Homework 7 - Planning an experiment
HT2023

Casper Kristiansson

November 14, 2023

1 Research Question

In the field of software development, an interesting question has constantly been asked which is the difference between open-source and proprietary software development models. Therefore the question "Comparative Study of Open Source and Proprietary Software Development Models" is extremely important especially for software development companies to understand both the drawbacks and benefits. While this is an interesting subject there is a lot of existing research already on the subject like "Comparative Study: Proprietary Software vs. Open Source Software" [2] by Gauri Sood, Shipra, Dr. Rachna Soni, which does a good comparison between open source and proprietary software. Another good paper on the subject is "Open-source versus proprietary software: Is one more reliable and secure than the other?" [1] by Boulanger, Alan introduces the reliability and security of software that has been developed both as an open-source and proprietary model.

2 Hypothesis

Open-source software development models lead to faster innovation and feature development compared to proprietary software development models, but proprietary models have higher consistency in software quality and security.

To operationalize this hypothesis it can be divided into three different parts:

1. **Innovation and Feature Development Rate**: Measure the number of new features and updates by collecting information such as releases, type of releases, new features, and updates.

2. **Software Quality:** Can be assessed through metrics like the number of bugs. For proprietary software, this parameter can be user-reported issues, etc.

3. **Software Security:** Evaluate based on the number of security vulnerabilities reported and the severity of them.

Suggested Experiment Layout

1. **Sample Selection:** A sample of open-source software projects that are similar to a set of proprietary software projects. Meaning they match in size in terms of size and usage.
2. **Data Collection Period:** Of the selecting projects data should be collected during an extensive period to get the best data possible. The range will probably be specified from the reporting of proprietary software projects.
3. **Data Collection Methods:** Innovation and feature development can be tracked by scanning release notes and tracking code changes. Software quality can be tracked by bug reports, user feedback, and static code analysis. Software security can be tracked by vulnerability databases and security reports.
4. **Statistical Analysis:** Use statistical analysis to compare the different data using tests like t-test to see and understand the actual data collected.
5. **Outcome:** The outcome can be determined from the statistical comparison to see any major differences.

3 Requirements

The first requirement is the access to data. This part will probably be much easier for the open source part but can be quite difficult for the proprietary projects. This means that for open source projects data that will be collected will be release notes, update logs, bug reports, security vulnerabilities reports, and code scanning. For proprietary projects accessing data will be harder therefore most data will be accessed through reports generated by the company like bug reports, vulnerabilities, etc. Gaining access through partnerships might also be a solution. The data collected will have to uphold legal and ethical guidelines especially when dealing with proprietary software data.

Analyzing, collecting, and interpreting data could take a lot of time. Therefore it's important to understand the time and effort required to do so especially if you are dealing with proprietary projects by companies.

4 Should Computer Scientists Experiment More?

Some of the objections that Walter Tichy's article on "Should Computer Scientists Experiment More?" [3] can be applied to the research question "Comparative Study of Open Source and Proprietary Software Development Models". One

good example is the control of variables which in the paper states “There are too many variables to control and the results would be meaningless because the effects I’m looking for are swamped by noise.”. Especially in the research question, there are a lot of uncertainties, especially regarding the contribution size and quality. For example, how many developers are working on the different projects or how good are those developers, etc. These are variables that are hard to control and understand. Another big part is the cost and effort, the paper talks about the reason why researchers don’t experiment a lot is due to the cost and effort it requires. Especially in this project collecting data from proprietary projects will be extremely hard and might be costly. But Walter Tichy highlights “Instead of being paralyzed by cost considerations, ... convinced that the research addresses a fundamental problem, an experienced experimentalist would then plan an appropriate research program, actively looking for affordable experimental techniques and suggesting intermediate steps with partial results along the way.” It’s important to not just look at the effort and cost but rather try to actively look for affordable and time-efficient experimental techniques.

References

[1] Boulanger, A. Open-source versus proprietary software: Is one more reliable and secure than the other? *IBM Systems Journal 44*, 2 (2005), 239–248.

[2] Sood, G., and Soni, R. Comparative study: Proprietary software vs. open source software. *Int. J. Innov. Res. Comput. Commun. Eng. 4*, 11 (2016), 19032–19038.

[3] Tichy, W. F. Should computer scientists experiment more? *Computer 31*, 5 (1998), 32–40.