

## 1.1 Managing Change (surfing the wave)

Because Agile uses prioritization as a key method for managing requirements and change, there isn't the same concept of change control in Agile methodologies as there is in traditional, Waterfall approaches. In Agile, while the scope is defined initially as best we can, there is the assumption that lots of types of change will occur. Most of this change is either (a) good (such as learning) or (b) unavoidable (eg, the current situation on Wall Street), so that those changes should not only be readily accepted but often welcomed. Still, change needs to be managed professionally, in part to avoid "bad" changes and in part to get the greatest amount of benefit or incur the least cost from these changes.

To manage change, we propose the following rules:

- Rule #1: When a change is identified of any significance, appropriate members of the Scrum Team will discuss it with the Product Owner (or, in his absence, his designee, such as a Business Analyst) to determine how significant it might be.

Changes can be of many types, some of which are indicated below:

- Notable in terms of increased or decreased business value but neither harder nor easier to implement. Still, might have led to some waste in terms of prior effort.
- Something that is clearly different functionality than previously expected. Perhaps with neither more nor less effort.
- Something that will cause a failure to complete the committed Sprint, or even to make the current expected next Release Date. Or, on the other hand, results in significantly less work and more can be done in the Sprint and possibly the expected Release Date can be brought in.
- Something that leads to higher (or lower) costs.
- Something that requires additional or different people. Or fewer.
- Something that could notably increase or decrease the quality of the product.

Clearly, discussion of the positive or negative impact of a change almost always has potential technical and business impacts, among others. So, we want both (or, really, all impacts) to be considered at the same time. The group will use good professional judgment in doing this impact assessment. It could be a one minute discussion if everyone feels the impact is small. At the other extreme, it could be a fancy analysis if the impact is very large.

- Rule #2: If the impact is very large... then the Scrum team and the Product Owner will discuss the impact with Stakeholders immediately. (This is rare).

- Rule #3: If the impact is large to medium... then the Scrum team and the Product Owner will decide on a course of action immediately and review this with the Stakeholders in the Sprint Review Meeting. And formal approval by the Product Owner and Stakeholders will be noted somewhere (eg, a note of "Approved at SR mm/dd/yy" in a spreadsheet). (Often one or two changes of this sort each Sprint.)

If the Product Owner feels that some Stakeholders will need to talk about the change at some length, he/she should use judgment about whether to hold some conversations outside the SR. Otherwise, the change might take the SR outside its reasonable time box. Still, the change should be reviewed with all parties (eg, including the Team) in the SR.

- Rule #4: If the impact is small... then the Scrum team and the Product Owner will decide on a course of action immediately. It is understood that the Product Owner and Stakeholders are, de facto, approving these small changes as they review the working software. No email or other annotations are necessary. (Always, there are many of these small changes in each Sprint.)

If the Product Owner (based partly on discussion with the Team), feels the cumulative impact of these small changes is notable, he should discuss this with the Stakeholders in the SR.

- Rule #5: Every Sprint, the Product Owner will update the Release Plan and, possibly, the Product Roadmap (eg, including future releases).  
This will be done with the participation of the Team. In some Sprints, there may be no changes to the Release Plan (other than Stories being completed in each Sprint).
- Rule #6: Periodically, the Product Owner must meet with the Chief Product Owner and the Stakeholders to review the Release Plan, Product Backlog and Product Roadmap.  
The frequency of these meeting will vary. Start by doing them each Sprint, and see what frequency is needed. In these meeting, the CPO (and others) give feedback on, revise and approve the then current Release Plan, Product Backlog, and Product Roadmap. A notation is made of these meetings.  
Note: All 3 “things” may be expressed in one spreadsheet tab. So, 3 names does not imply volumes of documentation. But does imply lots of hard thinking from multiple people.
- Rule #7: In every Sprint Review, the Product Owner will lead at least a brief discussion of the Release Burndown (or burnup) chart. And discuss the implications.  
Any party may comment or ask questions. Sometimes this will lead to a discussion of the Emergency Procedure (see below).
- Rule #8: Periodically, the ordering of the stories in the Release Plan and the Product Roadmap (eg, including later releases) will be reviewed with all parties (eg, the Team and the Stakeholders).  
Notation will be made of these meetings, and it will be understood by that notation that all parties are approving the then current Release Plan and Product Roadmap.  
The Product Owner and Stakeholders must agree on how often this should be done. The frequency will vary based on many factors (eg, how long to the next Release, the nature of the project, the degree of change occurring, etc.).  
Often these Release Plan reviews can occur in a Sprint Review (or Sprint Planning Meeting).

#### Key Questions:

- Could we have identified these changes earlier and made things better for ourselves? (Ask the 5 Whys.)
- Are we learning as fast as we can?
- Are we adapting to good/necessary changes as fast and as well as possible?
- Are we minimizing the negative impact of changes? Are maximizing the positive impact ? (Is our lemonade as good as possible?)
- How can we identify bad changes better? And reduce their occurrence? (eg, people sneaking lower value things into recent Sprint, disruptions, gold plating, the Business side not identifying changes as timely as they might, the IT side not identifying technology risks as soon as they might)
- Are we keeping ourselves in a balanced position (emotional, business, etc.) given the degree of change we are experiencing? (given the world, normal human error, human learning, etc.)
- Are we maintaining high motivation amongst all the key parties given how we have adapted to change so far?

#### Emergency Procedure:

Assuming we are acting professionally with change, it will still happen that from time to time, our Release Burndown chart (or burnup) will show that the project is in a dangerous position in meeting our expected Release Date. In this case, the Product Owner must decide when to execute the following Emergency Procedure. The Product Owner must lead discussions amongst the Team, the Stakeholders and appropriate managers regarding the following:

- a. Remove Impediments. So that team velocity can increase. Either purely within the Team or by going to managers to approve fixing an impediment “now” .

- b. Add People. Either by starting up a new Team (if there is enough time) or by finding outside “experts” who can do some of the Team’s work much faster.
- c. Reduce Scope. Either by taking Stories out of the release or putting some stories on a diet.
- d. Abort the Release Date. And re-plan against a new date, based on better information.

Often, when push comes to shove, managers will approve removing impediments that they would not approve removing earlier.

Often the best business decision (given what is possible, feasible, realistic and all the other trade-offs), is to make small adjustments in all four areas. At the other extreme, sometimes an adjustment is made in only one of the areas.

The Product Owner (and ScrumMaster) should have an early discussion to assure all parties who would likely get called into Emergency Procedure discussions are basically familiar with the concepts. At the time of actually executing it, emotions are typically high, which is not a good time to introduce the general framework.

Content by Joe Little, based on an earlier draft by others.