Jonathan Hirsch

1665635

Estimation

Estimation can be a very powerful tool that can have many drawbacks, but just as many benefits if used correctly. To have an estimate requires someone making one, which requires time and effort; time and effort that could have been put towards the actual project itself. There is pressure put on those who create these estimates to be accurate, and pressure on everyone else to meet this estimate. If projects take longer than estimated, then everyone gets upset and then developers are told to reduce quality in order to increase the pace which only makes things worse. But estimates are not all bad, there is a place for them and it must be decided if your work environment is such a place. It must asked “why are we doing an estimation”. Estimation are useful when making significant decisions such as project to start working on. If one team is not able to release a feature until a separate team is done with a new service, then it would be useful to know when that service is expected to be done to decide if the team should begin developing the feature now or put it off till later. Estimates can also force there to be better communication among team members about architectural directions, and design problems. There are many different ways to handle estimates for teams to choose from, such as using points.

A point is a “a subjective unit of estimation used by Agile teams to estimate User Stories” and are used to represent the amount of effort required to implement a user story. Stories are estimated using relative sizing which means they are compared to previous stories. This has proven to be more accurate especially with larger sample sets of stories. Points are decided by the complexity and usually is a factor of n^2 or a number in the Fibonacci series. These points should also be decided by those who are actually responsible for getting the work done. It is effective to provide best, likely, and worst case estimations for the points of a story.

There is also using “Bucket Theory” which estimates stories against each other, but not individually. A reference is created by picking a story that is small (but not too small) and call it 2. Relatively size each story against this new benchmark by only discussing the implementation details that directly affect its size. Then put each story into a bucket 1, 2, 4, 8 or 16 only. An 8 should be able to be delivered in one iteration and anything bigger should be broken down.