Planning Poker

or

How to avoid analysis paralysis while release planning

By James Grenning

The customer reads a story to the team. Two guys are involved in discussing the impact of the story on the system. Reluctantly, an estimate is tossed out on the table. They go back and forth for quite a while. Everyone else in the room is drifting off, definitely not engaged. The discussion oscillates from one potential solution to another, avoiding putting a number on the card. When the discussion finally ends, you discover that the estimate did not really change over all that discussion. You just wasted 20 minutes of valuable time. You have 25 more stories to estimate, you don't want to make a career of release planning.

Extreme programming employs two levels of planning, iteration planning and release planning. Iteration planning is short term and very detailed. Iteration planning is a good time to get into the details of how to implement the story. Release planning is a higher-level plan with a long-range time horizon. During release planning, the team is taking a less precise and longer-term view of the project. There is usually a long backlog of stories. The release-planning objective is to get a ballpark estimate of the effort to build the product, and to split the product into interesting release. Precision of individual estimates is not the goal. Determining the project scope is.

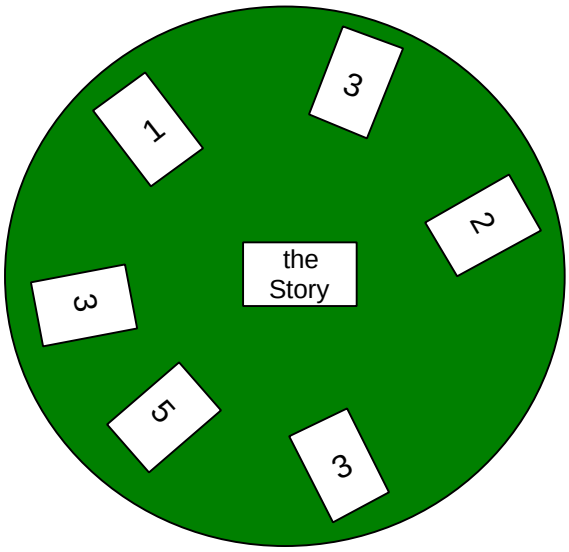
Story estimates are just what their name implies. They are estimates. Some estimates are over-estimates. Some are under-estimates. Every now and then, the estimates turn out to be just right. The team is looking out far, at many stories, including ones that will never be implemented. It is OK to be less precise. Why invest in precision before it is needed? A deck of fairly accurate estimates are better than a pair of very precise estimates (and these probably are not that precise anyway). The team should be trying to get good at using the intuition, their gut. Don't be cavalier about the estimates, but speed is important if the team wants to finish the meeting and get back to building the product.

There were two problems identified in the opening paragraph: estimates were taking a long time, and the whole team was not involved. There is nothing like a little game of planning poker to solve both those problems. The mechanics of planning poker are simple. The customer reads a story. There is a discussion clarifying the story as necessary. Each programmer writes their estimate on a note card without discussing their estimate. Anticipation builds. Once all programmers have written their estimate, turn over all the cards. If there is agreement, great, no discussion is necessary, record the estimate and move on to the next story.

If there is disagreement in the estimates, the team can then discuss their different estimates and try to get to consensus. If you can't get consensus, don't sweat it. It is only one story out of many. Defer the story, split it, or take the low estimate.

grenning@objectmentor.com

Lets play out a story hand that does not start with consensus. The customer deals the story. Programmers ponder their estimate. All together, they flip over their card. The current story hand looks like this:



The programmers at the extremes start the discussion. It might go something like this.

> "Why do you think this story is so hard?"
> "Why do you think this story is so easy? Did you think about having to modify the communications protocol? The last time we did that it cost us four days alone!"
> "Yeah, but this is basically the same as the last time we changed the protocol and all we have to do is add another message class"
> "OK, but it is still not a one"
> blah… blah… blah
> "Let's call it a three"
> "OK, Done"

Do you remember the other problem mentioned in the opening paragraph? Not all meeting attendees were participating. Now all participants are players in the game. They all have to be a turtle and stick their neck out when they play their card. Everyone gets experience at estimating. The whole team plays the game, not just the most vocal, or most senior. Everyone participates.

Play this game, you will notice that the same estimates are reused. Each player ends up with a set of cards, each with an estimate on it. It looks like a poker hand. When the story is read, the estimate is pondered, odds calculated, other players scanned, and a card is chosen from the players hand and played face down. Then all players roll'em[1] simultaneously.

One concern with planning poker is that important discussions might not happen. Estimates could become meaningless without the discussions. When I've used this technique, I have noticed that the team is quick at estimating familiar stories. Stories that are a natural

[1] Roll'em – poker terminology, player turns a face-down card over.

grenning@objectmentor.com

progression from already built stories. Things tend slow down when the customer deals stories from a new part of the application. The discussions seem to happen as necessary. When the customer deals a story that is not understood, take the time to understand it.

This idea of getting consensus without discussion sounds weird but does work. It can be used in other areas besides planning. Those less likely to participate will have input into the plan. There are many good ideas in your more quiet people. Common ground and differences become evident. The team can focus its energy on the differences and not waste valuable time on where they already agree. I have seen this positively impacted the team's story estimation velocity. Instead of spending 10, 20, or 30 minutes on each story, most stories estimates took a minute or so.

Acknowledgments

Thanks to Symantec Corp. for experimenting with this technique. Thanks to Lowell Lindstrom for telling me something similar a while ago that probably helped me come up with this and to Brian Button for warning me about potential pitfalls.

About the card deck

Attached to this paper is a set of cards. Print them on perforated 3x5 note card stock. One set for each player. You may notice there are some gaps. The deck is designed for unitless numbers or ideal programming days. As the estimates get longer, the precision goes down. There are cards for 1,2,3,5,7,10 days and infinity. This deck might help you keep your story size under 2 weeks. Its common experience that story estimates longer than 2 weeks often go over budget. If a story is longer than 2 weeks, play the infinity card and make the customer split the story. If you really feel compelled to play a 4, 6, 8, or 9, go ahead an use two cards at once. I bet that the added precision probably won't help a lot.

grenning@objectmentor.com