CS 250 Module Five Journal

Dylan Cavazos

October 1st, 2023

As one of the core developers for the SNHU Travel program it’s important to prioritize work ahead of time to have an accurate estimation for completion of our assigned tasks. Considering whether the Product Owner wanted to reevaluate some of the details regarding the project, I propose a structured approach to address these changes during the development process.

First, I will ask the product owner if we need to go back and gather additional requirements to accommodate the proposed changes. Given that these changes could significantly impact ongoing sprints, it’s imperative to reassess our project’s trajectory. While code refactoring is inherently a normal process for agile development, is the product backlog going to be reassessed to meet a new deadline for the change? If the product backlog is being redefined, are our user stories also going to be modified to reflect the updates?

Due to the overhaul of the project to meet the new demands, code refactoring will need to be prioritized to compensate for the shift in development, as well as the quality testing to ensure the new requirements are being met. Also, how will this change affect the latent integration of the code, or is this phase being postponed until the other phases have been satisfied?

To ensure I get the response I need; I would write a formal email to both the product owner and the Scrum master. I would bring up these questions and concerns during the daily scrum as well, and if others are on the same page there is a higher chance that these questions or requests will be acknowledged by the product owner. By sufficiently outlining the potential problems with a sudden shift in development, I could easily express the new requests to the product owner and testers.

While suddenly shifting focus towards development on new requirements isn’t new to the world of agile it is certainly challenging. However, due to the failsafe built around the agile methodology and its constant refactoring it makes sudden shifts easier to handle. Earlier phases are easier to reevaluate as opposed to needing to start the entire cycle all over again as you would with Waterfall methodology. Since the definition of “done” is purely speculative and the goalpost for a “potentially shippable” product is always changing, this makes for a far superior and flexible environment to incorporate any last-minute changes into the development process. Attempting a complete requirement overhaul in a Waterfall methodology would be a monumental challenge, if even possible at late stages of development.

Dear Christy, and Brian,

I need your assistance in addressing some of the recent changes for the SNHU Travel Project. Previously, I’ve made an informal request regarding what is required for these new changes to be successful and how we will implement them into our current development phase. For instance, I need guidance on how this will affect the current phase of development and if we can expect our deadline to be shifted out to meet the new requirements. Not only that but what can we expect to change regarding the current product backlog and user stories that we were assigned. Are we moving to a previous phase to meet this sudden shift, or are we simply moving forward with the current material and focusing on integration and revision? Thank you for taking the time to read and respond to this email, best wishes!

-Dylan