ABSTRACT

Software defective testing technology is important means to find and remove software defects. Software defective detection is a part of software inspection process. Around the software defective detection, this paper deeply analyzes the strengths and weaknesses of current popular software defective testing technology and the key problems. This study was to discover whether software inspectors with different background can perform defect detection differently using Traceability-based reading technique. The result showed that software inspectors with different undergraduate degrees, undergraduate object oriented courses taken, and object-oriented work experience do not perform defect detection significantly differently.

INTRODUCTION

A majority of software inspection research has been on the technical view of software inspection. Software reading technique for defect detection was one of the main issues. These software reading techniques can be used on software artifact including software requirement, software design, and code.

As defect detection in early stage in software development lifecycle is less cost to repair in later stages[6], software reading techniques can be applied to object oriented design and requirement for defect detection. Most of scenario-based reading techniques for object-oriented design require some training on how to use
the reading for the inspectors. Some experiments include a whole day subject training in the experimental protocol.

Object-oriented design specifications usually contain UML diagrams and descriptions, as well as descriptive requirement. Thus, object oriented software inspectors should understand the UML diagrams well enough. Traceability based reading technique [7] allows inspectors to check whether the design corresponds to the requirements using vertical and horizontal reading. This reading technique provides software inspectors with steps on where and how to find defects in object oriented design documents.

Most students with software related undergraduate degrees have taken at least one object-oriented course during their undergraduate study. These students should be ready to become software inspectors after graduation. With the software reading containing clear instructions, there should be a small amount of training on defect detection. With a readable reading technique such as Traceability-based reading technique, prior training and experience may not be necessary for a software inspector in defect detection on object-oriented design documents.

TRACEABILITY BASED READING TECHNIQUE

This technique was specifically developed for object-oriented design document inspection. The main purpose of this technique is to ensure consistency and completeness of the work product. This reading technique covers horizontal readings, for checking the different design artifacts against each other, and vertical readings, checking whether the design corresponds to the requirements.

There are six readings in a set of Traceability-based reading technique. Each reading asked inspectors to read specific design documents together, the steps to read them, and where and how to find defects in each step. The six readings [8] are:

- Class Diagrams with respect to Class Descriptions
- Class Diagrams with respect to State Machine Diagrams
- Interaction (Sequence) Diagrams with respect to Class
- Class Descriptions with respect to Requirements Description
- Interaction (Sequence) Diagrams with respect to Use cases
- State Diagrams with respect to Requirements Description and Use cases Diagrams.

It was claimed that Traceability-based reading technique can help user without experience in software inspection through defect detection process in software reading.

EXPERIMENTAL PLANNING

The design document in this study was a car rental application consisting of requirement specification, use case diagram and description, sequence diagram, and
As defect detection in software designs had identified a defect classification scheme for object-oriented design in 5 classifications (Omission, Incorrect Fact, Inconsistency, Ambiguity, Extraneous Information)[9].

To determine appropriate seed defects in the design document for this study, twelve object-oriented student group project design documents were inspected to analyze the highest frequently defect subtype occurrence in each defect classification. Incorrect classification found least from the twelve object-oriented student group project design documents inspected. A variety of seed defects were placed into the design document in this study according to the subtype of each defect classification.

However, there were defects that can be detected due to the main defect. For example, when a class was missing from a class diagram, attributes, methods, and associations of the missing class were also missing from the design and thus detected as Omission and Inconsistency type of defect classification, respectively. In this study, a missing class defect leads to additional 4 Omission defects and 2 Inconsistency defects.

The subjects were given a questionnaire asking for their profiles on work experience, undergraduate degree, and courses taken. These questionnaires were filled out before any of the inspection tasks began. The subjects were given no time limitation to inspect the design documents. Once they detected a defect, they filled out the defect detection report indicating where (which design document) that defects were detected and description of the defects. The task were given out the subjects in the first class of object oriented analysis and design course offering in the first semester of the program. In this first class, only object-oriented concept was given to the subjects.

**EXPERIMENTAL RESULT**

The summary report of defect detection behavior from this study was shown in Table I.

When looking at the number of defect detected by individual subjects by their degree earned (software-related and non-software related), the result is shown in Table II. It can be concluded that the undergraduate degree does not differ the subject performances in the number of defect detected in object-oriented design document inspection.

The number of defect detected for each individual subject was counted regardless of defect classification. The average number of defect detected by 29 subjects was 7.0 with the highest number of defect detected of 14 defects and the lowest number of defect detected of one defect.

The result in Table II also indicated that the defect detection performance of the subjects with software-related undergraduate degree might be different from the
performance of the subjects with other undergraduate degrees at the level of significance of 90%.

TABLE I. DEFECT DETECTION BEHAVIOR OF SUBJECTS IN THE STUDY.

<table>
<thead>
<tr>
<th>Defect classification</th>
<th>No of defects Not detected by any subjects</th>
<th>Average percentage of subjects that were able to detect defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Omission</td>
<td>2 out of 9</td>
<td>27.72</td>
</tr>
<tr>
<td>2. Incorrect Fact</td>
<td>0 out of 1</td>
<td>6.90</td>
</tr>
<tr>
<td>3. Inconsistency</td>
<td>2 out of 7</td>
<td>40.89</td>
</tr>
<tr>
<td>4. Ambiguity</td>
<td>0 out of 5</td>
<td>24.83</td>
</tr>
<tr>
<td>5. Extraneous Information</td>
<td>1 out of 1</td>
<td>0.00</td>
</tr>
</tbody>
</table>

TABLE II. TEST ANALYSIS COMPARING THE NUMBER OF DEFECTS DETECTED BY THE SUBJECTS’ UNDERGRADUATE DEGREE.

<table>
<thead>
<tr>
<th>Subject Category</th>
<th>N</th>
<th>Mean</th>
<th>Std.Dev</th>
<th>Sig (2-tail)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software-related undergraduate degree holders</td>
<td>20</td>
<td>7.7</td>
<td>2.96</td>
<td>0.08</td>
</tr>
<tr>
<td>Non-software-related undergraduate degree holders</td>
<td>9</td>
<td>5.44</td>
<td>3.36</td>
<td></td>
</tr>
</tbody>
</table>

CONCLUSIONS

It can be concluded that the software inspectors with software-related undergraduate degrees, undergraduate object oriented courses taken, and object-oriented work experience do not perform defect detection significantly differently. However, the software inspectors with higher undergraduate performance perform significantly better in defect detection on object oriented design documents.

Traceability-based reading technique can be used for the newly recruited software inspectors with or without prior object-oriented profiles during training. However, the average defect detection performance of these subjects in this study was not high, especially on extraneous information and incorrect fact classification of defects.

REFERENCES