Tammy Hartline

CS-320

04/30/2023

1st Journal Assignment

Software Testing Stage in the SDLC

The Software Development Lifecycle consists of six stages, planning, defining, designing, building, testing and deployment. While it may appear that software testing is one of the final stages in the SDLC, it actually begins, or at least should begin, much earlier on, within some of the beginning stages of the lifecycle. The reason to at least begin conversations about testing in the planning stage are vast, but one reason is to ensure that when meetings about deadlines and project specifications are underway, going ahead and generating a testing plan, and idea sharing can help ensure the testing stage does not get overlooked later on in the process, or pushed to the back of the task line, resulting in a high pressure work environment, leading to less than efficient testing measures due to deployment deadlines.

The more courses I complete, and research I do, the more I think software development teams should look deeper into test driven development. While I understand it is a complex process to learn, and takes some out of the box thinking, I think this skillset would provide a much more efficient product when it is released into production.

Outside of using test driven development, it is absolutely not impossible to follow the SDLC and end up with a quality product. This typically is only possible though, when you include software testing in the original planning stage, and then begin testing as soon work *items* are complete, versus waiting for the software testing in stage five to plan and perform the tests on an entire program/product. When stage two is in progress, it is important that all client requirements are defined, but also well documented. This documentation can then be reviewed by the software testing team members, who can plan ahead and begin conceptualizing how they will create the products acceptance testing. The same can be done for the designing and building stages as well. One important aspect that is often overlooked in the development process is considering and more importantly, *documenting*, ALL requirements and design elements.

By the time we reach the software testing stage in the SDLC, if done properly and with consideration, testers have already designed, configured, and been using their created tests for the product as each requirement was developed. Incorporating the testing process into each stage of the life cycle is one way to ensure the client ends up a superior product, that functions and operates in the way they requested it to. Otherwise, products get deployed that are likely to fail given something common, such as a user entering the incorrect data type, which was skipped over during the testing phase for example. While most may see the software engineering or developing as the two most important roles with programming, I believe more emphasis should be made on behalf of software testers, as it is their hands that are responsible for ensuring that the client’s needs and expectations were met and or exceeded before being released into production. I think a good comparison is that of an architect, contractor, and builders who are involved in a new project for a bridge. While each role is very important, and they all contribute, we are all human. A builder could have forgotten to add a support beam somewhere that was crucial for the structure’s stability, or the contractor could have purchased an incorrect material, or even the architect could have forgotten to consider what may be needed for additional structural support due to the bridge being in an area that has frequent landslides. There are so many scenarios where the project could fail, but ensuring a seasoned and detail-oriented safety inspector was involved throughout the entirety of the project, could keep them from loosing 100’s of thousands of dollars by catching these design and structural errors early on in the project and will ultimately conclude with a much more superior product as the end result.

**References**

Software Testing: An ISTQB–BCS Certified Tester Foundation guide, Fourth edition, Pages 64-66