

An Introduction to Planning Poker

Bob Hartman, <http://agile.dzone.com/articles/introduction-planning-poker>

Planning Poker is an agile estimating technique which has become very popular in the last few years. It is based on an estimation technique known as Wideband Delphi which was created by the RAND Corporation either in the 1940s or 1968 depending upon which sources you find credible. The technique was refined by James Grenning in 2002. It was made much more popular when it was included in Mike Cohn's book "Agile Estimating and Planning." Although the basics of the technique have been around for many years the refinements by Grenning made it usable by agile teams and they have taken full advantage.

Planning Poker is extremely simple to play while also being accurate enough to use for agile planning. The basic rules are as follows:

- Each participant gets a deck of estimation cards representing a sequence of numbers. The most popular sequences involve doubling each card (0, ½, 1, 2, 4, 8, 16, etc.) or the Fibonacci sequence (1, 2, 3, 5, 8, 13, 21, etc.). The Fibonacci sequence is generally more popular.
- The moderator presents one user story at a time to the team.
- The Product Owner (or equivalent role) answers any questions the team might have about the story.
- Each participant privately selects a card representing his or her estimate of the "size" for the user story. Usually size represents a value taking into account time, risk, complexity and any other relevant factors.
- When everybody is ready with an estimate, all cards are presented simultaneously.
- If there is consensus on a particular number then the size is recorded and the team moves to the next story.
- In the (very likely) event that the estimates differ, the high and low estimators defend their estimates to the rest of the team.
- The group briefly debates the arguments
- A new round of estimation is made (steps 4 and 5 above).
- Continue until consensus has been reached and the moderator records the estimate.
- Repeat for all stories.

People who promote the use of Planning Poker understand some of the main reasons why it is successful. People like Mike Cohn have been very instrumental in pushing planning poker and even created www.planningpoker.com for people to be able to play planning poker with distributed teams. There are very good reasons why most agile trainers and coaches (myself included) promote its use.

My personal top 3 reasons planning poker is great are (all of these assume the team is using planning poker appropriately):

- Fosters collaboration by engaging entire team
- Creates consensus estimate rather than having a single person driving the estimate
- Exposes issues early through discussion of each user story

All of those are great reasons to use planning poker, but there are also some ways to help make it work even better. The three items listed above are all very obvious if you watch successful teams using planning poker. But there are small things most people miss which can be done to improve the process. Knowing of those things allows successful coaches to help teams get even better. The beauty of it is you don't have to be a coach, just someone the team trusts, to help the team get better. So what are these nuggets of wisdom?

Those who actually could do the work are the ones that vote. Too often agile teams have everyone vote even if they have no idea about the work involved in the story.

Managers don't vote. Managers are usually incented to have the work take less time, so they often skew the vote too low. However, managers have more experience than the average team member, so I usually give them veto power over the team consensus in one specific circumstance – they can ask the team if they have considered something which may INCREASE the size. They are never allowed to try to get the team to decrease the size. Their opinion carries too much weight and acts as an anchor to the size, dragging it down in direct proportion to how vigorously they defend their position.

When there is a tie in the voting between two sizes which are consecutive, just pick the larger size and move on. Remember, consecutive sizes might be 5 and 8 if you are using the Fibonacci sequence for sizing (1, 2, 3, 5, 8, 13). No one should complain about using the higher number and an equal split usually takes a long time to resolve. It also usually resolves to the higher number, at least in my experience, so let's use those facts to our advantage and move on.

Stop implementation discussions before they go too deep. Teams commonly drive down to the technical details when they are discussing a user story. This is fine to a certain extent, but it should be severely limited. No more than a one minute discussion. The people doing the sizing should determine in their mind the simplest possible solution and pick a size based on that scenario. It seems like more discussion will make the size more accurate, but in reality this just isn't true. The scale is not very granular so there is a distinct lack of precision. This is done partially to encourage teams to just make their best guess and move on.

Use an "I need a break" card. Too often teams will be working hard playing planning poker and not realize some people on the team need a break. Having the "I need a break" card allows someone to draw everyone's attention to this.

Use a timer of some sort to limit discussion. This is self-explanatory. I usually like to limit discussions to no more than one minute.

If consensus cannot be reached by the end of the third round of voting pick the largest size and move on. After two rounds of discussion further discussion usually does not yield great results for the time invested. By choosing the largest size the team has a chance to improve upon it, but they will not be in any danger of not having enough time. Not having enough time is a major problem the team is trying to avoid, so this particular short cut should not cause major issues.

Have the person creating the user stories meet with QA and Development leads prior to playing planning poker to make sure the user stories have answers to the most obvious questions the team will ask. This allows the team to focus on the size, not spend time gathering information.

Remember the baseline. Whatever the team picks as a baseline needs to be consistent from iteration to iteration. If an ideal day is a size 1 then all iterations need to use that reference point. If a particular user story is a size 1 or a size 3 then that needs to remain consistent across iterations. It can sometimes be helpful after a few stories have been sized to bring up the baseline and ask the team if they agree the sizes are truly relative to the baseline. Failure to do this could lead to what I'll call "size creep" over time. In other words the team slowly changes their baseline mentally to be either larger or smaller than it truly is. This usually shows up as the team not being able to meet their commitment for several iterations even though everything looks more or less the same.

Have fun! Playing planning poker should be a fun, collaborative exercise. Too many teams try to grind it out for an hour or two and forget to enjoy their work. There are many ways fun can be injected into the process. One I particularly like is to play real poker with the sizes. Every sized story counts as a poker card and every five stories makes a poker hand. Prior to starting everyone tries to pick which hand will be the best poker hand (i.e. pick hand 1, hand 2, hand 3, etc.). This encourages them to look at the user stories ahead of time so they can try to make a good guess about which set of 5 will make a straight or 4 of a kind. Winners can even get a small prize.