CSCC01: Reading Summary
Kalindu De Costa 1001392619

Analyzing the Evolution of Testing Library Usage in Open Source Java Projects

1. Summary

This paper summarizes the development of testing-related libraries in the testing of functionality of the software product automatically and efficiently. Primarily Java, contains many such libraries and this paper analyses some of the most widely used libraries for testing.

Zerouali and Mens analyze the usage of eight testing-related libraries in over 4500 open source Java projects that are hosted on GitHub. Mainly the frequency of libraries over time is analyzed along with of and when a library usage is replaced with competing libraries and the popularity of certain libraries over the others. It is also shown that most libraries are used along with others to complement each other and it is observed the permanent and temporary migration between competing libraries.

Lastly, this papers findings are given as a way for recommendation tools that allow project developers to choose the most appropriate library for their needs, and to be informed of better alternatives.

2. Findings

Zerouali and Mens analyzes 8 popular testing libraries over GitHub hosted Java projects, with Junit being the most popular and prominent testing library. Some libraries such as PowerMock which is an extension of Mockito and EasyMock are observed to complement and reinforce Junit. While libraries such Junit versus TestNG and Mockito versus EasyMock are in competition with each other. With the projects that have undergone library migration, in which a project replaces one of its used libraries by another, most project migrations are mainly permanent. For instance, projects tend to migrate from EasyMock to Mockito and Hamcrest to AssertJ, where these migrations only going one way around. However, in very limited cases, the library migrations were temporary and later reversed back to their original project.

2.1. Future work

Although the library migration and testing capabilities are thoroughly analyzed, Zerouali and Mens claim it requires a more in-depth analysis. In the future Zerouali and Mens plans to consider the effect of the specific library version on the migration phenomenon. Further, the preliminary results to provide recommendation tools or dashboards are aimed to be extended for a larger data set. Lastly, further analysis and data on library frequency and library migration between different libraries as well as upgrading of library versions. Most of these analyses is aimed to reduce this effort and provide tools and support for such actions.

3. Thoughts

The evolution in testing libraries have drastically improved and eased the Java project testing-related activities as described by Zerouali and Mens. In my personal experience I widely use Junit as a testing library over any other along with libraries such as Mockito and JMockIt to compliment the testing process. Current tools such as Maven allows quick integration of testing libraries to Java projects and this paper provides some very meaningful insights into common practices, techniques and reasoning for certain testing libraries and testing habits.

References

1. Zerouali, A., & Mens, T. (2017, February). Analyzing the evolution of testing library usage in open source Java projects. In Software Analysis, Evolution and Reengineering (SANER), 2017 IEEE 24th International Conference on (pp. 417-421). IEEE.