

Overlooked during #Agile



Written By Clay Rehm, CCP, PMP www.RehmTech.com @ClayRehm
@RehmTechDotCom

A very important area that is overlooked within Agile is the Business Analysis and Requirements Gathering tasks and deliverables. There is a very common misperception and misunderstanding when it comes to documentation/artifacts and Agile.

Some of the misunderstanding comes from some Agile instructors, but I believe it also comes with people's interpretation of the Agile Manifesto. Of the values in the manifesto, it includes "Working software over comprehensive documentation". The manifesto also says "That is, while there is value in the items on the right (documentation), we value the items on the left (Working Software) more."

It does NOT say that the team should not create requirements or documentation. The manifesto values working software over artifacts. This is understandable, since working as intended software is the goal.

However, we cannot forget what needs to happen after the project is over and the working software is in production. And we cannot forget that the working software needs to address the business need.

There are the following items to consider in Agile and documentation:

1. Business Requirements and Business Modeling
2. Describing the Behavior
3. Supporting the application after it goes live

Business Requirements and Business Modeling:

This is an area that I think as project teams we fail or do a lackluster job. First off, the Business/Stakeholder requirements must be elicited, captured and agreed to. They do not address the solution. They are not technically written. They are easy to read statements that address WHAT the business needs, and WHY they need it.



To supplement the business user understanding of the project, current and future state models should be created depending on the complexity and size of the project. This at a minimum should include Relationship Maps, Context Diagrams and Business Process Flow diagrams.

Behavior:

While many Agile teams may favor User Stories instead of Use Cases, I would argue that Use Cases are far superior for capturing the behavior of the proposed system. In addition, you can use User Stories to build and maintain your Product Backlog, while at the same time building Use Cases to document the behavior and functional requirements. The team should also document the non-functional and transition requirements in an iterative fashion.

Ongoing Support:

If you are not willing to create any documentation or artifacts during your Agile project – how will you support it going forward? Will you need to ask the developers to look at the code each and every time before any enhancement? And, a User Story hardly provides the level of ongoing support that is needed for the life of the application.

In summary, the Requirements process cannot be overlooked in Agile. The deliverables I described above can be created in minutes and updated iteratively during the course of the Agile project. Watch for future posts from me regarding more details on the processes of how to create appropriate artifacts (documentation) for any project.

@PMIMadison #BusinessAnalysis #Agile #theAgileBA