Team Estimation[1]

We have found it useful to size stories based on their relative complexity. More complex stories will take more effort to complete. Feature complexity typically relates to the degree of interconnectedness of a feature. This can be its own inter-connectedness or the number of connections to other features. Interestingly enough, it takes about the same number of words to describe a simple story as it does to describe a complex story.

Challenges in Estimation

Some of the biggest challenges we have in estimation include:

- Too much detail
- Doing design while doing estimation
- Too much precision
- A reluctance to commit to the estimate

We have to remember that estimates are really just our best guesses of the effort required based on our current information. We'll have more information later and will refine our estimate then, and as we go along.

Relative Estimation

We have found it easier to make estimates based on a value relative to another estimate. In other words, it is easier, and more accurate, to say "feature A will take twice the effort feature B will" than it is to estimate features A and B separately. A simple way to do relative estimation is with a game called "Team Estimation".

Team Estimation Game Play

The "rules" for team estimation are very simple:

- Place your story cards in a pile on the table. Take the top card from this pile and place it on the playing surface a foot or so away from this pile.
- The next player take the top card off the pile and places it relative to the first card based on size. She can place it to the left if she feels it is easier, underneath it if she feels it is of the same size or to the right if she feels it is more complex
- In succession, each play can either:
    - Play the top card from the pile as just described
    - Move a card on the playing surface (i.e., declare their disagreement as to the previously stated size of the story by moving it to a different column)
    - Pass

---

[1] As far as we know Steve Bochman invented Team Estimation.

- The above step is repeated until no more cards remain in the pile and no player wishes to move a card

Throughout this time, anyone is free to invoke others in conversation about why they are moving cards and what they think about the size of the stories. Remember, however, that we are just making estimates, so conversations to help clarify what the stories are is useful, but getting too hung up on the exact size of the estimates is not worthwhile.

The following diagrams illustrate how the story cards are arranged on the table. Figure 1 shows how the cards are laid out at the beginning.
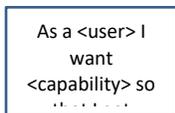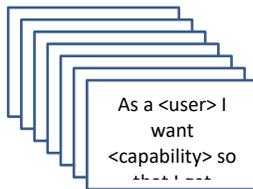
Figure 1. Story card layout at the beginning of Team Estimation

After a few cards have been played, the story cards might look like those shown in Figure 2:
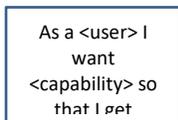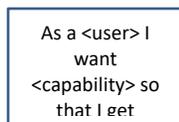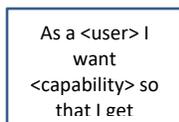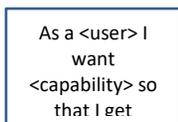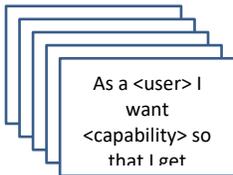
Figure 2. Story cards after a few have been played.

At this point, possible plays would be to:

- Take a card off the top of the deck putting it under any existing story, or creating a new column to the left, between any columns, or to the right
- Moving any story into an existing column or creating a new column to the elft, between any columns, or to the right

When you move a story to another column you are basically saying "this story has been misestimated." For example, figure 3 shows a possible play of a card (moving the card from one column to another). In this case, it means that don't think the card being moved is significantly larger than the stories in the leftmost column.
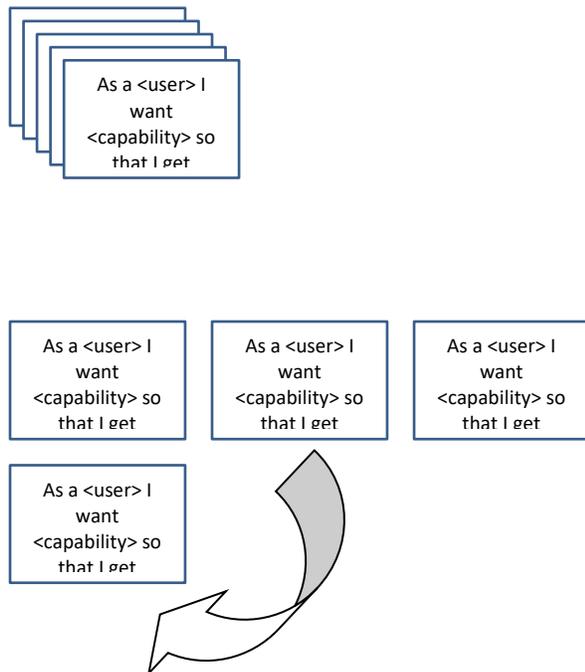
As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

Figure 3. Card play indicating that someone thinks a story is currently mis-sized.

Finally, we get to the point shown below:

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

As a \<user> I want \<capability> so that I get

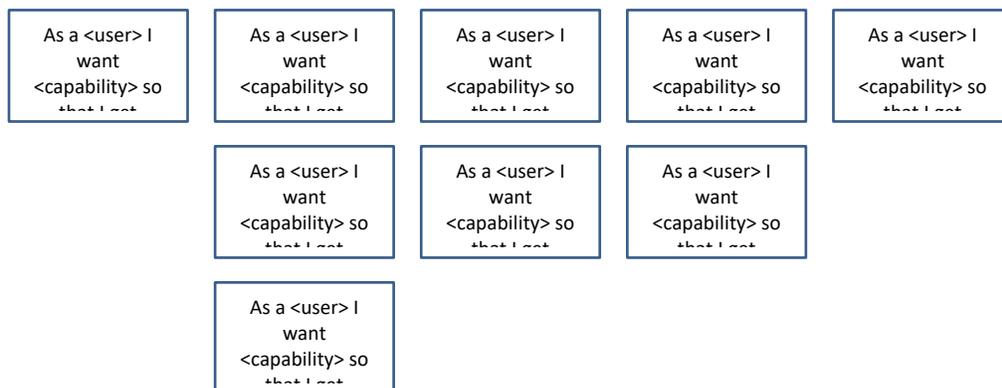As a \<user> I want \<capability> so that I get

Figure 4. Possible card layout after all cards have been played.

At this time, the number of story points for each column can be assigned. These should be based on what you consider to be allowable story points. We like using the sequence:

0, 1, 2, 3, 5, 8, 13, 20, 40, 100, 200, 400, 800

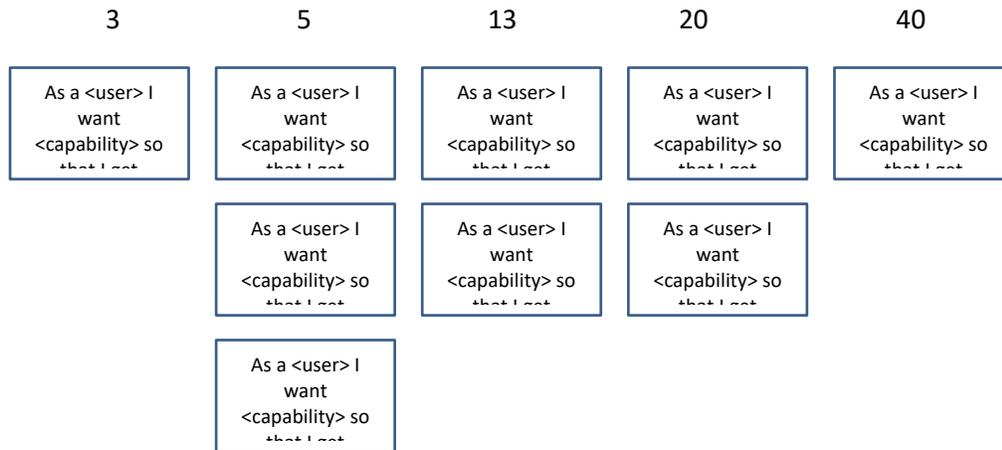Figure 5 shows labeling the columns with the estimated story point sizes.

3	5	13	20	40
As a <user> I want <capability> so that I get	As a <user> I want <capability> so that I get	As a <user> I want <capability> so that I get	As a <user> I want <capability> so that I get	As a <user> I want <capability> so that I get
	As a <user> I want <capability> so that I get	As a <user> I want <capability> so that I get	As a <user> I want <capability> so that I get	
	As a <user> I want <capability> so that I get			

Figure 5. Estimated story points to use for each column

After these are agreed upon, these story point sizes should be written on each card.

Team Estimation for MBIs (Minimum Business Increments) and Features

While we have described Team Estimation for stories, it should be clear we could use it just as well for MBIs and features.

Advantages of Team Estimation Over Planning Poker

Faster and More Fun

Prior to learning about Team Estimation, we had used a technique called Planning Poker. We have found that Team Estimation is faster, easier, more fun and that people tend not to get bogged down in too many details.

When Not Everyone Can Estimate the Entire Item

Planning poker requires that everyone can estimate all of the items. In reality, however, two variations sometimes occur:

1) Some people are needed to estimate parts of an item. Although there isn't the role of tester in Scrum, some teams have people who do more testing than development. In this case, those doing the testing may only be able to estimate the testing role. In this case it be necessary to estimate the testing effort separately.
2) When only certain people have the capability to implement certain items we can have them do the estimates within the context of the other estimated items.

Part of the advantage of Team Estimation is that it works better than Planning Poker when team members can't estimate all aspects of a story. For example, even if a tester can't estimate how long it might take to write a story, they can make a reasonable estimate as to the relative complexity of one story to another.