A Real Release-Planning Experience

Kevin Thompson, Ph.D.
www.kevinthompsonphd.com
January 7, 2015

I've been consulting for a Accela's Agile transformation, to help them implement a Scrum process for developing their Web and mobile applications. I thought it would be interesting to describe the nuts and bolts of their very first Release Planning meeting. First times are always rough, but they are also of interest to people who may be facing their own 'first time' soon.

Contents

Preparation 1
9:30 – 9:50 AM: Kickoff for the Release Planning Meeting 2
9:50 to 10:15 AM: Assignment to Teams 3
10:15 to 11:30 AM: Estimation 4
12:30 to 2:00 PM: Release Planning 8
2:00 - 2:30 PM: Closing the Meeting 12
Concluding Thoughts 13

Preparation

During my previous visit to Accela on October 16th, 2014, I asked them to prepare for the Release Planning meeting by

- Clarifying the Team definitions
    - What is the focus of each Team?
    - Who is the ScrumMaster for the Team?
    - Who is the Product Owner for the Team?
- Estimating the Velocity numbers for each Team's Sprints in the three-month Release cycle
- Preparing the Product Backlog Items (Epics and Stories, hereafter "PBIs") to be considered in planning
    - Write them
    - Estimate them
- Printing out the PBIs, one per sheet of paper
- Bringing
    - Blue masking tape, for drawing lines and attaching paper to the wall

    - String, to show dependencies between PBIs
- Setting up the Release Planning board
- Scheduling the meeting for the participants

9:30 – 9:50 AM: **Kickoff for the Release Planning Meeting**

The meeting was held on October 22nd. I arrived just before the 9:30 AM start time, to find Frank (the head of the PMO) taping row and column headings to the glass wall of the Boardroom conference room. That wall was to be our Release Planning board. That the wall had six columns of glass panes, exactly matching the six two-week Sprints of the Release Cycle, was pure serendipity.

The columns correspond to Sprints, and each Team's row is denoted by the Team name at the left. (Frank is at the far right, speaking to the group.)

The rest of the group soon entered. Approximately twenty people attended, including Team and Area Product Owners, ScrumMasters, and a scattering of Engineering and PMO personnel. Although I would have preferred to have all members of all Teams present, this was not possible, because

- At least half of the Team members were in Shenzhen, China. It was not possible to bring them to the US for this meeting, or practical for them to participate by videoconference.

- The Team definitions were still in a state of flux, so having the “Teams” attend was, in fact, logically impossible

Not all of the preparation work had been done. I was very impressed to learn that the Product Owners had written 170 PBIs since the last meeting, which was much more than I expected. Some of these had even been estimated prior to this meeting. All of these PBIs had been printed on paper, or hand-written on sticky notes.

Some things had not been done. I listed the remaining work to be on the white board, along with the Scrum Roles associated with them

1. Assignment of PBIs to Teams (Team Product Owner responsibility)
2. Estimation of PBIs, via Affinity Estimation (Teams responsibility)
3. Sequencing of PBIs based on prioritization (Team Product Owners)
4. Estimation of Teams’ Sprint Velocities (ScrumMaster responsibility)
5. Development of Release Plan (All)

Frank expected four teams for AA product, and two for ACA product. However, the Teams’ areas of responsibility had not yet been defined, so I suggested collapsing these into one bigger Team each for AA and ACA, with larger corresponding Velocity values. Otherwise the division of work into undefined Teams would cause more confusion than benefit. That division would have to occur later, when the Teams’ areas of responsibility have been identified.

I suggested using Affinity Estimation to get through estimation of many items quickly, since almost no estimation had been done. Affinity Estimation is less reliable than Planning Poker, but much faster, and thus more practical for this scenario.

9:50 to 10:15 AM: **Assignment to Teams**

The PBIs had been categorized into topical areas, rather than associated with any concept of development teams. Thus the first step was to divide the PBIs into different stacks for our new Team definitions, namely

- AA1-3
- AA UX
- ACA1-2
- API
- AGIS

This work did not take the Product Owners very long.

10:15 to 11:30 AM: **Estimation**

We discussed different options, and chose a particular approach that was driven by two logistical constraints:

- The number of PBIs required a fast estimation approach, and hence the Affinity technique.
- Concern over a possibly-lengthy process suggested tackling the first two Sprints-worth of PBIs, or about a quarter of the total, to ensure that we could at least plan the first month of the Release Cycle

The Product Owners divided the scope for the new Team definitions into High, Medium, and Low priority piles (i.e., for near term, medium term, and longer term time frames). Everyone agreed to estimate only the high-priority items for now, and revisit the other items later.

Different tables were selected for use in estimating the PBIs, by different groups of participants. Each group had a set of high-priority PBIs, more than they could easily lay out at once in the space available. Thus they divided their PBIs into smaller batches of roughly 10-15 items, and proceeded to use the Affinity technique to estimate each batch.

For the first step, I asked them to sort the PBIs on the table, in increasing order from left to right. This took a few minutes.

The next step was to clarify the units of estimation. Prior conversations indicated that everyone favored absolute sizing (time-based units), so we discussed the units and agreed to estimate in "man-days." A man-day, or person-day, is defined to be eight hours of a Team member's time, spent working on a PBI. Thus four people who work on Story for six hours each expend three man-days of effort. (Saying that something requires a man-day of work says nothing about the duration, i.e. the number of calendar days, required to do the work. Knowledge of effort alone provides no information about duration.)

Once everyone agreed on the units, I asked them for a rough conception of the size of the smallest Story. "Around twenty hours," was the answer. I suggested that this should be considered a "3" in the chosen units. I asked them to make columns on the table, labeled with the numbers of the Fibonacci sequence (starting with 3, which meant following numbers were 5, 8, 13, etc.), and then decide which PBIs belonged in which column. The result of binning the PBIs this way was an estimate for each.

In between helping the first group, I updated Frank on the progress and techniques, and he facilitated a second group's estimation session.

The groups estimated around 60 items in 75 minutes, of the roughly 170 items they'd brought to the meeting. An additional fifteen or so had been estimated by one Team prior to the meeting.

We took a one- hour lunch break at this point.

12:30 to 2:00 PM: **Release Planning**

After lunch, we began to create the Release Plan. One wrinkle to this process was lack of the right Velocity values. The Team definitions had changed from earlier expectations, so I walked Daman (another PMO person) through a quick Velocity estimate, using the formula

Velocity = (Team size) ×(Workdays in Sprint) ×(Focus factor)

I prefer more sophisticated formulas for use in Sprint Planning, but the longer time horizon of a Release Cycle entails so much more uncertainty about everything, compared to a Sprint, that a quick-and-dirty calculation is likely to be good enough.

The "Focus factor" is the fraction of a Team's Sprint that a Team spends working on PBIs. I suggested 60% as a good-enough number, which yields a Velocity of 42 man-days for a two-week Sprint with a Team of 7 people.

Participants put PBIs on the planning wall, in the order they preferred (based on business value), assigning them to Sprints in ways that satisfied the Velocity constraints of the Sprints. This first pass did not attempt to account for cross-Team dependencies.

For dependency planning in the second pass, Frank wanted to put a blue sticky dot on each item that had no predecessors (i.e., that did *not* depend on another item's completion), to ensure that every item had been reviewed for dependencies.

I suggested adding a different mark to any item that depends on another. Since no other colors of dots were available, we decided that a half-dot sticker (made by cutting dot stickers in half with scissors) would indicate items that had predecessors. (I suggested thinking of this approach as having a half dot signify that an item that is "incomplete," and depends on something else.)

Frank liked the idea, but decided to put a half-dot on both predecessors and successors. This way, the only items lacking a blue sticker were items that had not been reviewed for dependencies, and soon every item had a sticker on it.

Participants continued working. They identified missing PBIs, wrote the new PBIs as they went, and added the new ones to the right places on the planning board. They linked predecessors and successors with String to show the dependencies, and ensured that all successors were scheduled to start after their predecessors completed.

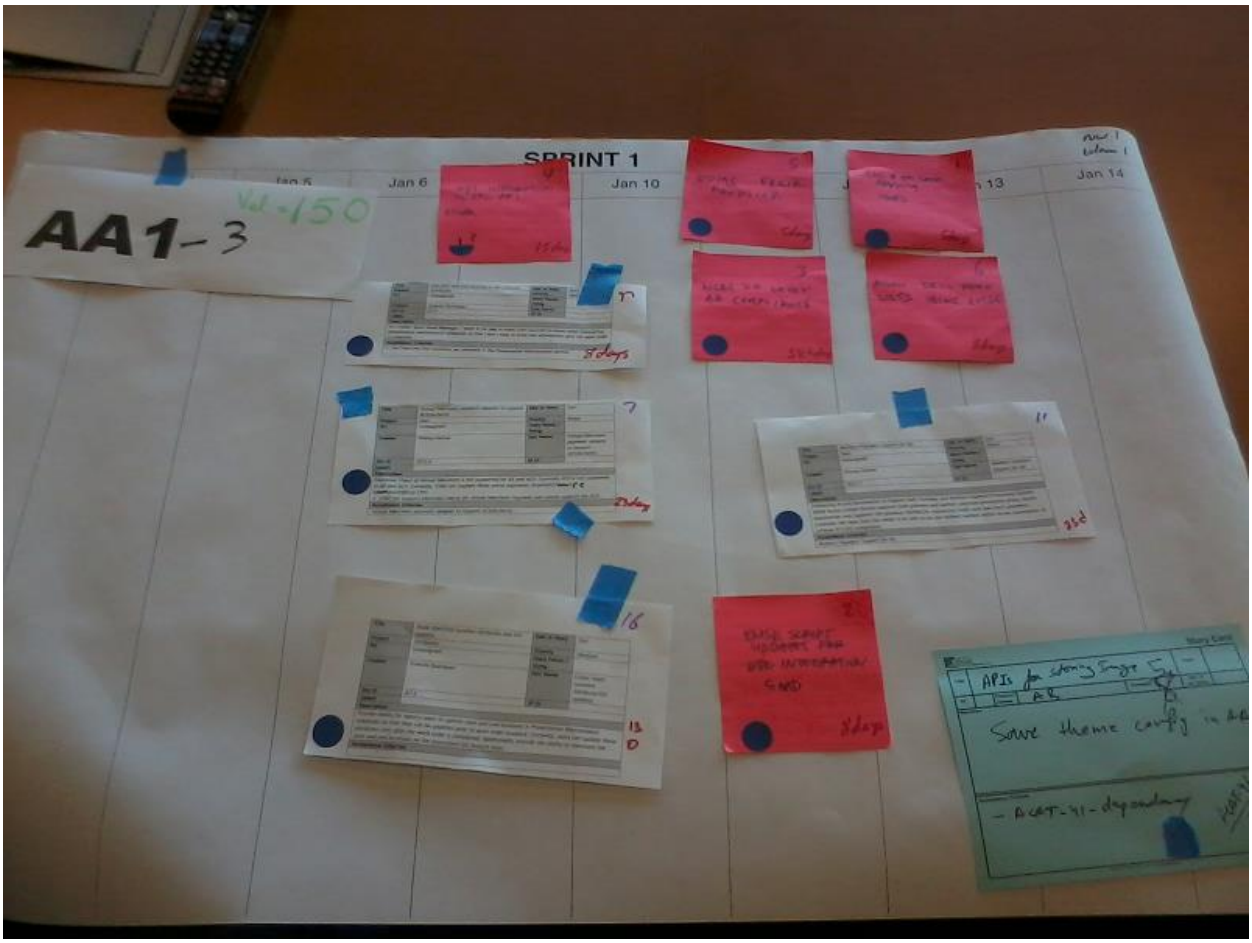
The completed Release Planning board contained 74 items, with only three cross-Team dependencies identified.

2:00 - 2:30 PM: **Closing the Meeting**

The conference room had to be cleaned by the end of the day, with all notes and tape removed, as it would be needed for a different meeting the following day. Frank planned to photograph the planning board before cleanup, and record the information in Jira and other forms, as needed. Kris, the VP of Engineering, suggested preserving the paper form (as an "Information Radiator") in the lunch room. Thus Frank first photographed the completed Release Plan, and then set about transferring the notes to paper Sprint calendars to be posted in the lunch room.

Frank also expressed the intent to re-visit and update the Release Plan after the first couple of Sprints had completed, and to incorporate the PBIs that we did not address today.

Finally, everyone agreed on the next step, which was for Product Owners to work with Teams to decompose Epics, write and estimate Stories, and prepare for the first Sprint (which was about two weeks away).



Concluding Thoughts

This Release Planning meeting was not perfect. Ideally, all of the Epics would have been decomposed into fine-grained Stories prior to this meeting. The Teams would have been defined, and the Stories would have been put into each Team's Product Backlog, ranked in value-driven order. Finally, all of the Stories would have been estimated, and each Team's Sprint Velocity estimated, prior to the meeting.

Obviously, many of these ideal things did not happen. Yet this company, starting Scrum for the first time, did indeed create a Release Plan. They succeed with their first Release Planning meeting with only one week's advance notice after Kris, the Engineering VP, set the date for the meeting. (The date was a reasonable one—we always have to start somewhere, and pushing the time out for better preparation does not necessarily yield better results.)

How was this possible?

I think the answer has two main pieces. One piece, obviously, was on-site guidance from someone who knew how to do Release Planning (me).

The other piece is easy to miss. A key to accomplishing this Release Planning meeting was the willingness and ability to make the best of an imperfect situation.

- PBIs are not estimated yet? Then we'll estimate them quickly now.
- PBIs are mostly Epics, not the finer-grained Stories? Then we'll use what we've got, which is mostly Epics, with a sprinkling of Stories.
- Actual Teams are not defined? Then we'll make "Jumbo Teams" for now, and refine later.
- Team members are split between US and China? Then we'll draft the plan here in the US, and get feedback from a review by the Team members in China.
- String ordered for dependencies hasn't arrived yet? Then we'll send someone to Target to buy string right now.

Sprint Planning is a routine and repeatable process, in which all of the Roles of the Scrum Team participate every Sprint. Release Planning is commonly described as similarly routine process, albeit larger. In reality, it may be far from straightforward, because far more people are involved. The size and geographic distribution of the workforce may make a single Release Planning meeting impractical. The key to success is to know the goal clearly (make a Release Plan), know the techniques required (e.g., estimation methods), and "inspect and adapt" to get the job done as well as it can be done.

The Release Plan created in this session was clearly incomplete. Details are sparse after the first two Sprints, due in part to the (correct) decision to limit the scope in order to get through the planning process in the time available. However, the plan will be refined over time.

Another common fallacy of Release Planning is the notion that the Release Plan is a firm commitment, or reliable description of how the product deliverables will evolve over the span of the Release Cycle. This is not generally the case, and to the extent that it happens to be the case, is very likely a bad idea.

The Agile world exists due to the need to adapt to the inherent uncertainty about what scope should be developed, and the amount of effort required to develop it. Our conception of the scope to be developed in, say, the latter half of a Release Cycle should be informed by what we learn in the first half of the Release cycle: What we learn from *building the product*, and what we learn from *changes in the business environment*.

A Release Plan should be thought of as a *forecast*, not a guarantee. It is the best picture of the future we can formulate at a particular moment in time. It is a useful thing, but we would be negligent if we did *not* refine that forecast during the Release Cycle, based on our best knowledge at each point in time. If we are diligent about our responsibilities, then those refinements will sometimes alter the Release Plan so drastically that it will bear little resemblance to its first draft, and that will be a good thing.

The participants in this Release Planning meeting are well aware that their Release Plan is not accurate, and that it will need to be improved over time. The best news is that they understand that this truth is even more important than the plan they created.