CS-320-T4208 Software Test Automation & QA

Module One Journal – The Role of Testing in the SDLC

Computer Science Department

Southern New Hampshire University

Eric Farkas

3/5/2023

Testers are ultimately responsible for the quality of software developed and deployed by their teams. Although the role is often summarized with terms like manual testing, automated testing, input/output validation, and risk mitigation, the tester’s responsibilities exist in all aspects of development. In an agile team, the tester will participate in project kickoff and provide input on feasibility and methodology after the user stories are presented. The test plan will often be created well before any development is started. This early input ensures developers do not create code or integrations that cannot pass testing due to ambiguity or misinterpreted specifications. As the test plan is defined, testers will determine tasks that can be automated and will create the scripts necessary to execute the tests. The test criteria and results must be documented, adding documentation to the tester’s responsibilities. Some results (even expected ones) may require further clarification by the product owner or customer to ensure the results meet all expectations. This adds communication and collaboration elements to the tester’s responsibilities. Although following industry best practices and company style requirements primarily falls to the developers, it is common practice for the tester to provide a second look as an aspect of quality assurance. To summarize, the tester’s role in an agile team is easy to underestimate. The role is much more involved in development than the name suggests. Still, it also requires a passion for quality, consistency, and relentless persistence to ensure the system provides the expected results. These characteristics often require out-of-the-box thinking and strategies to accomplish within the project’s time constraints.

**What occurs during the testing stage of the SDLC?**

The main task in the testing stage is validating that the development code meets the requirements and does not contain any significant bugs. Testing may be manual or automatic and it is common to utilize both methods. It is important that the system meets the functional requirements and that the outputs are consistent with expectations. The testing stage should identify and report defects and inconsistencies in the application. Testers in an agile environment must be diligent, as requirements may change throughout development. Documentation is a key element of testing. Documentation includes updating the test plan with results, adding regressions as the development progresses, and refining the test plans to accommodate specification changes or strategy changes by the development team. Automated test scripts may also require updating as part of the documentation phase.

**Why is the testing stage vital to a successful SDLC?**

The testing stage is important because it enables high-quality and reliable applications to be released. Likely, end-user requirements would not be consistently met without testing. It is also likely that the released product would suffer from the discovery of bugs by end users. This would increase the likelihood of exploitation of these vulnerabilities. Bugs found after the release are more costly to fix than ones found during development. As most applications continuously evolve, eliminating bugs during development reduces maintenance costs for future phases and features.

**Are there any exceptions in which the testing stage would occur earlier or later than it typically does in the SDLC? Explain.**

There are always exceptions, especially with agile development. It is important to understand the risk that flexibility and making exceptions add to the project. Agile pulls testing forward to overcome the long and costly cycle present with waterfall development. Testing starts early in the SDLC process. Some might say that questions on user stories and development strategies during the kickoff meeting is a type of testing. The test plan is normally in place before the first code sprint is delivered. Automated scripts are developed before the code they’ll challenge as well. Where there is leverage is later in the development. While not advisable or considered best practice, testing can be delayed to combine the results of multiple sprints or to test integrations between systems. It is also necessary to sometimes add rounds of user tests led by a tester to gain feedback and refine code as the final deliverable nears release. It may also be necessary to add regression to already completed testing if a requirement change affects the data values, type or attributes.