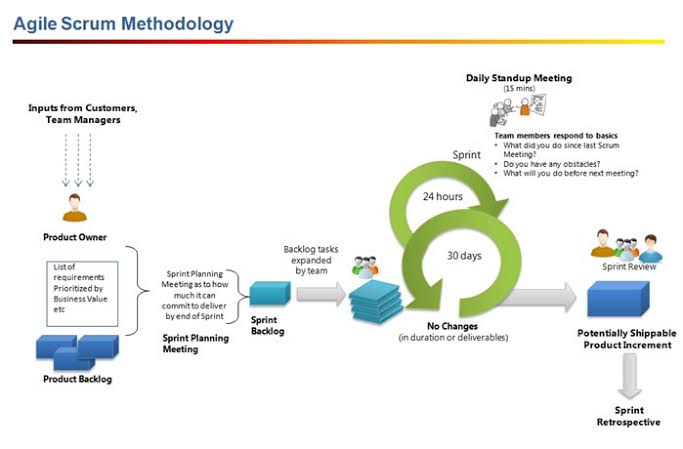
Agile:

Agile is an iterative development methodology, which gives you continuous integration, means at the end of every iteration it will add some value to product.

Also requirement can be change or add or remove at any point of life cycle.

Scrum, Kanban, Lean are agile frameworks. Scrum is most commonly used framework.

# Scrum process workflow:



1. Create Product Backlog (bunch of all user stories) by dividing all features of products into smaller chunks.
2. Create Release Backlog (bunch of user stories for current release).
3. Create Sprint Backlog (bunch of user stories with high priority) in sprint planning meeting.
4. Create Tasks for sprint. ( sprint length should be between 2-4 weeks )
5. Estimation in time/ story points.
   1. 1 Hours /2 Hours /4 Hours /8 Hours
   2. 2 Days /3 Days /5 Days /10 Days
   3. 1 Months /2 Months /3 Months /6 Months
   4. Story points
6. Daily stand up meeting for 15 min.

Discuss what done on yesterday, what doing today, and obstacles if any.

1. Sprint Retrospective meeting at end of every sprint.

Discuss what are the plus points and what need to improve based on previous sprint.

1. Product backlog refining and reprioritize by product owner.
2. Start with point 3 again till all sprints are done in release backlog.
3. Increase the product version number.
4. Start with point 2 again till full product is released.

**Scrum** **Team:** Product Owner, Scrum Master, Developers, Designer, Tester

**Stack Holders**: Sponsors, End Users

Agile removes designations like project manager, team lead from team structure, it encourage equality in team.

Here product owner is the chain between stack holders and scrum team, who knows all about product.

Scrum master is a person who make sure that team follows all the processes and remove obstacles and help team members.

# Story points:

Planning poker is played using poker cards.

Story pints are in Fibonacci series, but some organizations using below series

0,1/2,2,3,5,8,13,20,40,100,infinite, and cup of tea

Here 13 is for most uncertain tasks.

Infinite means task will never be completed.

Tea means I need a tea break, and I will make tea whole team.

Also t-shirt sizes are used like XS, S, M, L, XL, XXL.

**Note:** When estimating with story points, each team member will choose a card with story point, high and low story point members will share the reason for their perception, and at the end all team members needs to agree on a common baseline for estimation.

**Question:** How to handle unfinished work from current sprint to next sprint?

**Answer:** There are two ways to handle unfinished work.

1. Move user story to next sprint if it is still valuable.

In this case no points added to team velocity credit, points will be added to the sprint in which task will be completed.

1. Rewrite user story with unfinished work and move current user story to complete status after removing unfinished work.

In this case, some points are added to team velocity credit. And new user story will be moved to product backlog.

**Question:** What is a team size for agile? Who can be agile team members?

**Answer:** 7 +-2. Agile team having developers, designers, QA testers as a member.

**Question:** Any tools for agile?

**Answer:** <https://taiga.io/> (open source)

<https://www.atlassian.com/software/jira> (paid)

**Question:** Who is responsible for defect after QA sign off and UAT?

**Answer:** Everyone. Developer, QA, Client.

**Question:** What is UAT?

**Answer:** User acceptance testing is known as beta testing or end user testing.

This testing is performed by end user or client.

**Question:** How many types of meetings are used in agile?

**Answer:** Sprint Planning, Daily scrum meeting, Sprint presentation, Retrospective.

# User Story

## Fields

ID

Title (As a <role>, I want <feature>, so that <benefit>)

Description

Acceptance criteria (Anyone can write US and acceptance criteria)

Story Points / time

Priority

1. Product cannot ship without the feature.
2. Product cannot ship without the feature, but it doesn't have to be addressed immediately.
3. Implementation of the feature is optional based on resources, time, and risk.

Risk (High, Medium, Low)

Iteration (Sprint number)

Assigned To

Assigned On

Related Work Item (Parent Feature and child tasks)

Comments/Notes/Discussions

## Statues

New

Active

Resolved

Closed

Removed

On Hold

**Notes:**

User stories should be independent, smaller in size, and testable.

It's product owner's responsibility to write user stories, but anyone else can write it.

Acceptance criteria should be fulfilled at the completion of user story.

# Task

## Fields

ID

Title

Description

Activity (Documentation, Requirement, Development, Design, Deployment, Testing)

Efforts (Hours) (Original Estimation, Remaining, Completed)

Assigned To

Related Work Item (Parent user story or issue)

## Statues

New

Active

Closed

Removed

# Epic & Features:

When user story is too long, then it is add as an epic. Then it is further divided into smaller chunks as a features and then features are divided into user stories.

# Issue

## Fields

ID

Title

Description

Expected behaviour

Actual behaviour

Steps to reproduce

Severity (critical, high, medium, low, nice to have)

Affected areas

Related Work items (feature, user story id)

Reported by

Assigned to

Attachments

Software release version

Comments/Notes/Discussion

## Statuses

New

Active

Resolved

Reopen

Closed

On hold

# Burn down chart:

The burn down chart is representation of work done vs time, so when you pull the line at the end you will get estimated time to finish the whole product.

On y-axis add how many story points covered in sprint,

On x-axis add week/time for that sprint.

So when you pull the line to intersect x-axis, that intersection point will be the estimated time to complete product.

# Velocity :

Velocity is an average story points that team can complete in a sprint.

Velocity = total story points of all sprints / total sprints

Agile is not a knowledge, it's a practice.

Agile will teach you something at the end of every iteration, which helps you to take decision in next sprint.