

# Two Examples of Splitting Epics

by Mike Cohn • 24 Comments

## Two Examples of Splitting Epics

A Scrum trainer recently asked for a couple of good, real examples of large user stories (epics) being split into smaller stories. I thought it would be useful to share those examples here as well since the companies where these came from gave me permission to share them.

### Example 1: Selecting Marketing Campaigns

This first example is from a company that sells software to large retailers—think of WalMart or a company such as that. The target user is a vice president of marketing who is trying to figure out how to spend the ad budget. The user's big goal is this first story, an epic:

**1:** As a VP Marketing, I want to review the performance of historical promotional campaigns so that I can identify and repeat profitable campaigns.

The team working on this said it was clearly an epic and so I asked them to split it. They turned it into:

**1a:** As a VP Marketing, I want to select the timeframe to use when reviewing the performance of past promotional campaigns, so that ...

**1b:** As a VP Marketing, I can select which type of campaigns (direct mail, TV, email, radio, etc.) to include when reviewing the performance of past so that ...

[Note I'm numbering these stories in a way to make it easy to see which story came from which parent story. On a real project I wouldn't necessarily do that unless I were using a tool that did it automatically.]

I didn't know if 1a or 1b were large and should be split again. So I asked the team, "If I were to ask you to estimate these, what unit instantly pops into your minds—hours, days, weeks, months or years?" For 1a they said days, so we left it alone. For 1b they said weeks. We split it

up into:

**1b1:** As a VP Marketing, I want to see information on direct mailings when reviewing historical campaigns so that...

**1b2:** As a VP Marketing, I want to see information on TV ads when reviewing historical campaigns so that...

**1b3:** As a VP Marketing, I want to see information on email ads when reviewing historical campaigns so that...

and so on for each type of ad campaign.

Each of the stories above was reasonably small (“our estimate of these would be in days”). However, each needed one more level of detail, called a story’s “conditions of satisfaction,” which are essentially high-level acceptance tests. For 1b2 (the television story above), they wrote the following on the back of that story’s index card:

- See how many viewers by age range.
- See how many viewers by income level.

## **Example 2: Maximizing Hotel Revenue**

Let’s look at another example. This one is from a company I worked with that was in the revenue maximization business. The idea was that a hotel wants to charge as much as it can for its rooms. They set room prices based on a large number of factors such as what other hotels are charging, the time of year, any known local events (e.g., if the Super Bowl or World Cup are in town, the rates go up). Here was the initial user story, an epic:

**1:** As a hotel operator, I want to set the optimal rate for rooms in my hotel.

To keep this example a little simpler while still using real stories from this product backlog, let’s assume the hotel has only one type of room. As I’ve said, a number of data elements can affect the desired price of a hotel room. The general formula for pricing a hotel room then becomes this:

RoomRate = f (a,b,c...)

RoomRate is a function of a variety of factors. For each factor there was a story such as these:

**1a:** As a hotel operator, I want to set the optimal rate for rooms based on prior year pricing.

**1b:** As a hotel operator, I want to set the optimal rate for rooms based on what hotels comparable to mine are charging.

and so on...

Story 1a is really just saying that prior year pricing information is used in the RoomRate function. Story 1b says that what comparable hotels are charging is also a factor in the RoomRate equation.

These stories are interesting in that each was fairly large (an epic) on its own. It would be impossible to implement all of them in one sprint as there were many more than the two above. As each was implemented, the RoomRate function above would calculate the wrong price because it was being built first as:

RoomRate = f (a)

then

RoomRate = f (a,b)

then

RoomRate = f (a,b,c)

and so on.

So if after one sprint the calculation is RoomRate = f (a), the system is generating the wrong RoomRate (so it can't be shown to customers) but it is being calculated with the right math as far as can be done. Maybe values for (b) and (c) are hard coded or maybe they're just ignored. But working this way allows complex algorithms like that to be built up incrementally.

But because story 1b above is too big, here's how it was split:

**1 b1:** As a hotel operator, I can define a 'Comparable Set' of hotels. [Comparable Set was a term used widely in the company.]

**1b2:** As a hotel operator, I can add a hotel to a Comparable Set.

**1b3:**

As a hotel operator, I can remove a hotel from a Comparable Set.

**1b4:** As a hotel operator, I can delete a Comparable Set I no longer wish to use.

**1b5:** As a hotel operator, rates charged at hotels in a Comparable Set are used to determine rates at my hotel.

**Here were a couple of additional stories:**

**1c:** As a hotel operator, I want to set the optimal rate for rooms based on current projected occupancy.

**1d:** As a hotel operator, I can set parameters related to optimal rates such as target occupancy, minimum acceptable occupancy and maximum acceptable occupancy [could be more than 100%].

I find that learning how to split user stories is something nearly all teams struggle with at first. My experience is that sometime within the first year of doing it, things start to click into place and team members discover all the little local patterns for story splitting unique to their domain and types of projects. So, if you're relatively new to agile or user stories, stick with—it will get easier.

---

Posted: December 17, 2012

**Tagged:** product backlog, user stories, sprinting, teams, customers

---