JOIN OVER 100,000 PROS! Join Log In | Member Benefits

STICKYMINDS

Agile Testing DevOps Security AI

Estimation Poker

ARTICLE

By Steve Berczuk | February 28, 2010

SUMMARY

Planning aoker, an estimating method popular with tgile teams can address some of these issues. Briefly, planning poker involves getting the developers on a team together to estimate stories using a deck of cards that have numbers that represent units of work.

Estimation is a necessary part of software development. Product owners want to know how much work can get done by a deadline, project managers need to make commitments, and developers want to know if they committed to a reasonable amount of work. While estimates are often inaccurate, estimates provide landmarks along the way of a project to gauge progress.

So, estimates are an inevitable, and useful part of the software development process. Many complain that the process of getting to those estimates, *estimation,* takes too long, so planning sessions are cut short and teams don't have enough time to discuss issues that have uncertainty. By appropriate use of planning poker, you can balance the needs for good estimates while minimizing time spent estimating.

Sometimes when a team is asked to estimate a backlog item, one or more people with expertise in an area are asked to estimate the item, but this is not the best way for an agile team to get a good estimate.

There are benefits to involving a larger part of the team in the estimation process. The challenge is that people feel that involving the whole team is wasteful if the estimation process takes too much time. On the other hand, inaccurate estimates have their own costs for the team and the other stakeholders.

Planning Poker, an estimating method popular with agile teams can address some of these issues. Briefly, planning poker involves getting the developers on a team together

We use cookies to ensure you get the best experience on our website.

Accept

Read More

People resist planning poker for reasons like:

- It seems inaccurate if the person doing the estimating does not having the "appropriate" expertise. A UI developer may not feel qualified to estimate a story that seems to be mostly backend processing, for example.
- It seems like a waste of time because people believe that one person can estimate for everyone.
- It seems inaccurate since the person who's been assigned the work should estimate it based on their skills.

Even if you find yourself throwing a wild guess at a planning poker session, the fact that you don't understand the scope of the issue is useful information. The benefit of having the entire cross-functional team understanding and estimating stores is that you can identify challenges across the application. What might be easiest to do in the back-end can add work to the application tier or UI, and also make testing harder. Having one person estimate can make it hard to identify misunderstandings and issues because we tend to want to agree with "the expert," and there is no forum for identifying misunderstandings. It's not always clear at the start of a project who the best person for task will be, both for the reasons I just mentioned and because assigning the work up front can lead to inefficiencies if work takes more or less time than estimated.

If you find that your estimates are inaccurate, or your estimation process takes too long, consider the following approach:

- Gather team members who are working on all aspects of the application. You need not have the whole team, but be sure to represent each "architectural layer". If your team is less than 7 people or so, include everyone.
- Look at the description of each story or problem report in priority order. Ask the team to pick cards based on what they read.
- See how close the estimates are.
    - If they are close, ask someone to explain what they envisioned doing to implement the issue. If someone has a vastly different idea, they should speak up.
    - If they are different, as someone with one of the extreme estimates to explain their reasoning. This will start a conversation about what the requirement means, and what implementation strategy makes sense.

This process helps you to focus discussion time on the hardest, highest priority issues. You will want to be sure that to allocate an **appropriate amount of time** to planning and estimating relative to your sprint length. You may still run out of time, but even if you do, you'll have discussed and estimated the highest priority items as accurately as you could have, knowing what you knew.

The biggest challenges to having accurate estimates are not having consensus on the "what" and not understanding the details of the "how." The process above is one way to focus discussion on the high-risk items in your backlog, while keeping the time spent on estimating reasonably low.

Topics: estimating,

We use cookies to ensure you get the best experience on our website.

ABOUT THE AUTHOR

Steve Berczuk

Steve Berczuk is a Principal Software Engineer with experience as a manager, Scrum Master and technical lead in Boston, MA. The author of *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*, he is a recognized expert in software configuration management and agile software development. Steve is passionate about helping teams work effectively to produce quality software. He has an M.S. in operations research from Stanford University and an S.B. in Electrical Engineering from MIT, and is a Certified ScrumMaster. Contact Steve at steve@berczuk.com or visit berczuk.com and follow his blog at blog.berczuk.com.

UPCOMING WEBINAR

LETS HANG!

Log In or Register to post comments

We use cookies to ensure you get the best experience on our website.

0 comments

Join the crew for FREE! Access exclusive content, contribute, and more.

See All Member Benefits

We use cookies to ensure you get the best experience on our website.