Further Investigations on Developer Contributions, Higher Software Quality

Jincheng He
Department of Computer Science
University of Southern California
jinchenh@usc.edu

I. Software is Everywhere

Software has been everywhere in our daily life since they made their way out of the lab since the last quarter of the 20-th century although sometimes people may not recognize them. Simple examples can be applications in the daily digital devices, such as PC, smartphones and tablets.

Some embedded software are less recognizable. The government, bank system, traffic, which are in the social level, and washing machine, instant pot, which are in the individual level, are also controlled under the help of software.

II. Open-Source Software

As software has become common in life, the way of how software developers collaborate with each other also varies in the past decades. Some software stay close-source which allow them to keep the development details secret while others open the development repositories, which are visible to everyone and allow contributions from outside the original developing groups.

These days, open-source software has become a popular way to develop software. Even large corporations conduct open-source development constantly. For example, Google, Netflix and Apache all have their own open-source software projects, on which developers from all over the world collaborate remotely. [TODO: Add footnotes here.]

III. Software Quality and Metrics

One of the difficulties of developing open-source software is managing and control the quality. As developers are from different time zones, with different skills and using different first languages, it is harder to manage the development team than the projects developed by employees working in the same floor.

Thus, we need to assess the quality and manage it to provide high-quality software to people, just like what we do to test body health conditions with medical devices. We want to know whether we are healthy and keep our body in a good condition and that is also what we want on software. A common way to evaluate the software quality is the metrics defined by the software engineering community, used in various tools [TODO: Add some references here]. Similar to the body height, the blood pressure and other medical test results, we have various metrics quantifying different quality metrics of software. For example, the lines of code, security, vulnerability are typical ones among them.

IV. Contributions from Developers

By analyzing the metrics, we are able to keep track of how contributions of developers impact software quality. During the development, they contribute to the projects with different purposes. Taking constructing a building as an example, we need someone to design the architecture, someone to prepare materials and someone to add bricks. Similar things happen when developers contribute to a software. Some design the architecture, some write trivial code, and some test and correct errors.

V. What We Do

As the contributions from developers varies in purposes, in this research, we investigate how these contributions with different purpose vary in their impact on the software quality and how we can improve it based on our discovery and analysis results. In the end, the research will help achieve a better software quality and either enable vendors to provide higher-quality software or lower the development and maintenance cost, consequently decreasing the price users pay.

(the citation here is for passing the compilation) [1]

References

[1] R. Wieringa, *Design science methodology for information systems and software engineering.* Springer, 2014, 10.1007/978-3-662-43839-8.