Table of Contents

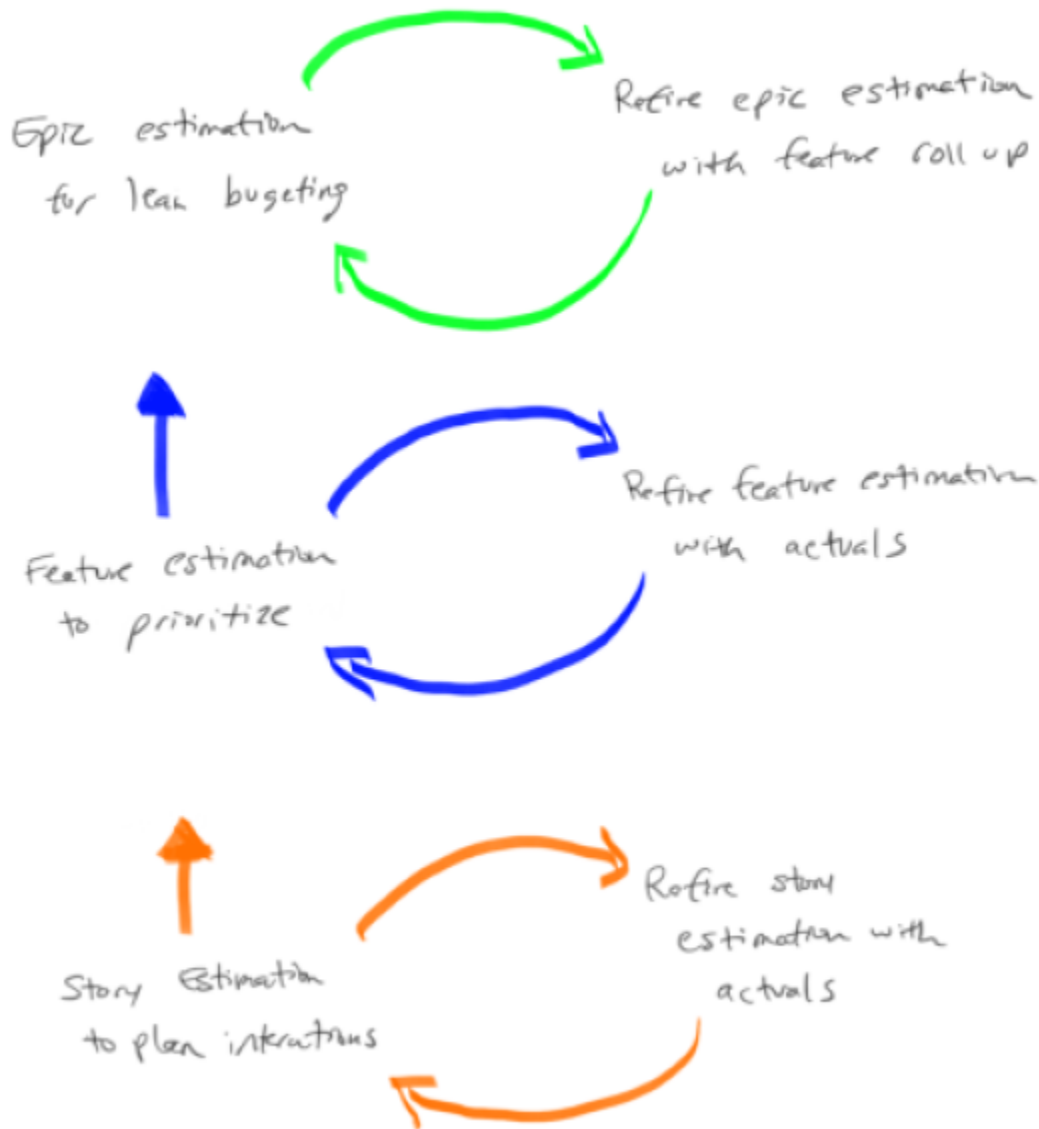# How Can We Scale Our Estimating Approach Beyond a Team?

Often, you need to provide estimates for items of work that are beyond the size of a typical Story, say an Epic or a Feature. For example, you might want to determine an overall road-map of intent and need to line that up with a calendar. This is only realistic if you have some kind of view of how long something will take. Or you might want to determine the Cost of Delay, which includes a sizing component in it, to determine what work you should schedule first to maximize the return.

The question therefore is "How do you go about getting size information for these bigger pieces of work?"

For the sake of discussion we are going to make a couple of assumptions:

- We are going to use the basic SAFe structure of Stories (less than two weeks), Features (less than a quarter), and Epics (more than a quarter). While the discussion is related to SAFe, there is nothing SAFe specific here. The scaled estimation problem is the same.
- We are using a relative team sized story point based estimating approach.

In general there is a different focus in the "why we estimate" question as we scale to epics and features. At the team level you are interested capacity to take on work. At the program level the main use is to prioritize the work. At the portfolio level its to help with overall budgeting. Pictorially this is represented as:

Epic estimation
for lean bugeting

Refire epic estimation
with feature roll up

Feature estimation
to prioritize

Refire feature estimation
with actuals

Story Estimation
to plan interations

Refire story
estimation with
actuals

(Note: Thanks to Steve Sanchez for the graphic.)

There are two basic ways to scale estimates:

- Pure Feature and Epic points which are used as an estimate: With this approach you use a similar
  process that you would use with Story Points. In other words you would use Team based relative
  size estimate on Features and Epics to generate "Feature Points" or "Epic Points". You might
  choose to define a new scale so that it is clear you are talking about Feature Points or Epic Points.
  For example, Feature Points might be 10X a Story Point, with a sequence that looks like 10, 20, 30,
  50, etc. In the same way that you track Team Velocity to determine how much you can get done,
  you can track closure of Features and Epics to reflect how much you can get done in these scaled
  environments.
    - Pros: In many ways, this approach is a "fractal" of the story point approach and benefits in a
      similar way.

- ○ Cons: The main downside of this approach is that it will take time to get useful (in terms of forecasting) data. Just like a Team has to wait an Iteration (a couple of weeks) to get their first data point, Features will take a quarter to get their first real data point and Epics will probably take longer still. Another weakness is if one class of work is significantly different from others it is difficult to get consistent estimates and capacity based on feature velocity.
- Summated Story Points to create Feature and Epic estimates: Based on a common understanding of the size of a Story Point, the idea here is that Features are estimated in terms of the number of Story Points it would take to complete the work, and similarly for Epic. So for example, if we look at a Feature and as we do the estimating we would say "this Feature is about the same amount of work as this other 60 Point Feature, so we'll call this a 60 for the estimate."
    - ○ Pros: One benefit of this approach is that you can start using data immediately. If, for example, you want to start managing Epics at the portfolio level now, you can quickly create estimates and understand capacity even if you do not have all the teams in place. The data you will have, while not highly accurate, will be sufficient to do capacity calculations and prioritization (see How Do I Facilitate a Prioritization Meeting? for an idea on how to get started).
    - ○ Cons: The main downside of this approach is that you need to have a common understanding of a Story Point. Many start with "a day equals a point" with the result this can quickly turn into a pure duration based estimate with all the problems that entails. Another problem occurs when one Team's velocity is significantly different (ie orders of magnitude different) from other Team's as, while each of the Teams can work their own estimates and velocity, when you summate very large numbers from one Team they can completely swamp other Team's numbers which means it is hard to understand capacity of the multi-Team Train.

In practice, the two approaches really are not that far apart from each other. What you will find is that even with the summated approach, you will end up with Features in the 10's (ie 10, 20, etc) and Epics in the 100's. The hard part is getting people to really do relative size estimates that include a view of risk and complexity at all levels. As a general note SAFe starts with the "Summated Story point" approach, but it is not that specific over the long haul. In fact SAFe assumes a relative size approach (not duration) as it says "Start with day is a point then never look back".

What this means is that in most places I have worked, organizations end up with the Pure Feature and Epic Points approach where Feature Points are the Fibonacci numbers times 10 (10, 20, 30, …, 130) and the Epic points are the Fibonacci numbers times 100 (100, 200, 300, …, 1300). The other thing that organizations do is limited the highest number to 13 or 20, no 100's etc, with the idea that this encourages people to split the work up if it gets to this level. So if there is a Feature Point estimate of "this is more than 130", the discussion is "Perhaps this is an Epic? Or perhaps we need to split the work so that it will fit in a quarter." This is a good discussion to have.

One final note on this. Many organizations I've worked with like to abstract estimation one step further by using t-shirt sizes for estimates. For many it is easy to say "in comparison to this small piece of work, this is a large"; it helps because since there are no numbers, you don't think about time. Once they have the t-shirt size, organizations usually settle on a mapping between these t-shirt sizes and Feature or Epic points. The following diagram shows a sample mapping that might be put in place:

Sample Mapping for T-shirt Sizes to Points (Your Mileage May Vary)							
Epic T-shirt Size	XS	S	M	L	XL	XXL	XXXL
Epic Points	130	200	300	500	800	1300	2000
Feature T-shirt Size	XS	S	M	L	XL	XXL	XXXL
Feature Points	13	20	30	50	80	130	200
Story Points	1	2	3	5	8	13	20

Note: Columns to the right of the Feature and Story scale are probably too big for the timebox and need to be split

Note that the mapping and the numbers would be validated to ensure that there is in fact a meaningful mapping between a feature we call "small" and the actual story points needed to complete that feature.

One final note. Some organizations I've worked with step back from an either or approach and work a both and approach. When little is know about the Feature, say when it is still being analyzed, they use a Feature point scaling based on S, M, L t-shirt sizing. They then equate the t-shirt size to numbers, so a S might be a 3 Feature Points, M might be 8, and so on. Then, as they learn more and understand the kind of work they have they move to a summated story point approach to Feature size; a second estimate, if you like.

Want to Know More?

- [Why the Fibonacci Sequence Works for Estimating](#) - Weber's Law approach from Mike Cohn
- [Why Progressive Estimation Scale is So Efficient](#) - Information theory approach from Alex Yakyma

Team, Estimates, Forecast, FAQ, Scaling, Points