

SIT725 Assignment 6.1P

Student Name: Kamal Acharya (Tikaram)

Student ID: S224517147

1. Unit Testing

Unit testing is considered one of the integral practices in software development, and it is majorly concerned with individual components or units of source code verification. A unit normally consists of a function or method, which is isolated from other parts of the code and tested in order to make certain whether it works as it is supposed to perform or not (Myers, Sandler and Badgett, 2011). Mainly, unit testing ensures that a small piece of code behaves as intended with certain input that returns a particular output. In the process, bugs can be found and fixed early on in development, reducing the probabilities of defects later on in the product's development cycle. These would include, but are not limited to, unit testing in a web application-form validation or data processing functions within an application-to ensure those handle various inputs correctly.

2. Integration Testing

The integration testing takes place when the performance of a unit test is fully operable. This kind of testing ensures that multiple modules or components of an application behave adequately with each other. Emphasis is laid on how different units interact with one another so they can be integrated into the system. In a web application, AIM, for example, integration testing would be carried out on the interaction between frontend and backend services, such that when correct data passes from view to database and vice versa, validation is carried out. It becomes possible for the various integration problems in different parts of the application to be seen by running the developers' tests and quit being unseen in the unit tests.

3. End-to-End Testing

End-to-End (E2E) testing resembles the most comprehensive approach in testing, since it encompasses appraisal of the complete application flow from its state of commencement to finalization. This specific type of testing reflects actual user experiences, hence testing whether or not the application functions effectively within the actual setting (Burnstein, 2003). E2E testing typically covers a wide range of user experience in terms of navigating through the application and performing certain activities, such as form submission or transaction processing. For instance, an E2E test associated with an e-commerce application will make the user mimic browsing through products, adding items to a cart, and finally making a transaction. The concept of testing level is important because it assures that the interaction among different components in a live environment will not give rise to issues that might compromise the reliability and satisfaction of the application for the users (Meszaros, 2007).

4.Conclusion

One of the principal stages in developing dependable and quality applications is software testing. For instance, one can detect early-stage defects in code using unit tests. During integration testing, it's guaranteed that segments of that application interact well. Generally, end-to-end tests contribute to the integrity of a completely implemented system through emulating natural environment users' experiences. Overall, these proactive testing strategies allow developers to provide solid, reliable, and easy-to-use Web applications with fewer numbers of defects to the consumers.

References

- 1.Myers, G.J., Sandler, C. and Badgett, T., 2011. The Art of Software Testing. 3rd ed. New York: John Wiley & Sons.
- 2.Burnstein, I., 2003. Practical Software Testing: A Process-Oriented Approach. New York: Springer.
- 3.Meszaros, G., 2007. xUnit Test Patterns: Refactoring Test Code. Boston: Addison-Wesley Professional.
- 4.Kniberg, H., 2007. Scrum and XP from the Trenches. London: C4Media.