Assignment 6 (21 March 2019)

Sarath Mahidhara

1. Do user stories replace a requirements document?

Although user stories and requirements document seem like interchangeable words and look like they serve the same purpose in a project, they are quite different and involve a different approach in building a software and how they actually work.

User stories are a short description of the functionality written from a user point of view. Its major focus is on why and how the user interacts with the software. User stories are usually written in simple language and follow a universal template that is “As a <type of user>, I want <a desired outcome> so that <some reason>”. This is a high-level description of a functionality and needs a lot of further clarification and questions answered.

A traditional requirement describes how a software should act. The requirement document is written in detail on how the whole or particular area of a software should work. Requirements document also include important aspects of the project like scope and risks.

In my opinion, user stories will not be able to replace a requirements documents because they are quite different in many aspects of how and what they are intended for. Each of them has its own importance and advantage according to the methodology used to build the software be it agile or waterfall. Both of these have certain disadvantages but at the same time they also act as advantages. For example, the requirements document has every detail laid out upfront which helps the developer face less ambiguity and more clarity on what to develop while the user stories are high level and require a lot of questioning and further analysis during the development process which leaves room for improvement and refinement while there is little to no possibility for a developer to make changes to the requirements document.

1. When are user stories written?

As you can see in the [Disciplined Agile Delivery (DAD) life cycle](http://www.ambysoft.com/essays/agileLifecycle.html) of Figure 1, there are several distinct phases or seasons in the life cycle. Figure 2 depicts the [AMDD project life cycle](http://agilemodeling.com/essays/amdd.htm), which calls out modeling activities during the delivery life cycle. There are three common times when stories will be worked on/written during an agile project:

1. Inception. You often create a stack of user stories during [Inception](http://www.ambysoft.com/essays/agileLifecycle.html#Iteration0) as part of your [requirements envisioning](http://agilemodeling.com/essays/initialRequirementsModeling.htm) activities to identify the scope of your system.
2. Construction. During construction iterations you will identify new stories, split existing stories when you realize that they're too large to be implemented in single iteration, reprioritize existing stories, or remove stories that are no longer considered to be in scope. The point is that your stories evolve over time just like other types of requirements models evolve. Furthermore, enhancement requests may be identified by your [support staff](http://www.enterpriseunifiedprocess.com/essays/operationsAndSupport.html) during the [production phase](http://www.enterpriseunifiedprocess.com/essays/productionPhase.html) and then forwarded to a development team as they are working on an upcoming release. These enhancement requests are effectively new stories (albeit in a different format).
3. Transition. Sometimes new stories will be identified during the [Transition phase](http://www.ambysoft.com/essays/agileLifecycle.html#Release), although this isn't very common as the focus of release is on hardening the system and not on new functionality. But it does happen, and these stories would be prioritized and placed on the stack in priority order just as you normally would.

Figure 1. The Extended DAD lifecycle.

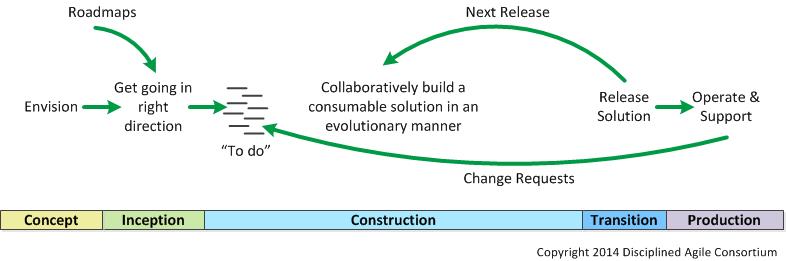


Figure 2. The AMDD lifecycle: Modeling activities throughout the life cycle of a project.

