AN APPLICATION OF REVIEW TECHNIQUE FOR TEACHING SOFTWARE QUALITY SUBJECT

Ramona Binti Ramli

College of Information Technology, Universiti Tenaga Nasional, Kajang, Selangor.

ABSTRACT

Review techniques play an important role in order to produce quality software. However, this technique was rarely being exposed to undergraduate students. Thus, this paper presents an overview of review technique application for software quality subject, taken by software engineering students. This included with the technique applied for their project assignment and there are two projects were assigned to be completed during the lesson. In the first project, students were exposed on how to conduct the review activity to evaluate a given Software Requirement Specification (SRS); and the second project focused on the development of proposal based on the guidelines given. After the proposal completion, self-review and peer review activity conducted to evaluate the quality. Findings from the perspective of experience using the technique and lesson learnt from it, also been presented in this paper.

Keywords: Self-review, peer review, review technique, teaching software quality

1. INTRODUCTION

Review technique have been widely adopted by the industries. It plays an important role in software validation and verification. Research has proven that it will improve the quality of software and reduce the time and cost of development (Humayun et al. 2010). In software quality subject, review is one of the main topics that should be taught to undergraduates. Normally, this the students were exposed to theoretical concepts without having the chance to practise it.

To the best of our knowledge, little research have been conducted on the integration of review technique by the academician. In order to align with the industry needs, this study is conducted to propose a standard guidelines on how to adopt review technique in teaching undergraduate students. The case study was conducted at College of Information Technology (COIT), UNITEN. Participants involved
are the student of Bachelor of Computer Science (Hons.) in Software Engineering who registered for the Software Quality subject.

Review technique was taught as one of the topics for the Software Quality subject. Adapted from Gailin, the objectives of review were discussed during the class. Two types of review were introduced to the student: Formal Design Review and Peer Review. Review methodologies such as design review, inspection and walkthrough were being discussed theoretically. The study focuses on peer review methodology.

Another aim of the study is to conduct a complete review activities from beginning until end, as part of their learning process. The activities are considered as part of the assessment evaluation. The evaluation is measured from the number of defects detected to indicate the activity’s efficiency.

This paper is organized as follows. Section 2 will provide a summary of literature reviews conducted in education context. Section 3 discussed the design of the review activities in detailed. Section 4 focus on results and the discussion. The paper ends with conclusions and the proposed future work in section 5.

2. LITERATURE REVIEW

There are five major references for this study as shown in Table 1. Recent study by Schilling (2012) highlights that review exercise help the student in their understanding and will improve their design project. The study used guidelines on introducing review topics to students as proposed by Colofello (1987).

Rong et al. (2012) proposed that checklist is useful and important in introducing review techniques for beginners. Checklist can guide them while reviewing the artefacts in order to detect defects. However, the content of checklist should align with the artefacts in order to ensure the efficiency of its usage.

Humayun et al (2010) conduct a study to differentiate the effectiveness of review activities with or without the author presence. Review without author detects more defect that with author. This study also investigates the effectiveness of Similarity Domain Review (SDR) and Cross Domain Review (CDR) in detecting defects.
Lu and Wang (2006) prove in their study that by conducting review activities, it can improve students’ technical and soft skills. Technical review should introduce at the beginner’s level. The setting of environment plays an important role where it should be casual and friendly, but focused and control.

Most of the literatures also recommended to use the Software Requirement Specifications (SRS) as the suitable artefacts to be used during the review activity (Collofello 1987; Humayun et al. 2010; Schilling 2012). The artefact was chosen due to familiarity of the student with the SRS. Most students were being exposed on developing SRS during their first year. Another reason is to allow defect detection as early as possible.

<table>
<thead>
<tr>
<th>Author</th>
<th>Subjects Applied</th>
<th>Level of student</th>
<th>Review Technique used</th>
<th>Review Artefacts</th>
<th>Findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Collofello 1987)</td>
<td>Software Engineering</td>
<td>Undergraduate</td>
<td>Inspection and walkthrough</td>
<td>Specification for programming assignment</td>
<td>Guidelines on how to introduce review topics to students. Suggest review exercise to be applied in class.</td>
</tr>
<tr>
<td>(Lu and Wang 2006)</td>
<td>Various programming courses</td>
<td>Undergraduate</td>
<td>Technical Review</td>
<td>Plan Design Source code</td>
<td>Technical reviews improve their technical and soft skills.</td>
</tr>
<tr>
<td>(Humayun et al. 2010)</td>
<td>Software Engineering</td>
<td>Undergraduate</td>
<td>Similar Domain Review (SDR) and Cross-Domain Review (CDR) without author</td>
<td>SRS</td>
<td>Presence of author makes difference in defect identification Reviewer from similar domain more effective that cross domain.</td>
</tr>
<tr>
<td>(Rong et al. 2012)</td>
<td>Personal Software Process</td>
<td>Undergraduate - freshman</td>
<td>Code review</td>
<td>Source code</td>
<td>Checklist is important and efficient for beginners to be used in code review.</td>
</tr>
<tr>
<td>(Schilling 2012)</td>
<td>Not mentioned</td>
<td>Undergraduate</td>
<td>Formal software Inspection</td>
<td>Design artifact</td>
<td>SRS is usually selected for software inspection. The review exercise helps the student in their understanding and will improve their design project.</td>
</tr>
</tbody>
</table>
3. RESEARCH METHODOLOGY

This section describes the assignment design and the activities involved in order to carry out this study. The study was conducted as part of their project assignment. Students were divided into three groups. It was carried out in two phases: Project 1 and Project 2 (Figure 1). Project 1 emphasizes on peer review and the usage of checklist. The objective is to develop the skills on peer review. In Project 2, students are required to perform self and also peer review. Number of defects detected will be used to evaluate the efficiency of these activities.

![Figure 1](image)

### 3.1 Assignment Design of Project 1

In Project 1, each group of students were given a SRS of a Toy Shop System and the SRS review checklist (Wiegers 2001). During the activity, they were required to complete the checklist to identify any defects in the SRS. The review need to be done firstly by individually. During the group discussion, each member has to present their finding. The focus was to identify whether the findings were really a defect and record it in their report. The summary of the activity are shown in Figure 2.
3.2 Assignment Design of Project 2

Project 2 was divided into two phases. Phase 1 focused on development of project artefact and self-review activity. Peer review activity in phase 2 was conducted after the completion of phase 1. During phase 1, students were required to develop a proposal for the requested system based on the scenario given. After completion, they evaluated their own proposal through self-review by using Proposal & Draft Reviews – Subjects Checklist (Galin 2004). Reworks were done based on the results from the self-review checklist. After reworks, the proposal was evaluated again by other group through peer review using the same checklist (Figure 3) in phase 2.
4. RESULTS AND DISCUSSION

4.1 Results of Project 1

Results of activities from Project 1 are shown in Table 2. Total number of defect for the SRS is 15. Defect is calculated as number of NO values obtained from the checklist. The result shown that two groups achieved review efficiency exceeds 50%. These two groups able to follow the steps defined in the activity. Review was done thoroughly to ensure the defects were highlighted. However, group 1 did not perform well due to participant's behaviour, commitment and dishonesty.

Table 2

<table>
<thead>
<tr>
<th>Group No</th>
<th>No of defects</th>
<th>Review Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>6</td>
<td>40%</td>
</tr>
<tr>
<td>Group 2</td>
<td>10</td>
<td>67%</td>
</tr>
<tr>
<td>Group 3</td>
<td>9</td>
<td>60%</td>
</tr>
</tbody>
</table>

4.2 Results of Project 2

Results from Project 2 as shown in Table 3. Actual number of defect is the number of defect found by the instructor. The proposal was evaluated twice: before and after reworks. The identified defects were then used to compare with the number obtained from the self and peer review activities.

The number of defects for group 1 decreased from 92% to 71%. From the result, it shown that this group has effectively used the checklists during the self-review. Reworks were done properly after the evaluation. Therefore, small number of defects was detected during the peer review.

However for the other two groups, it was found that the reworks were not done properly after the self-review activity. The percentage of defects has increased after the peer review activity. This is due to the commitments of the group member with assignments for other subjects during the submission week.
Table 3

<table>
<thead>
<tr>
<th>Group no</th>
<th>Self-review</th>
<th></th>
<th></th>
<th>Peer review</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual no of defects before reworks</td>
<td>No of defects found</td>
<td>Efficiency (%)</td>
<td>Actual no of defects before reworks</td>
<td>No of defects found</td>
<td>Efficiency (%)</td>
</tr>
<tr>
<td>Group 1</td>
<td>13</td>
<td>12</td>
<td>92</td>
<td>7</td>
<td>5</td>
<td>71</td>
</tr>
<tr>
<td>Group 2</td>
<td>17</td>
<td>10</td>
<td>59</td>
<td>18</td>
<td>12</td>
<td>67</td>
</tr>
<tr>
<td>Group 3</td>
<td>18</td>
<td>14</td>
<td>78</td>
<td>14</td>
<td>13</td>
<td>93</td>
</tr>
</tbody>
</table>

4.3 Student Feedbacks

An interview was conducted with the students at the end of semester. Student feels that the project assignment had improved their knowledge on the importance of improving software quality through effective reviews. The students also feel motivated to develop better artefacts for their future project. After the activity, they were more willingly to share and criticize each other works. This activity were also recommended for future students that will be taking this subject.

5. CONCLUSIONS AND FUTURE WORKS

In this study, the complete process of review was introduced to the inexperienced students. Students were able to follow the suggested activities in order to complete the assignment. However, total success of these activities depends on few factors such as participant’s behavior, commitment and honesty.

For future works, time constraint will be included as one of the factors. Other review technique such as walkthrough will also applied to same profile of students. Results from both techniques will be compared and analysed. In term of project artefacts, other types such as source code and user manual will be adopted.

REFERENCES


