Research Proposal(小論文)

試験区分: 情報科学区分 氏名:Wong Ho Yin(黄　皓然) 希望研究室:ソフトウェア工学

Research Question: How does leadership of lead developers/ project managers affect the quality of code review?

If I become a student of NAIST, I would like to research how the leadership of lead developers/ project managers affect the quality of code. In this essay, I will first briefly introduce my background, then introduce my research plan in detail and finally explain why I would like to study at NAIST.

In my undergraduate life, I majored in Japanese Studies, which is a combination of Japanese Language Study and Anthropology of Japan. After graduating from the university and experiencing the demanding job-hunting activities in Japan, I realized that foreign candidates who graduated with a Bachelor of Arts degree are not cherished by companies in Japan, so I started learning programming by myself to sharpen my competitive edge. Fortunately, I got a freelance job offer from a web and AI development company called Datumix in August 2021 and since then I have been working as a frontend developer remotely.

Based on my work experience and my personal experience in collaborative web development, I reckon that code review plays an essential role in the quality assurance of code. Code review is an indispensable inspection of code change by an independent third-party code reviewer in order to identify and fix bugs or improve the structure of code before integrating code into the main branch (Thongtanunam et al, 2015).

Many pieces of research have been conducted to investigate the impact of human factors such as review workload and familiarity between reviewers and patch authors on the quality of code review. For example, Ruangwan et al. (2019) discovered that the more the reviewers are invited, the higher the rate that the reviewers do not respond to the review invitation. Baysal et al. (2016) revealed that factors like priority, organization and review queue also have a great influence on patch acceptance and patch writer experience has a great impact on code review outcome. However, there is no research regarding the influence of the leadership of project managers/ lead developers on the quality of code review.

Hogan and Kaiser (2005) argued that leadership is the utmost consequential for the success of organizations by building effective teams and persuading people to pursue a common goal. Indeed, according to my experiences in software development, the leadership of lead developers/ project managers brings a great influence to ensure the usefulness of code review through setting different rules related to pull requests and code reviews and encouraging team members to follow rules, and to actively participate in code review. A useful code review(comment) can be defined as the one which triggers one or more code changes within its proximity (i.e., 1-10 lines) in the consecutive commits (i.e., patches) of a pull request (Rahman et al.,2017).

In this research, I will first figure out (1) what kinds of rules lead developers/ project managers usually set for opening a pull request and conducting a code review. I would like to investigate whether any stop words or keywords, deadline and rule of pull request or other rules are set. Then I will figure out (2) how they implement the rules to ensure that useful code reviews are conducted constantly.

In order to collect data for this research, I will conduct an empirical study by participating in several software development teams as a developer. First, I will check whether lead developers/ project managers set and write down the rules on README files or other crucial documents for development. Apart from checking the existence of rules, I will

also check the pull request page in GitHub regularly to investigate whether developers follow the rules strictly and whether the rules facilitate useful code reviews. Moreover, I will attend meetings of the teams and keep a record of all messages in Slack to gain a better understanding of how leader developers/ project managers guide other developers to follow rules and to improve the quality of code review.

This research will not only contribute to the academic field of software engineering by suggesting a brand-new human factor (i.e., leadership) affecting the quality of code review, it will also enable leader developers, project managers or even the Chief Technology Officer of software development companies to have a better understanding of the proper management method for striving for excellence in collaborative software development.

Below I would like to explain why I want to study at NAIST. The first reason is that I would like to conduct research related to software engineering under Professor Matsumoto's supervision in Software Engineering (SE) Laboratory. Professor Matsumoto and his assistants have been publishing a lot of well-known journals related to the exploration of npm package and code-review, showing that they are knowledgeable to provide me with useful advice for conducting the research. He and his assistants hold meetings once a week in SE laboratory so that students can exchange ideas with each other and receive practical opinions from professors regularly, which let me feel secure when conducting the research. The second reason is that the syllabus of NAIST is attractive to me. Unlike other universities, NAIST welcomes students from different academic backgrounds and offer fundamental information science courses. As a graduate with a Bachelor of Arts Degree, I would like to take the information science courses such as algorithm and software development and study information science comprehensively for striving for excellence in my research as well as my career.

As a foreigner and a graduate with a Bachelor of Arts Degree, I cannot say that I do not have worry about studying at NAIST. However, I would like to take this opportunity to challenge myself in learning information science and conducting research related to software development. I hope I can utilize what I learn in NAIST to contribute to the academic field of software engineering as well as to improve my career development as a software engineer in the future.

References

Baysal, O., Kononenko, O., Holmes, R., & Godfrey, M. W. (2016). Investigating technical and non-technical factors influencing modern code review. Empirical Software Engineering, 21(3), 932-959.

Hogan, R., & Kaiser, R. B. (2005). What we know about leadership. Review of general psychology, 9(2), 169-180.ISO 690 Rahman, M. M., Roy, C. K., & Kula, R. G. (2017, May).

Rahman, M. M., Roy, C. K., & Kula, R. G. (2017, May). Predicting usefulness of code review comments using textual features and developer experience. In 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR) (pp. 215-226). IEEE.ISO 690

Ruangwan, S., Thongtanunam, P., Ihara, A., & Matsumoto, K. (2019). The impact of human factors on the participation decision of reviewers in modern code review. Empirical Software Engineering, 24(2), 973-1016.

Thongtanunam, P., Tantithamthavorn, C., Kula, R. G., Yoshida, N., Iida, H., & Matsumoto, K. I. (2015, March). Who should review my code? a file location-based code-reviewer recommendation approach for modern code review. In 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER) (pp. 141-150). IEEE.