

Using Vertical Slicing and Estimation to Make Business Decisions at Adobe

by Peter Green • 0 Comments

As any regular reader of this blog knows, I advocate estimating product backlog items but only when actions will be made based on those estimates. A team shouldn't estimate just to make their boss feel better but rather when the estimates lead to actionable decisions. When I heard the following story from Peter Green of Adobe, I asked him if he'd share it with all of us. He kindly agreed. Here's his story.—Mike

I recently helped to facilitate a two-day planning session for an important initiative at Adobe that would require coordination across several component teams in multiple locations. Going in, my agile compromise-o-meter was pretty far pegged since I felt like the teams were facing several challenges to their ability to deliver value to customers rapidly:

- Vertical slicing would be difficult to accomplish within a single scrum team due to their cross-geography, component based team structure. For any readers unfamiliar with this term, see my blog post here: <http://blogs.adobe.com/agile/2013/09/27/splitting-stories-into-small-vertical-slices/>
- The Product Owners had written a fairly detailed requirements document that each team had been studying and breaking down into something they were calling “engineering stories”, essentially high-level component team tasks that would help achieve the end business value described in the requirements.
- There was a separate integration testing team, rather than having the integration testing included in the scrum teams' definition of done.
- There was no alignment in sprint boundaries or durations across the various teams, resulting in a lack of transparency when determining what features had met the definition of done at any given point.
- There was no common Continuous Integration system across the technology stack, resulting in a slow feedback loop one whether changes from one team had broken the code of any other part of the system.

Putting those concerns aside and taking a pragmatic approach, we decided to focus on a few goals for the session:

- Visualize the work and how it flowed across the various teams and where the dependencies existed.
- Seek opportunities to deliver “done” vertical slices (even across teams) as early as possible so that the integration testing team could be engaged early and frequently.
- Project some kind of timeline of delivery across all of the slices and teams.

We spent the first day and a half getting the work written on cards and up on the board in rough order of implementation, spacing it out to ensure that the order of development and dependencies were clear to all. We then took markers and started circling sets of cards that delivered a vertical slice. We were able to identify several slices of development and cut quite a bit of scope from the original requirements documents in order to identify a Minimum Viable Product, and put the rest of the backlog items in a prioritized list for future releases.

The second afternoon we did a bunch of estimating to turn the ordered list into a projected timeline so that we could make some projections for the business. At the end of that session, we had a high level of consensus on a date to deliver those MVP slices, several months in the future. We used the “fist of five” technique to gauge individual confidence in the projected schedule (see <http://blogs.adobe.com/agile/2013/12/17/using-the-fist-of-five-technique-to-gauge-confidence/> for more information on the Fist of Five technique). Everyone in the room was a 4 or 5 out of 5 with the exception of a few folks who were going to be (justifiably) uncomfortable with any projection of dates that far in the future prior to having started the work. I asked a few questions to test the hypothesis that the date was the earliest we could deliver that value to the business as we were currently structured, including things like “if a magic fairy bequeathed a bunch of new team members experienced in the technology upon us...”. The answer kept coming back to the same date, maybe with higher confidence if we made those changes.

So I turned to the overall product owner, and said “well, with the current product backlog, it looks like Aug-Sep when we'll get the “MVP”. We had a quick discussion about the fact that if new stories are discovered in the course of the work, or if anything takes longer than our guesses that we would need to push the date out further. She had a despondent look, one that she later told me was closer to nausea than despondence. She had already mentioned that she was not

looking forward to the conversation she was going to have to have with the VP about the longer-than-wished-for timeline.

The real problem was that the team's estimate project's goal was to open up a particularly important revenue stream that the business knew had customers waiting for them to deliver it. Those customers made purchases in a specific market window due to the nature of their industry, and the Product Owner said that the issue was that the current projected date for the MVP would miss that market window by a few months, making it nearly pointless to do the work in this fiscal year at all. Having participated actively in the breakdown and slicing for the two days, she understood why it would take that long; it just wasn't an acceptable business outcome. I asked her what she thought we should do about it.

At this point, we had an hour left in the two days before people started flying back to their home offices, etc. After a moment of silence, someone in the room stood up and asked: "what if we hired some contractors to manually do the work that would have encompassed slices one and two on the board, until we could get the software built to do it the right way?"

We all paused for a second, realizing we might have a path to satisfy the business, and a flurry of activity broke out. Everyone gathered around the board, started moving cards around, talking about how the new approach would work technically, validating that it was feasible. Having done the estimation and slicing, the teams were able to quickly re-plan and forecast that the "new MVP" plan could be delivered in time to meet the market window. The teams would build out the other slices to replace the manual portions directly after that, delivering in a similar time frame as the original MVP, but meeting the business needs and allowing the Product Owner to meet her revenue targets.

It occurred to me as a huge breakthrough, one that we would not have reached without the hard work of turning the requirements into vertical slices, visualizing all of the dependencies, and estimating the work.

There is often debate about the real value of estimation in software, given its inherent lack of predictability. There are arguments to be made that it is a non-value-add activity, what the lean community would call “Muda” waste. I think this experience highlights where estimation can be a value-add – when we need to make business decisions that are dependent on the work being feasible within a specific time frame to meet a market window.

I’m hopeful that such an approach will become more commonplace in the group, providing an opportunity for all of the right conversations to happen and breakthrough results to become the norm.

Posted: January 15, 2014

Tagged: product backlog, product owner, sprinting, teams, estimating, testing, customers, lean
