Experimental Evaluation of Effect of Specifying a Focused Defect Type in Software Inspection

Shuji Morisaki Yasutaka Kamei Ken-ichi Matsumoto

This paper investigates impact of focus on specific defect type in software inspection. We conducted a controlled experiment with three groups in respect of giving a prioritized defect type. In the experiment, 32 practitioners are grouped into (a) inspection without a focused defect type, (b) inspection with a focused defect type determined at the beginning of inspection, (c) inspection with a focused defect type determined at the beginning of inspection and verified whether the detected defect is categorized as the focused defect type at every defect detection. The result of the experiment shows that specifying the focused defect type increases the proportion of defects categorized as focused type to other defects. The number of defects categorized as focused type in both of groups (b) and (c) is three times larger than that of group (a). In group (c), 14 defects are detected and 13 defects are categorized as focused type. The number of trivial defects (e.g. cosmetic and typographical ones) in group (a) is three times larger than that of groups (b) and (c). The subjects estimated the cost of the saved correction effort by early detection of each defect. The numbers of defects that decreases the correction effort is the largest for group (b). The difference between the results of groups (a) and (b) indicates that in ordinal inspections, the procedure employed by group (b) is expected to provide larger efficiency.

1 Introduction

Many techniques have been proposed for increasing development efficiency and improving software quality. Software inspection is such a technique that is aimed at early defect detection. In the article [1], software inspection techniques are classified into those for supporting existing sub-processes without change and those for creating new sub-processes or redefining existing sub-processes.

Reading support [2] [11] is one of the major techniques for supporting existing sub-processes without change. For example, perspective-based reading [2] involves assigning roles to an inspector such as user or tester at the preparation sub-process in inspection process. Each role has a scenario described as a procedure to detect defects. At the defect detection sub-process, inspectors try to detect defects according to the assigned role and scenario.

On the other hand, phased inspection [8] and N-fold inspection [9] are major techniques that change the inspection sub-processes. These techniques repeat the defect detection meetings for the same artifact changing focus on defect type such as maintainability, reliability, and usability. Setting different focuses among defect detection meetings is expected to increase efficiency and effectiveness.

In the article [10], setting the defect type to be focused during inspection is recommended. Focused defect types are selected after analyzing detected defects in past projects. However, as far as the authors are aware, there is no published study that investigates the impact of determining and sharing a focused defect type as a parameter before the inspection.

In this paper, we define the focused defect type as a defect category that has higher priority than other defect categories in detection. Focused defect type is determined before defect detection. Inspector prioritizes the focused defect type in identify-

ソフトウェアレビューにおける着眼点の設定が指摘結果に与える影響の分析

森崎 修司, 亀井 靖高, 松本 健一, 奈良先端科学技術大学院大学 情報科学研究科, Graduate School of Information Science, Nara Institute of Science and Technology, Japan.

コンピュータソフトウェア, Vol.0, No.0 (0), pp.0–0.
[研究論文 (レター)] 1999 年 8 月 3 日受付.

ing and detecting defects during preparation and inspection (defect detection) phases, defined in the article [7].

We experimentally investigate determining and sharing a focused defect type before defect detection. In the experiment, the target document is a software design document of conference registration system. Subjects are 32 practitioners. The subjects are divided into three groups. In group (a), no focused defect type is provided. In group (b), the focused defect type is determined at the beginning of the inspection. In group (c), the focused defect type is determined at the beginning and each detected defect is verified to confirm whether it is classified as the specified focused defect type at each detection.

2 Design of the experiment

Table 1 shows an overview of the experiment. The subjects performed design inspection of a web-based conference registration system. The design document was written in natural language. The size of the document is approximately 8000 Japanese characters (approximately 4000 words in English), 6 pages in the A4 format.

Figure 1 shows an overview of the system. The purpose of the system is to accept conference registration requests from individuals and provide the conference organizer with a list of registered individuals. The system can manage conferences simultaneously. The system administrator initializes a conference and obtains a unique URL for the conference. The conference organizer passes the URL to potential conference attendees (call for participation). An attendee registers herself/himself to the conference by using the system. Conference organizers can obtain the list of registered attendees.

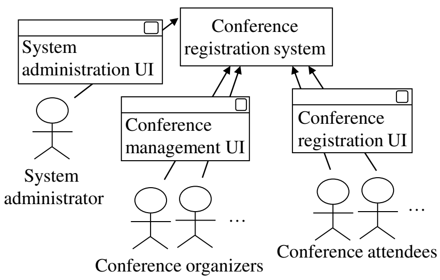

Fig. 1 Target system

Subjects are grouped into three groups:

group (a): without focused defect type

group (b): with focused defect type determined by inspectors and shared among inspectors as the prioritized defect type at the beginning of the inspection

group (c): with focused defect type determined by inspectors, shared among inspectors as the prioritized defect type at the beginning of the inspection, and verified whether it is categorized as the focused type by the inspectors at every detection.

Each group consists of two teams, each of which consists of five or six subjects. The inspectors are asked to detect defects in the design document of the online conference registration system in a face-to-face inspection meeting. All the inspectors are practitioners engaging in commercial software development.

In all teams, one moderator and one recorder roles are assigned. Teams in group (a) are asked to perform inspection as they usually do. In groups (b) and (c), the inspectors are asked to select a focused defect type.

Teams of group (b) and (c) are asked to select one focused defect type from the sub-characteristics of ISO/IEC 9126 Software Quality before beginning defect detection. Although there are major defect type schemes such as ODC [5] and Beizer's defect type [3], we choose ISO/IEC 9126 as a defect type scheme because existing defect type schemes do not cover maintainability and portability considerably.

Tables 2, 3, and 4 are the defect report forms for groups (a), (b), and (c), respectively. The column "Focused defect type" on the forms for group (b) and (c) shows the determined type. The teams of groups (b) and (c) are asked to detect defects with the focused type in mind. The teams of group (c) are also asked to record the verification result at every detection, this result is shown in the column "Classified as focused defect?" in Table 4. The teams of group (b) are asked to determine whether each detected defect is classified as the focused defect after the experiment (defect detection).

The explanation of the focused defect type to the teams of group (b) and (c) took five minutes. The inspection (defect detection) is one hour, excluding the explanation to groups (b) and (c). Teams

Table 1 Overview of the experiment

System	Web-based conference registration system	
Document	Design document written in natural language, 4000 words (6 pages in A4 format)	
Duration	1 hour	
Number of subjects	32	
Group	a (a1, a2)	No focused defect type
	b (b1, b2)	Focused defect type is determined and shared at the beginning of the inspection.
	c (c1, c2)	Focused defect type is determined and shared at the beginning of the inspection; On each detection, the detected defect is categorized whether it is classified as the focused defect.

in group (b) are asked to determine whether each detected defect is classified as the selected focused defect type after the experiment.

3 Result of experiment

All teams in groups (b) and (c) determined security issue as the focused defect type. They took about five minutes to determine and share the type.

Table 5 shows the result of the experiment. The values in the table are the numbers of detected defects. Although the focused defect type is not determined in teams a-1 and a-2, some detected defects can be categorized as security issues. The values of focused defect type for teams a-1 and a-2 indicate the number of defects classified as security issues; these values are listed in Table 5. "Trivial" refers to the number of typos, grammar, incomplete formatting, or cosmetic defects.

The total number of detected defects is the largest for group (a), followed by that for groups (b) and (c). The number of defects classified as security issues by teams a-1 and a-2 are 1 and 2, respectively. In group (a), the percentage of trivial defects is about 50 percent of the entire defects detected and is larger than the percentages of those in groups (b) and (c).

The number of focused defects in group (b) is larger than that in group (a). In both teams b-1 and b-2, the number of focused defects is half the number of other defects (the sum of "non-focused & non-trivial" and "trivial"). The number of trivial defects in group (b) is smaller than that in group (a).

In group (c), the total number of detected defects is 7 in both teams, c-1 and c-2. Almost all detected defects are categorized as focused defect type. The number of focused defects in group (c) is similar to those in group (b).

Table 5 Number of detected defects

	a-1	a-2	b-1	b-2	c-1	c-2
Focused	(1)	(2)	5	9	6	7
Non-focused & non-trivial	(22)	(10)	5	13	0	0
Trivial	24	11	4	5	1	0
Total	47	23	14	27	7	7

After the experiment was completed, each team evaluated the contribution of the defects detected in inspection to the decrease in the correction effort (by early detection). In the evaluation, each team was asked to estimate two types of correction efforts: (i) correction effort for fixing the detected defect in the inspection and (ii) correction effort for fixing the defect in testing, assuming the defect was overlooked in the inspection. If the cost for (i) is smaller than the cost for (ii), the detection apparently decreases the correction effort.

Table 6 shows the result of the evaluation. In group (c), the total number of detected defects is 14 and the number of focused defects that decrease the correction effort is 12. In group (b), the number of defects that decrease the correction effort, not restricted to the focused defect, is the largest among three groups. In group (a), the number of defects that decrease the correction effort is 12 while the total number of detected is 70.

Table 2 Defect report form for group (a)

ID	Time	Page, line	Description of defect
1	*14:10*	*p. 4, l.1*	*Spelling mistake in the title of section 1*
2			
...			

Table 3 Defect report form for group (b)

Focused defect type: *Exception handling*

ID	Time	Page, line	Description of defect
1	*14:12*	*p. 13, l.55*	*No exception handling when the data exceeds maximum file length*
2			
...			

Table 4 Defect report form for group (c)

Focused defect type: *Exception handling*

ID	Time	Page, line	Description of defect	Focused type?
1	*14:09*	*p. 62, l.1*	*Spelling mistake in the title of section 3*	*no*
2	*14:12*	*p. 13, l.55*	*No exception handling for network disconnection*	*yes*
...				

Table 6 Number of defects decreasing correction effort

	a-1	a-2	b-1	b-2	c-1	c-2
Focused	(1)	(2)	5	4	6	6
Non-focused & non-trivial	(6)	(3)	4	4	0	0
Total	7	5	9	8	6	6

4 Discussion

4.1 Impact of focused defect type

The result of the experiment demonstrates that specifying a focused defect type affects the number and category of defects detected. The number of the detected defects that are categorized as focused defect type in groups (b) and (c) indicates that specifying the focused defects prior to the defect detection increases the number of focused defects and decreases the number of trivial defects. The result of group (c) particularly shows only one defect that is not classified as the focused type, while the other 13 defects are classified as focused type. Although in both teams of b-1 and b-2, the number of focused defects is smaller than the number of other defects, the number of focused defects in group (b) is larger than that in group (a).

The total number of detected defects in group (a) is the largest. However, half the detected defects are trivial defects in both teams of a-1 and a-2. We conducted further investigation by interviewing a subject of group (a). The subject said, "Without focused defect type, some inspector's trivial defects triggered similar trivial defects by other inspectors." The focused defect type may prevent such a situation.

In the article [6], the assignment of the reading technique causes a significant difference in the number of trivial defects detected. The percentage of trivial defects detected by the group assigned reading technique was 12.8% and the percentage of trivial defects detected by the group assigned no reading technique (ad-hoc reading) was 25.2%. This tendency is similar to our investigation in terms of the percentage of trivial defects.

The number of defects that decrease the correction effort is largest in group (b). The number of defects that decrease the correction effort is the same for groups (a) and (c). The percentage of focused defects that decrease the correction effort is the largest for group (c). In real situation, if a focused defect type is identifiable, the procedure employed by group (c) is expected to increase the efficiency of inspection. Comparing groups (a) and (b), specifying the focused defect type may increase the number of defects that decreases the correction effort.

4. 2 Estimation of inspection result

The result of the experiment indicates that specifying a focused defect type enables project manager to easily estimate and predict the category of defects detected during inspection. Identifying the risk of the software and arriving at a consensus for a focused defect type by project manager, software purchaser, and software developer before inspection lead to the decrease in the number of critical defects that are overlooked during inspection. Arriving at a consensus for a focused defect type also leads to optimization of the inspection effort.

In phased inspection [8] and N-fold Inspection [9], a series of partial inspections are conducted. Each inspection is called a phase. Each phase has a specific goal. In such inspections, assigning the focused defect type to the goal of each phase and verifying whether the defect is classified as the type at every detection are expected to increase effectiveness and efficiency in various ways such as reducing the number of duplicated defect detection.

4. 3 Threats to validity

In the experiment, each group has only two teams. Further investigations are required for generalizing the result of the experiment. However, all teams in groups (b) and (c) show the same trends that support specifying a focused defect type increases the proportion of detected defects to the focused defect type.

All subjects in the experiment are practitioners who are working on various software products including enterprise application, embedded software, and package/web-based services. No team is biased with respect to security knowledge. All teams include one or more inspectors who have knowledge of security.

In the experiment, the determined focused defect type is the only security issue. Various other focused defect types are required to generalize the result. We interviewed several subjects of the experiment and obtained feedback that other focused defect types such as performance issue, resource issue, and exception handling also provide similar benefit because these types are sufficient clear for detecting defects. We also asked some subjects why they selected security issues even though there was no experimental design for selecting the same defect type. The subjects replied that they thought that security issues were the most important for a web application. Experiments with specifying other focused defect types and investigations on relationship detected defects are some of the most important future works.

Both teams of group (b) took several minutes to determine whether each detected defect is classified as the focused defect. The time of type is not included in the experiment. However, the time for type is less than three minutes. The authors believe that this duration is short and can be considered negligible.

The team a-1 detected larger number of trivial defects than other teams. There is a possibility that poor quality of the target document increases the number of trivial defects.

5 Conclusion

We conducted an experiment to investigate the impact of focused defect type on detected defects in software inspection. In the experiment, 32 practitioners were divided into three groups: (a) those without focused defect type, (b) those which determine and share a focused defect type at the beginning of the inspection and (c) those which determine and share a focused defect type at the beginning of the inspection and further verify whether the detected defect is classified as the focused defect type at every detection.

The experiment demonstrates that determining and sharing a focused defect type at the beginning of the inspection affects the number and category of detected defects. In comparison to group (a), groups (b) and (c) detected three times the number of defects categorized as the focused defect type. In groups (b) and (c), the number of trivial defects is decreased to less than 26%.

After the inspection, subjects estimated two types of correction efforts. One is effort for fixing a defect detected in inspection. The other is correction effort in testing if the defect is overlooked in inspection then detected during testing. The number of defects that potentially decrease the correction effort is the largest for group (b). The number of defects that potentially decrease the correction effort is the same for groups (a) and (c).

The effort of determining focused defect type is small. When a focused defect type can be specified, there is no required effort for learning reading. Determining a focused defect type can increase inspection efficiency.

In particular, for inspections with a set of specific goals and phases, such as phased inspection and N-fold inspection, the procedure employed by group (c) is expected to eliminate the duplicated defects among the phases and to increase efficiency. In usual inspections without phases, the procedure employed by group (b) is expected to detect the defects that potentially decrease the correction effort.

Acknowlegement

This work is being conducted as a part of the StagE Project, the Development of Next Generation IT Infrastructure, supported by MEXT (Ministry of Education, Culture, Sports, Science and Technology) Japan. And also supported by Grant-in-Aid for Young Scientist (B) 21700033.

References

[1] A. Aurum, H. Petersson, C. Wohlin: State-of-the-art: software inspections after 25 years, Journal of Software Testing, Verification and Reliability, vol.12, no.3, pp.133-154(2002)

[2] V. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sorumgard, M. Zelkowitz: The empirical investigation of perspective-based reading, Journal of Empirical Software Engineering, vol.2 no.1, pp.133-164(2006)

[3] B. Beizer: Software Testing Techniques, Van Nostrand Reinhold(1990)

[4] J. K. Chaar, M. J. Halliday, I. S. Bhandari, R. Chillarege: In-process evaluation for software inspection and test, IEEE Trans. Softw. Eng., vol.19, no.11, pp.1055-1070(1993)

[5] R. Chillarege, I. S. Bhandari, J. K. Chaar, M. J. Halliday, D. S. Moebus, B. K. Ray, M. Y. Wong: Orthogonal defect classification-a concept for in-process measurements, IEEE Trans. Softw. Eng., vol.18, no.11, pp.943-956(1992)

[6] B. Cheng, R. Jeffery: Comparing inspection strategies for software requirement specifications, Proc. of Australian Software Engineering Conf., pp.203-211 (1996)

[7] M. E. Fagan: Design and code inspections to reduce errors in program development, IBM Systems Journal, vol.15, no.3, pp.182-211(1976)

[8] J. Knight, A. Myers: Phased inspections and their implementation, SIGSOFT Software Engineering Notes vol.16, no.3, pp.29-35(1991)

[9] J. Martin, W. Tsai: N-fold inspection: a requirements analysis technique, Comm. of the ACM, vol.33, no.2, pp.223-232(1990)

[10] M. V. Mantyla, C. Lassenius: What types of defects are really discovered in code reviews?, IEEE Trans. Softw. Eng., vol.99, no.2, pp.430-448(2009)

[11] T. Thelin, P. Runeson, B. Regnell: Usage-based reading - an experiment to guide reviewers with use cases, Inform. and Software Tech., vol.43, no.15, pp.925-938(2001)