



BASICS OF SCRUM IN AGILE

Abstract

Basic Scrum handbook for the beginners in the Agile world and CSM (Certified Scrum Master) aspirants.

SudaRamakrishna Thiparthy CSM[®], ITIL[®]

References:

Scrum Alliance classroom training
www.scrumalliance.org

Contents

Definition of SCRUM	2
Standards of Scrum	2
Estimation and Techniques	2
Agile Manifesto	2
Scrum Master	3
Product Owner	3
Scrum Roles and Responsibilities	3
Development Team	3
Typical Sprint Phases	4
1. Product Backlog:	4
2. Sprint Planning	4
3. Sprint Backlog	4
4. Sprint	4
Sprint Burn-down charts	6
5. Daily Scrum	9
6. MVP (Minimum Viable Product)	9
7. Sprint Review	9
8. Sprint Retrospective	9
Typical Sprint flow Diagram	9
Scrum Jargon	10

Definition of SCRUM

Scrum is one of the Agile frameworks followed to complete challenging projects wherein there are dynamic changes in the requirements by using one or more cross-functional, self-organizing teams of about 7 +/- 2 people on each team.

Standards of Scrum

- Scrum consists of 3 roles: product owner, ScrumMaster, and development team/engineering team. Roles and responsibilities of each role will be elaborated in the following sections.
- Scrum uses fixed-length iterations called sprints ranging from one Week four weeks (or 30 days long).
- Scrum teams should consist of 7 +/- 2 people.
- Sprint cannot exceed more than 30 days.
- Sprint is time bound.
- Scrum team aims to build a potentially shippable product by end of each sprint.
- Daily Scrum / Stand-up meeting should be only for 15 mins. and as the name goes, the team should be standing during entire meeting time.

Estimation and Techniques

It's nowhere a rule saying you have to adopt one of the techniques to estimate. The team can use any logic to estimate their user stories.

Once the team has worked for awhile in Agile projects, their ability to estimate the user stories becomes much better and more accurate.

But teams new to Agile sometimes face difficulty in estimating the points for the user stories.

Below are the few estimation techniques, which can be used to estimate the user stories accordingly:

Planning Poker

Planning Poker is a game that the team members can play during planning meetings to make sure everyone participates.

To begin with, each team member is given with set of cards with numbers on them. Usually the team follows the Fibonacci sequence: 0, 1, 2, 3, 5, 8, 13, and 21.

Then each story is read aloud. After each story presentation, each of the team member is asked to hold up the card showing the level of effort involved according to them.

Once every team members gives the points for the story, the one with lowest value and the one with highest value will be asked to explain why they have chosen that value. Experts with detailed knowledge will be able to explain to the team why a certain story is much easier/more difficult than it first appears.

Then the team will come to an agreement on the estimate for that user story and conclude on that.

T-Shirt Sizes

At times it happens that the team members start aligning the story points with the hours of effort, which can create confusion. To avoid this, it may be more effective to switch to a non-numeric estimation technique.

With T-Shirt Sizing, the team is asked to estimate the story effort in terms of extra-small, small, medium, large, extra-large, or double extra-large. By removing the numeric concept, team will be free to think in more abstract way. But there is a practical issue involved with this method. Non-numerical scales are generally less granular. While that can speed up the voting process by reducing the number of options, it may also reduce the accuracy of velocity estimates.

For the above reason, the T-Shirt Sizes method of estimates will be good for the teams just starting with Agile projects. Eventually it's a good idea to move toward numeric method of estimation.

Relative Mass Valuation

Usually the process of deciding the story point for each story will consume lot of time. With this method, it consumes less time.

Agile Manifesto

While there is value in the items on the right, we value the items on the left more:

Individuals & interaction	over	Processes & Tools
Working software	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to Change	over	Following a Plan

Scrum Roles and Responsibilities

Product owner

- Responsible person for release burn-down chart.
- Responsible for the product vision.
- Person who maintains the product backlog and the responsible person for the same.
- Only person who can make a decision whether the product is ready to ship.
- Only authoritative person to take a decision about abruptly discontinuing/stopping the sprint.
- Person responsible to provide clarification on the user story to the development team whenever needed.
- Only authoritative person to take the decision of including any further functionality to the product backlog.
- Person to constantly reprioritize the product backlog.

ScrumMaster

- Facilitator of the Scrum process.
- Helps resolving any issues / impediments faced by the Scrum team.
- Shields the team from external interferences and distractions.
- Creates a positive environment for the team to be self-organizing.
- Enforces timeboxes.
- Person who maintains sprint burn-down / burn-up charts.
- Person who facilitates the required meetings.
- Has no management authority for the team (for example, he or she cannot command the team to do a specific task).
- Promotes improved practices of engineering.
- Only POC for escalations.
- Only person who can take the escalations or request to the top management.

Development Team

- Should be a self-organizing and cross-functional team (consists of the members with testing, development, business analyst, domain expertise, etc., skills).
- Should self-manage the tasks among the team.
- Should resolve people management issues if any among the team and should only take it to ScrumMaster if it has gone out of the team's control.
- Works with product owner in reprioritizing the product backlog items.
- Consists of 7+/- 2 members.
- Responsible to complete the committed task for the sprint.

Typical Sprint Phases

1. Product Backlog

The product backlog is the list of functionalities with the short descriptions derived from the requirements and the roadmap of the project; and they are displayed in terms of user stories. Product backlog is the single source of requirements for any changes to be made to the end product.

User stories are prioritized by the product owner in discussion with the development team considering the business priority of those user stories and the dependencies on the other stories, if any.

Responsibility of managing the product backlog lies with the product owner.

3. Sprint Backlog:

Sprint Backlog is the list of the product backlog item for which the Development team commits to deliver as part of that particular sprint.

4. Sprint

Defined period of time in which the specific committed work has to be completed. Sprint is timeboxed and cannot be extended under any circumstance, irrespective of whether all the user stories committed are delivered or not. If any user story or part of a user story is pending after the sprint completion time, that particular user story will be moved back to the product backlog for reprioritization.

2. Sprint Planning

Sprint planning meeting is attended by the product owner, ScrumMaster, and the development team.

Sprint planning meeting will last the length based on the following rule of thumb:

'Multiply the number of weeks in your sprint by 2 hours' and this defines the duration of the sprint planning.

How and when it will be decided what should be the duration of the sprint and into how many sprints the project be divided into to address the requirements in the product backlog.

What is 'Sprint 0'?

Sprint 0 is the phase wherein the resource planning, project planning, sprint duration, and the number of sprints will be decided and also a bit more discussion happens on the product backlog items.

Sprint planning is basically divided into 2 parts:

Sprint Goal

Sprint Backlog

Sprint Goal: The first half of sprint planning goes with deciding the sprint goal. Here the team discusses on what needs to be achieved at the end of that particular sprint. It is discussed and decided collaboratively by the team and the product owner.

Sprint goal can be used for quick reporting on what has been decided to complete as part of that sprint to the interested stakeholders who are keen to know what is happening.

Sprint Backlog: This is the other output of sprint planning. The Sprint backlog is the list of the product backlog item for which the development team commits to deliver as part of that particular sprint.

Sprint Velocity: Sprint velocity is the capability of the development team to deliver the number of user stories based on the estimated story points. Sprint velocity of the first sprint decides the capacity of the development team and hence will be useful for planning further sprints accordingly.

For a better explanation of the entire sprint process, I consider the various stages of the sprint as user stories and explain the progress along with each phase of the sprint.

Example:

Product Backlog

Let's say below are user stories with story points. Let's say it has been decided to complete all these user stories in 4 sprints, and the team has assumed a velocity of 10 story points per sprint. Let's say it's a 1 week sprint.

Product Backlog		
S.No	User Story	Estimation (User Story points)
1	Scrum Introduction	2
2	Product Backlog	4
3	Sprint Planning	4
4	Sprint backlog	3
5	Sprint	8
6	Sprint Burntdown charts	6
7	Daily Scrum	4
8	Sprint Review	4
9	Sprint Retrospective	4

As part of sprint planning, the goal for sprint 1 is to complete first 3 user stories and the same has been moved to the Sprint backlog as below.

Sprint backlog:

Day 1:

Sprint Backlog - Sprint 1				
S.No	User Story	Estimation (User Story points)	In Progress	Done
1	Scrum Introduction	2	YES	
2	Product Backlog	4	YES	
3	Sprint Planning	4		

Day 2:

Sprint Backlog - Sprint 1				
S.No	User Story	Estimation (User Story points)	In Progress	Done
1	Scrum Introduction	2		YES
2	Product Backlog	4	YES	
3	Sprint Planning	4		

..

..
..

Day 3

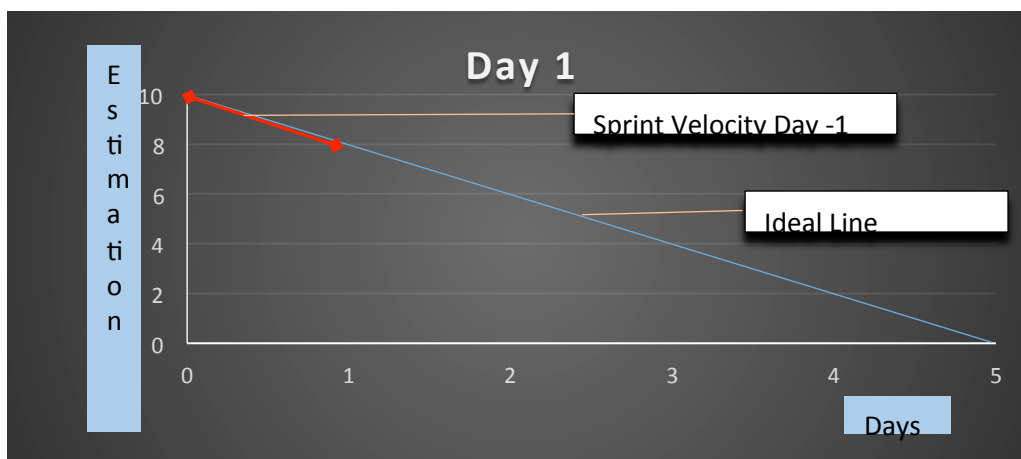
Sprint Backlog - Sprint 1				
S.No	User Story	Estimation (User Story points)	In Progress	Done
1	Scrum Introduction	2		YES
2	Product Backlog	4		YES
3	Sprint Planning	4	YES	

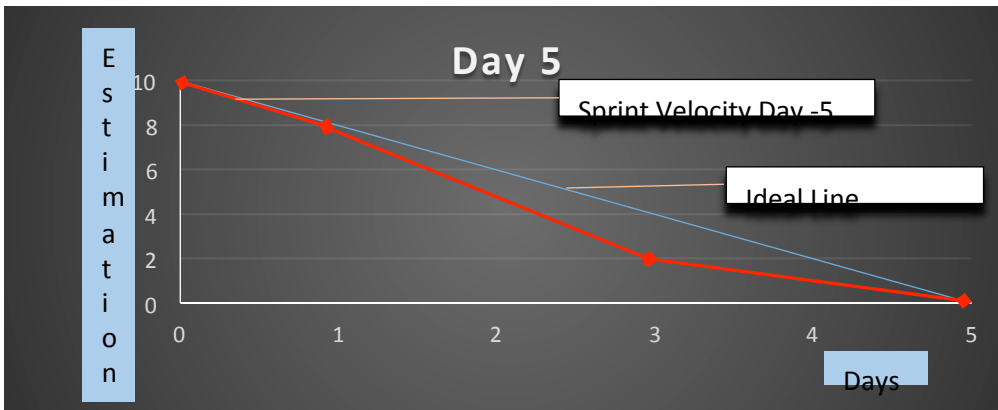
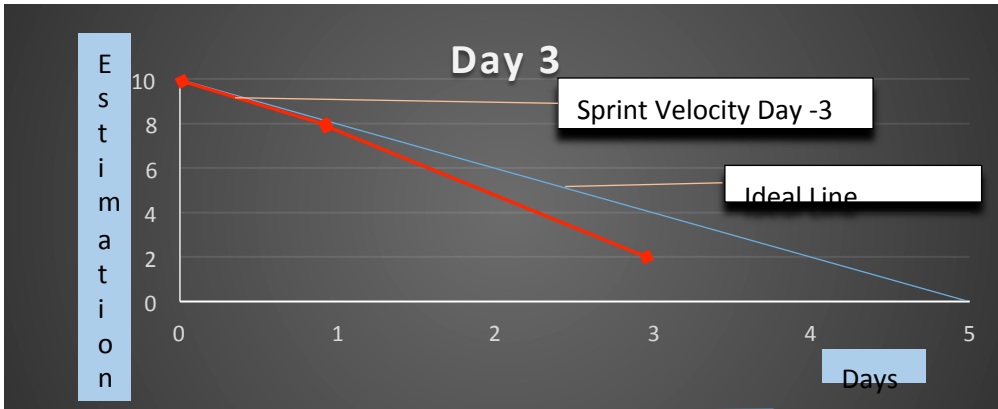
..
..
..

Day 5

Sprint Backlog - Sprint 1				
S.No	User Story	Estimation (User Story points)	In Progress	Done
1	Scrum Introduction	2		YES
2	Product Backlog	4		YES
3	Sprint Planning	4		YES

Sprint Burn-down charts:

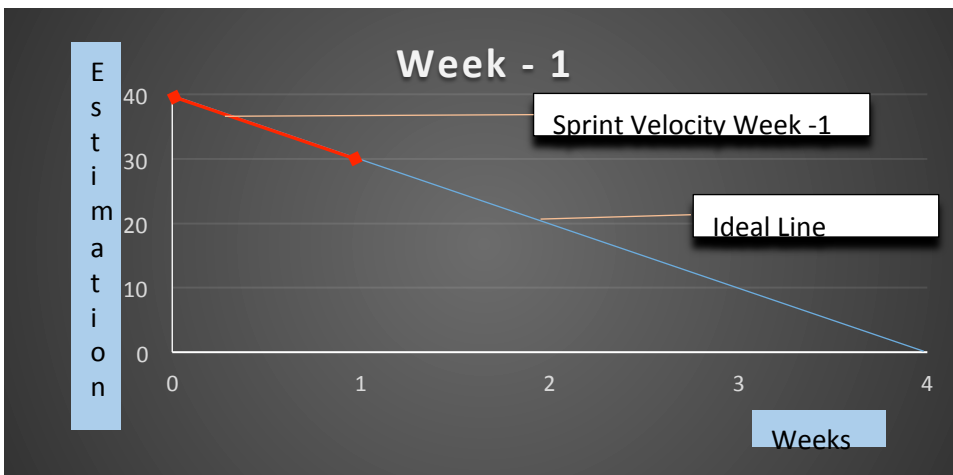


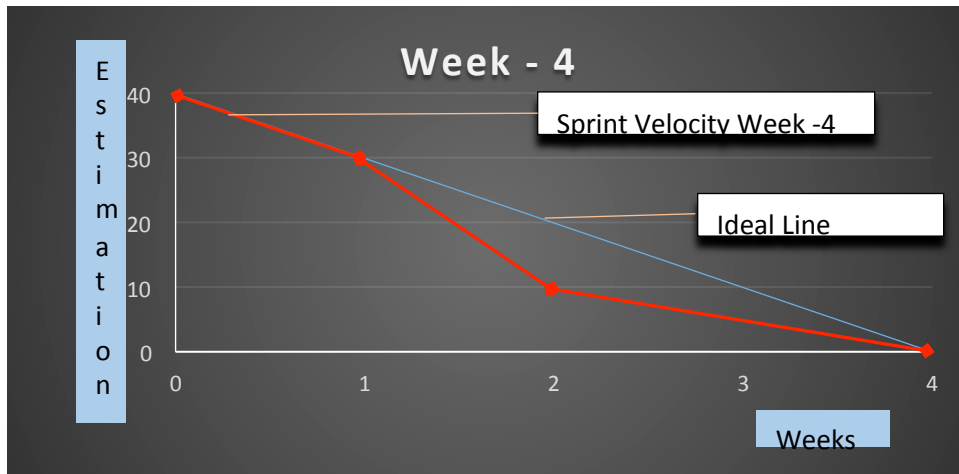


X Axis – Represents Days

Y Axis – Represents Estimation for the User stories committed for sprint 1.

Release Burn-down chart:





X Axis – Represents Weeks

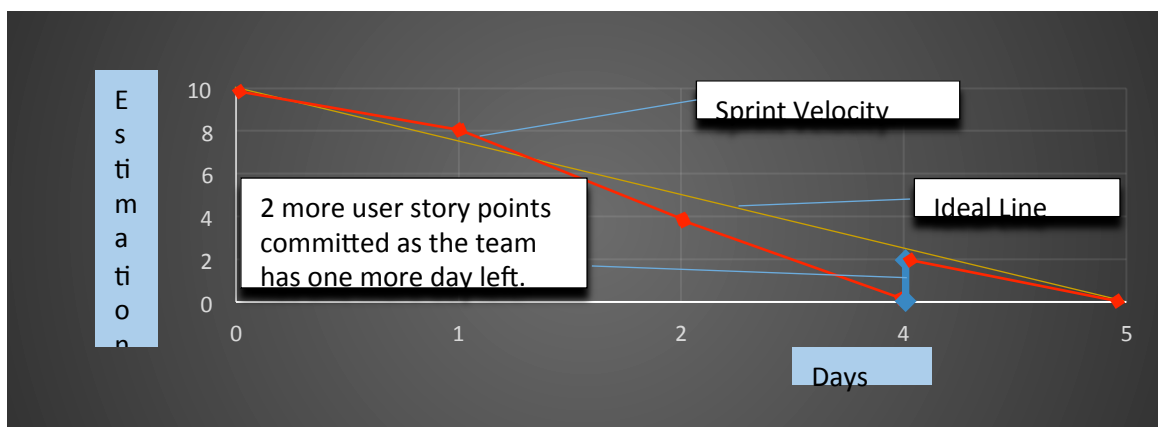
Y Axis – Represents Estimation for the User stories committed for Entire project.

The example above shows the happy scenario where in everything goes per the plan.

What happens if the development team has completed the committed user stories well before the sprint time line and they still have time?

In this scenario, the Scrum team can have a quick sprint planning meeting again and, based on the capacity and capability of the development team, they can commit to additional user stories and execute them. In this scenario, the sprint burn-down charts looks like those below.

Let's say, for example, the development team has committed to an additional user story of 2 points to complete after completing their usual capacity of 10 points. Let's say there is one more day left for them.



What happens if the development team couldn't complete all the committed user stories with in the specified sprint time line?

As the sprint is timeboxed, the sprint time line cannot be extended. The leftover user story will be pushed back to the product backlog for further prioritization and accordingly it will be considered for a future sprint.

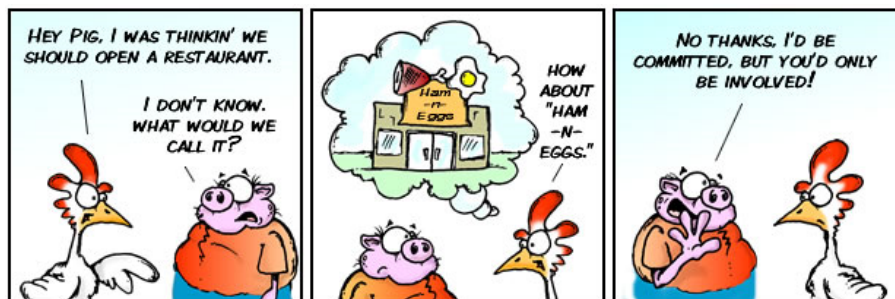
Scrum Jargon

Epic: Requirement/story for which the effort to complete is more than one sprint. They should be split into logical smaller requirements so that smaller stories can be planned to be addressed in a sprint.

Pig and Chicken in Scrum: Pigs are the roles who are committed (Scrum team – product owner, ScrumMaster, development team).

Chickens are the roles who are only participants (other than the Scrum team – Stakeholders, management team, etc.).

The classic cartoon below (© 2006 Implementingscrum.com) illustrates the Pig and Chicken roles in an Agile-driven project.



Scaling of Scrum: Usually the Scrum team consists of 7 +/- 2 people; more or less, and the team will not function efficiently. Now when we have a bigger project to execute, how to deal with the team strength? Here comes the concept of scaling Scrum, dividing the project into modules and then having a separate Scrum team for each module to complete and then finally integrate all the modules.

Scrum of Scrums: Whenever any technical issue arises, the ScrumMaster will arrange for a team consisting of one key member from each different Scrum team, and then form a separate team to analyze and address the issue.

Sprint velocity: Sprint velocity is the number of story points completed by a team in a sprint. Why do we need this? We need to know the velocity of the development team for the following reasons:

- We can predict the coverage of the user stories and plan the project accordingly.
- We can understand the capacity and capability of the team before committing the user stories for a sprint.

Usually we can come to an understanding of the team velocity after first 2 to 3 sprints.

.... Thank you