**Let’s talk estimation!**

<http://www.reactiongifs.com/fake-excitement/>

<http://www.productmanagerhq.com/wp-content/uploads/2015/03/Agile-Estimation-Sizing-Stories.png>

**What’s the purpose of estimation?**

The practice of estimating work in software development is a challenge that we often encounter. It is not always easy to estimate effort, especially when the complexity and uncertainty of the story increases. The purpose of estimation using story points is to agree on the effort and complexity involved of a user story order in order to better plan and execute product development. Some other goals of estimating using story points are as follows:

* To be prepared to answer “What can be completed in two weeks?” during sprint planning
* To increase the accuracy and precision of that ^ answer
* To estimate against the team and not an individual
* To set expectations and align as a team
* To be able to forecast and prioritize the backlog

**What is a story point and why do teams use them instead of a measure of time?**

A story point is a relative measure used by teams to evaluate the effort and uncertainty involved with implementing a story. Story points make it easier for Scrum teams to think abstractly about the effort required to complete a story and should not reflect a set amount of hours a feature will take to complete. Story points help the team understand their own ability to take on and complete a new storyrelative to other stories they’ve previously completed. Without the useful abstraction provided by points, a team has a much harder time predicting how much work they can actually accomplish in a given period of time.

We use the Fibonacci sequence here at LFO as our unit of measure for story points. “The reason for using the Fibonacci sequence is to reflect the inherent uncertainty in estimating larger items.1” The difference between a 1 and 2 are probably better understood than the differences between a 34 and 55. The Fibonacci sequence helps teams to recognize that uncertainty grows proportionally with the size of a story. The larger a story is the more uncertainty there is around it and thus the less accurate the estimate will be.

<http://i.stack.imgur.com/eHWK9.png>

Story points also account for the non-project related work. With time values, we don't often account for the cost of doing business. Meetings, email, code reviews, etc. While they all are important parts of any workday, they don't actually count as work. Using story points allow for teams to isolate the work stream from the non-work stream items so that the estimates are more consistent.

http://www.thisiswhyimbroke.com/images/i-survived-another-meeting-that-should-have-been-an-email-ribbon-640x533.jpg

What about another paragraph that indicates what estimates/story points are NOT for – specifically, I’m thinking that they are *not* for grading a team on how much work/how well they’re doing or how quickly they’re going, and *cannot* be used to compare multiple Scrum teams.

The ability to navigate and prosper through change is what Agile is all about. Estimates will shift and a team’s velocity can help correct any inaccuracies in our estimates. The velocity of an iteration is the amount of story points completed in the previous iteration. When a team underestimates their velocity, it means they are not able to finish all the stories by the end of the sprint, which will make their velocity decrease. Similarly, if velocity is overestimated, a team is able to complete more stories by the end of the sprint which will make their velocity increase. By identifying if a team has over or under estimated their velocity, they are able to properly asses their sprint workload and make improvements moving forward. As the development team makes better estimates, their velocity stabilizes and becomes consistent and predictable which will allow the product owner can more accurately plan the product roadmap.

There are a variety of different ways to estimate. Here is a good list of some of those strategies and why they have worked: http://www.deltamatrix.com/agile-estimation.

Sources:

1. <https://en.wikipedia.org/wiki/Planning_poker>

2. http://en.wikipedia.org/wiki/Agile\_software\_development

3. http://en.wikipedia.org/wiki/User\_story