The secrets behind story points and agile estimation

Good estimation helps product owners optimize for efficiency and impact. That's why it's so important.

BY DAN RADIGAN

Estimation is hard. For software developers, it's among the most difficult–if not the most difficult–aspects of the job. It must take into account a slew of factors that help product owners make decisions that affect the entire team–and the business. With all that at stake, it's no wonder everyone from developers to upper management is prone to getting their undies in a bunch about it. But that's a mistake. Agile estimation is just that: an estimate. Not a blood-oath.

There's no requirement to work weekends in order to compensate for under-estimating a piece of work. That said, let's look at some ways to make agile estimates as accurate as possible.

Up Next
Metrics →

In agile development, the product manger is tasked with prioritizing the backlog–the ordered list of work that contains short descriptions of all desired features and fixes for a product. Product owners capture requirements from the business, but they don't always understand the details of implementation. So good estimation can give the product manager new insight into the level of effort for each work item, which then feeds back into their assessment of each item's relative priority.

When the engineering team begins its estimation process, questions usually arise about requirements and user stories. And that's good: those questions help the entire team understand the work more fully. For product managers specifically, breaking down work items into granular pieces and estimates via story points helps them prioritize all (and potentially hidden!) areas of work. And once they have estimates from the dev team, it's not uncommon for a product owner to reorder items on the backlog.

Agile estimation is a team sport

Involving everyone (developers, designers, testers, deployers... everyone) on the team is key. Each team member brings a different perspective on the product and the work required to deliver a user story. For example, if product management wants to do something that seems simple, like support a new web browser, development and QA need to weigh in because their experience has taught them what dragons may be lurking beneath the surface.

Likewise, design changes require not only the design team's input, but that of development and QA as well. Leaving part of the broader product team out of the estimation process creates

So don't let your team fall victim to estimates made in a vacuum. It's a fast track to failure!

Want to give story points a try? First, sign up or log into Jira Software >>

- Try this interactive tutorial to set up your first scrum project
- Learn how to configure story point estimations and track your user stories

Story points vs. hours

Traditional software teams give estimates in a time format: days, weeks, months. Many agile teams, however, have transitioned to story points. Story points rate the relative effort of work in a Fibonacci-like format: 0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100. It may sound counter-intuitive, but that abstraction is actually helpful because it pushes the team to make tougher decisions around the difficulty of work. Here are few reasons to use story points:

- Dates don't account for the non-project related work that inevitably creeps into our days: emails, meetings, and interviews that a team member may be involved in.
- Dates have an emotional attachment to them. Relative estimation removes the emotional attachment.
- Each team will estimate work on a slightly different scale, which means their velocity (measured in points) will naturally be different. This, in turn, makes it impossible to play politics using velocity as a weapon.
- Once you agree on the relative effort of each story point value, you can assign points quickly without much debate.

- Story points reward team members for solving problems based on difficulty, not time spent. This keeps team members focused on shipping value, not spending time.

Story points and planning poker

Teams starting out with story points use an exercise called planning poker. At Atlassian, planning poker is a common practice across the company. The team will take an item from the backlog, discuss it briefly, and each member will mentally formulate an estimate. Then everyone holds up a card with the number that reflects their estimate. If everyone is in agreement, great! If not, take some time (but not too much time–just couple minutes) to understand the rationale behind different estimates. Remember though, estimation should be a high level activity. If the team is too far into the weeds, take a breath, and up-level the discussion.

Ready to give it a try?

- Install this Planning Poker App
- Learn more about Planning Poker here

> Keep estimations high level. If the team is too far into the weeds, take a breath, and up-level the discussion. #AgileTips

Estimate smarter, not harder

No individual task should be more than 16 hours of work. (If you're using story points, you may decide that, say, 20 points is the upper limit.) It's simply too hard to estimate individual work →

items larger than that with a high degree of confidence. And that confidence is especially important for items at the top of the backlog. When something is estimated above your team's 16-hour (or 20-point) threshold, that's a signal to break it down into more granular pieces and re-estimate.

For items deeper in the backlog, give a rough estimate. By the time the team actually begins to work on those items, the requirements may change, and your application certainly will have changed. So prior estimates won't be as accurate. Don't waste time estimating work that is likely to shift. Just give the product manager a ballpark figure she can use to prioritize the product roadmap appropriately.

Learn from past estimates

Retrospectives are a time for the team to incorporate insights from past iterations–including the accuracy of their estimates. Many agile tools (like Jira Software) track story points, which makes reflecting on and re-calibrating estimates a lot easier. Try, for example, pulling up the last 5 user stories the team delivered with the story point value 8. Discuss whether each of those work items had a similar level of effort. If not, discuss why. Use that insight in future estimation discussions.

Like everything else in agile, estimation is a practice. You'll get better and better with time.

DAN RADIGAN

Agile has had a huge impact on me both professionally and personally as I've learned the best experiences are agile, both in code and in life. You'll often find me at the intersection of technology, photography, and motorcycling. Find me on Twitter! @danradigan

Five agile metrics you won't hate

How to use agile metrics. Learn about sprint burndown, epic and release burndown, velocity, control charts & the cumulative flow diagram.

Read this article →

Agile Topics

Agile project management
Scrum
Kanban
Design

Software development
Product management
Teams
Agile at scale
DevOps

Sign up for more agile articles and tutorials.

Email

email@example.com

Subscribe