Agile Process_201_Assigning Story Points Procedure

Last Updated: 03/18/2016

Audience:
CWDS-NS Service Delivery Teams

Frequency:
As needed

After reading this procedure, the audience will be able to understand the procedure for assigning story points to user stories, which is part of the Backlog Refinement Process.

Setup Tasks:

Each service delivery team will need to establish the duration of each sprint iteration (1 week, 2 weeks, 3 weeks, etc.) based on team strength, skill and expected deliverables (can the work be done in 1 week, 2 weeks…will it take 3 weeks?) Each service delivery team will also calculate the estimated default velocity of your sprints (10 points, 20 points, 30 points, etc.) before starting story point estimation.

Have your Scrum Master see John Williamson for more detail.

Before Assigning Story Points: Developing and Prioritizing User Stories:

Once your service delivery team has defined the product backlog and organized the product roadmap, the next step is to decompose the backlog into smaller components, called stories. User stories are part of an agile approach that helps shift the focus from writing about requirements to talking about them. All agile user stories include a written sentence or two and, more importantly, a series of conversations about the desired functionality.

User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. Stories typically follow a simple template:

As a <type of user>, I want <some goal> so that <some value>

Once the backlog has been decomposed into user stories, it needs to be prioritized. Prioritization can take place in several ways - below are listed two examples:

1. Service Manager is responsible to determining priority (what needs to be done first and what is most important).
2. Service Manager sorts stories by Quarter or Timeline (assign a label of QTR1, QTR2, QTR3 and QTR4 to each story, and then sort only by QTR 1

After the stories are developed and the prioritization in place, you are ready to estimate the size of the stories.

Why do we need to estimate?

We need to provide some high-level initial estimates in order to get an idea of the complexity of our product backlog items. This is helpful because it helps to inform the decision maker about priorities and whether or not the features are likely to be worthwhile. In addition, from a management point of view, estimates give a perspective of how long an implementation may take.

The solution is to do a very high level indicative estimate. In fact it's a ***guestimate***. You estimate your product backlog in points, not in units of time. So we're not asking the team "how long will it take?" We're asking "how big is it?"

Estimate Story Point Logic:

Estimates on stories are meant to be relative to other estimates on stories. In other words, estimate in *relative* terms rather than *absolute* terms. For example, if you assign 3 points on a story, find a 1-point story and assess whether this story is 3 times as complicated as the smaller story. This is called triangulation, and is vital to ensuring consistency. Do this periodically throughout planning, particularly for stories that generate discussion and disagreement.

Estimates don't need to be accurate, but they do need to be consistent. Pivotal Tracker will correct velocity inaccuracies of the estimates.

Estimating Story Points via Fibonacci numbers:

There are three basic factors you should consider when assigning points to a story: complexity, effort, and uncertainty. You want your points to be forced into buckets with wide enough gaps between them that there is no need to argue over insignificant differences. Our management team has decided we will be using a Fibonacci-like sequence for assigning story points:

0, 1, 2, 3, 5, 8

Fibonacci numbers are a scientifically significant set of numbers, where each number is the sum of the previous two. The gap between the numbers increases with Fibonacci numbers, reflecting the difficulty to estimate higher numbers precisely. The thought is that a story with high points is probably a story that needs to be decomposed further.

Example: A backlog item describes a report. You've done similar reports before but this one has some complexity in the underlying data so you decide to assign this a 3. Next on the backlog is another report. You compare this one relative to the prior one. Is it bigger or smaller? Clearly 8 is a lot bigger and 1 is a bit smaller. And so on. To make sure the scale works for you, select what you think is the smallest story on the backlog. Give this a 1. Then find what you think is the biggest story on the backlog. Give this an 8. Now you have your markers. Begin sizing the backlog, working from the top, using the Fibonacci number

At this time, the number 0 will not be used for user stories. However, at some point in the future, the project may decide to use zero-point stories to identify sprint milestones.

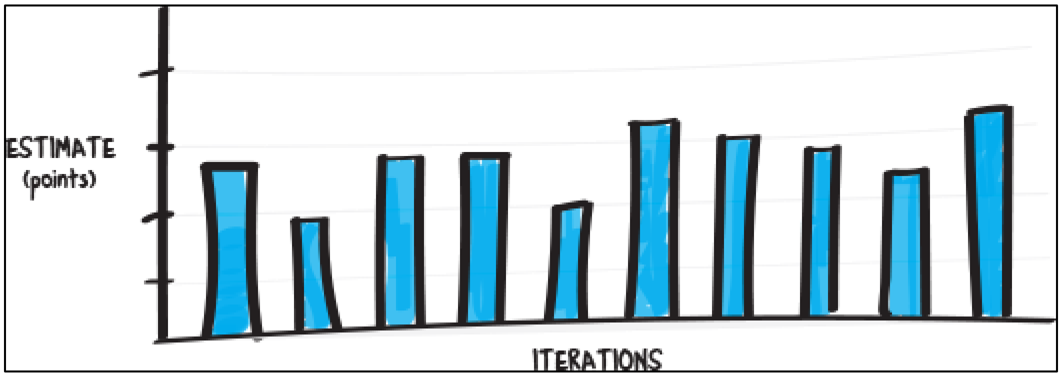
Review Priorities:

Once you’ve estimated the stories in the backlog – or enough for two sprints worth of work – ask the Product Owner to take another quick look at priorities. Maybe now they can see the relative size of the features they’ve asked for, they might change their view of priorities. ‘Wow, if that’s an 8, I’d rather have the other stuff first’, or ‘if that’s only a 2, let’s get it in the next release’. If any priorities are changed, simply move the item’s position in the order of the backlog. Stories at the top of the list are most important and worked first.

Velocity:

Velocity is the number of story points that were completed in any given iteration. With this value known (approximately), the team can plan within this figure and know that they will most likely be able to finish. Velocity is not the same as speed. Once you can estimate your velocity over an average of three sprints, you will have a better idea of the planned velocity for future iterations.

A typical velocity chart for an agile project team might look like the image here.


ESTIMATE
(points)
ITERATIONS