

Estimation in Story Points¹

Ajay Shenoy

Agile Projects integrate numerous techniques that will not work waterfall for estimation. One such technique is estimating the size of user stories with abstract measure of effort and use of story points to define effort of work, which can be completed in a sprint.

Scrum Project Teams can implement a set of features, which are broken down into stories, which can be completed by the scrum team in short sprints. User Stories would be broadly pulled in from the Product backlog, which would be in the priority order set by the Product Owner. The Scrum team may not be able to implement all stories present in the product backlog in a single sprint, so there could be different stories, which could be pulled up from the product backlog. The number of stories depends on the amount of effort it takes to fully implement each story.

Sprint Planning

One of the ceremonies in Scrum is Sprint planning meeting where the team would estimate out all the tasks associated with each user story and in the Sprint Backlog. All the required resources needed to complete all the tasks for each sprint is considered during effort estimation. A shared task list is created to estimate the duration and effort of each user story in Sprint backlog. This practices the shared responsibility among agile teams, which results in common estimation process, which is consistent.

How do Scrum Team Estimate?

The most commonly used technique in Scrum is the usage of Story points to represent the relative effort of user story or task. Although estimating in hours is very common in traditional waterfall projects but is rarely used in agile. A common approach used is usage of story points to estimate the relative complexity while implementing a user story. You can set any value to a story point that a scrum team decides upon. Each scrum team will have their own scale of Story point. There would be no set value defined to a story point. Although the value needs to be consistent across the team. The scale of story point may not be the same scale, which another scrum may use.

Story points are typically based on modified Fibonacci series, which is a number sequence. The sequence goes like 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100. For Example, the team might have decided that value of a story point is 1 day, and so 5-story point will have a value of 5 days. If team has picked User Story A and assigned 1 story point. If team picks Story B

¹ How to cite this article: Shenoy, A. (2019). Estimation in Story Points, *PM World Journal*, Vol. VIII, Issue V, June.

and think it will take twice the effort to implement, team would assign 2 story points to user story B. There is no per-define formula about use of 1 or 2 story points. Another scrum team might assign this same two story 8 & 16 respectively depending on the scale they have agreed upon.

Suppose a new café opens in your neighborhood and you decide to try a cup of coffee. Since you have been to many other cafés you will probably order a cup of coffee without asking the server as to how big is the coffee cup. If you do ask the server as to how big is the coffee, he would most likely respond with by holding his hands to illustrate the size. Here you are ordering by relative size rather than measured size. You are not ordering by exact size like 2 ounces of coffee or 4 ounces of coffee. Even though the numbers are different, the key is that the story points represent **relative sizing** of the user stories. In the examples, user story B will take twice as much effort to implement as user story A.

Once the team has established their story-sizing factor, they can now determine how many points can be delivered in a sprint.

Many techniques can be used to estimate stories in story points for each user story. The below are the norms of the estimations techniques which can be followed.

- The team and not one individual define the story points for each user story. However, there could be a scenario where some team members may not have the skillset to estimate the user story.
- The story points represent the total amount of work required to implement the user story.

What Goes into a Story Point?

- **Complexity:** Consider the complexity of the story. Please also keep in mind that **complexity** and **effort** (again, or *difficulty*) are not one in the same. By definition, *difficulty* in this context is based on the **amount of effort needed** to solve a problem and to complete tasks. On the other hand, *complexity* is associated more towards a person's **way of thinking, action, and knowledge and not the amount of effort**.
- **Risk:** Consider the team's inexperience with developing this story. 'Quality is key', is the credo in scrum. However, based on the story risk – which is ascertained prior to the sprint – it is possible for stories with a higher risk to be tested more extensively than those stories with a lower risk. The risks and the ways in how to cover these are directly related to the acceptance criteria, as indicated on the Confirmation segment of the story card.
- **Implementation:** Consider the implementation factors. A proper estimation of time in user stories is a crucial task for both the IT team as well as for the customer, especially in agile projects. Although agile practices offer a lot of flexibility and

promote a culture of continuous change, there are always clearly the need time boxed periods where an IT company has to commit to delivering working software. Estimating time of user story implementation provides clarity and the opportunity to control the project by the management, yet at the same time, it can increase pressure on software developers. Thus, incorrectly estimated user stories may lead to quality problems including system malfunction, technical debt, and general user experience issues

Consider all Factors and have a Definition of Done

Scrum defines the **Definition of Done** in simple terms: it is the acceptance criteria that are common to every single user story. For scrum teams, it is important to have a solid definition of what “done” means. They work in sprints and need some way of deciding whether a user story is actually finished. It is no good ending a sprint with a user story that meets all its acceptance criteria, but had no code review, has not been tested and isn't deployable. Such a story is clearly not done. Scrum teams solve this by strictly defining “done”

What must happen for the story to be marked as complete?

- All story should have automated acceptance test.
- The story should have well defined acceptance criteria.
- Code must be completely checked in to the source control system and the build should pass with all the automated tests running.
- The product owner must accept all stories completed for the sprint.
- Product is deployed to the test box and makes it to staging.
- All stories for the release are completed and accepted.
- Training manuals are available for users.

References

Agile Estimation and Planning (Mike Cohn)

About the Author



Ajay Shenoy

Bangalore, India



Ajay Shenoy, a certified Scrum Professional and Agile Coach, has been involved in Technology Solutioning since 2007. He started working as a Solution Engineer and slowly incorporated into a technical program manager. He is a Certified Scrum Professional and has good knowledge on Prince2, Agile, Lean, Scrum, Kanban and SAFe frameworks. Along with expertise in Project management, he has deep interest in Technology side. With these skills, Ajay can help people understand process as well as Agile. Ajay has a perfect blend of project management with technical skills and business acumen.

Ajay started his Agile journey in 2012, as part of engineering teams. He practiced scrum and other agile frameworks in delivering successful products within limited time frames. Ajay is proficient in Engineering practices such as Scrum, Lean Software development, and Kanban and has designed several solutions and market rollouts working with product/services companies. He believes in following key agile practices like Just In Time, Value Stream mapping, Refactoring, Improving lead and cycle time.

He single handedly built a group comprised of 700 employees with different skills/roles. He indulges in several meets/ conferences and sharing knowledge on public platforms like linkedin with reference to agile. Ajay has coached/trained several teams in different organizations; he was part of an agile team to improve an existing framework.

He has a Master's degree in IT & Finance and is currently based out of Bangalore. You can reach him on his email @ shenoyajay82@yahoo.com.