Michael Ladderbush

Journal 1

During the testing stage of the SDLC, software is evaluated to identify and fix errors. This stage involves test planning, designing test cases, executing tests and user acceptance testing. It ensures the software meets quality standards, fulfills requirements, and is stable before deployment. The testing stage is so important for the SDLC because if a system is not rigorously tested then there can be no confidence that the other stages of the SDLC were completed successfully. If a system was released without testing then the following steps in the SDLC could also be greatly affected. Deploying a system that hasn’t been testing might be impossible altogether or could be on course for a major failure that could potentially harm the client or developer. Furthermore, maintaining a system that has never been tested could prove to be a futile effort as tracing the route cause of issues and correcting them may require major refactoring that would set back developers and negatively affect clients.

There are exceptions for when the testing stage could occur in the SDLC. When a system is being designed and initial created earlier in the SDLC it may be useful for developers to test certain components and their integration with other components. If there are issues with combining certain aspects of a design it could be beneficial to know of those incompatibilities ahead of time by testing early. While it is usually always best practice to test before deploying a system it may also be beneficial to test certain aspects of the system after deployment and during maintenance to account for future needs or unforeseen issues. Because testing cannot always account for all user cases there may be a need to include testing after deployment once those cases are revealed by actual users.