The Effect of Checklist in Code Review for Inexperienced Students: An Empirical Study*

Guoping Rong, Jingyi Li, Mingjuan Xie, Tao Zheng**
State Key Laboratory for Novel Software Technology, Nanjing, 210093, China
Software Institute, Nanjing University, Nanjing, 210093, China
{ronggp, mg1132013, mg1132006, zt}@software.nju.edu.cn

Abstract

Code review is believed to be an effective technique to remove defects in early development stage and improve software quality. Therefore, it is regarded as one of the basic skills of qualified software engineers. Consequently, most curricula for SE students incorporated knowledge about code review in different courses. However, how to teach students to conduct efficient code review remains challenging. Many reports claimed that using checklist during code review could increase review efficiency (percentage of defects removed in code review). Nevertheless, we found a quite different result through analyzing the data collected from a PSP course took by freshmen. Results indicate that checklist contributes more to helping beginners conduct code review than to improving review efficiency. This finding implies that educators need to properly recognize the role of checklist in code review for students and explore more approaches to help students master skills to conduct efficient code reviews.

1. Introduction

Personal code review means manually check code after coding. The goal is to find and fix defects as many as possible before compiling and running the code [1]. For sake of both economical purpose and quality purpose, code review is a practice adopted by many practitioners. According to Humphrey’s research, code review could find and fix defects much faster than testing, therefore, it could dramatically reduce the testing time [1]. Besides, research from Jun-Suk Oh et al. [2] indicated that defects which have not been detected in testing could be found in code review. Code review could be applied in many project contexts. Forrest Shull et al. [3] found that reviews could catch more than 50% of a product’s defects no matter what domain, maturity level of the organization, or lifecycle phase they were applied in. From many aspects, code review is regarded as one of the essential skills that qualified software engineers should master and should be taught to SE students.

However, how to train beginners (inexperienced students) to conduct useful and efficient code review still remains challenging. Typical issues include: (1) lacking of helpful review guidance; (2) chaotic review process leading to unreliable review results; (3) lacking of motivation to conduct review.

One typical way to conduct review is the ad-hoc code review, which is informal and has no guidelines on review rate and review method [4]. This requires that the reviewer must be experienced so that he can know the best rate and method. Another typical way to conduct code review applies checklist during the review process. Reviewer must exactly follow a checklist. The entire code should be read line by line according to each checklist entry. Many

* This work was partially supported by the Project of Software Engineering Specialty Construction (2007-2201~2203), the Project of Software Engineering Teaching Team (2010-118), and the Project of Software Engineer Education Innovation (2008-042) by the Ministry of Education of China.
** Corresponding author: Tao Zheng (Email: zt@software.nju.edu.cn)