PBR vs. Checklist: A Replication in the N-Fold Inspection Context

Lulu He and Jeffrey Carver
Mississippi State University
300 Butler Hall, Box 9637
Mississippi State, MS 39762
+1 662-325-8798
{lh221, carver}@cse.msstate.edu

ABSTRACT

Inspection is considered a powerful method to check software documents for defects. Many published work shows that inspections in requirements specification phase are particularly effective and efficient. Perspective-Based Reading (PBR) is one of the systematic techniques to support defect detection in requirements documents. In this paper we describe an experiment to validate the effectiveness of PBR in a meeting-based N-fold inspection. Our goals were: (1) re-test the hypothesis of the original experiment that PBR helps to increase individual and team defect detection effectiveness compared to an checklist approach; (2) investigate the different impact of PBR and checklist on the effectiveness of N-fold team meeting; and (3) investigate some interesting characteristics of PBR (e.g. the relationship between background experiences and performance of the subjects). The results of the study showed that PBR was significantly more effective than checklist (supporting the original study). We also found that the team meeting is much more important for checklist teams, based on the number of meeting gains and the number of false positives eliminated. Finally, we found that teams using the PBR techniques have less overlap in their defect detection than those using checklist. The ultimate goal is to provide best practices (guidance) for applying PBR in software inspection and also some advice for PBR (or software inspections) process improvement.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – *Productivity.*

General Terms

Management, Measurement, Documentation, Performance, Experimentation, Human Factors, Verification

Keywords

PBR, N-fold inspection, experimentation, human subjects

1. INTRODUCTION

Defects in the software system requirements, design or code, are usually considered an unavoidable aspect of software development. The later the defects occur in software development life cycle, the more difficult to detect and correct them. Thus, discovering and removing defects early is crucial to the success of software development projects. Most published work shows that an inspection by qualified reviewers is an effective and efficient way to remove defects [12]. In particular, inspections in the software requirement specification (SRS) phase [2, 18] can detect most of the inconsistent or incorrect defects in requirements and therefore greatly contribute to the overall system quality.

The inspection process typically consists of several phases: planning, overview, defect detection, defect collection, and follow-up [2]. The defect detection, usually done by individual reviewers and the defect collection, often done during a meeting, are the two central steps to this process. The emphasis on these two phases varies with the inspection approach. On one end of the spectrum is the *walkthrough*, in which the emphasis is on the defect collection team meeting [6]. In a more formal inspection process, such as that proposed by Fagan [5], the individual team members have specific preparation responsibilities prior to the tem defect collection meeting. So in the formal inspection process, much of the defect detection is done prior to the meeting.

Much of the previous research in the software inspections area has focused on understanding these two phases in isolation. For example, there has been previous research focused on how to structure the team meeting including the *N-fold Inspection* process [14] and the *Phased Inspection* process [11]. Other researchers have called into question the need for a team meeting [17, 30].

There has also been considerable research done in the individual defect detection phase. During this phase, reviewers scrutinize a software artifact and apply some technique to elicit defects. According to Porter, the reading techniques play the key role in improving the effectiveness of software inspection rather than the changes in the structure of inspection process [19]. So far, several inspection techniques have been proposed in the literature. They range from intuitive, nonsystematic procedures, such as Ad Hoc or Checklist techniques [4, 5], to explicit and systematic procedures, such as Defect-based Reading [18], Perspective-based Reading (PBR) [2], Object-Oriented Design Reading [24], and Usability-based Reading [32]. It is necessary to gather the knowledge about the practical benefits of these techniques to better support the inspection process.

Much of the previous work on PBR has focused on comparing its performance to that of a checklist during the individual preparation phase without taking the team meeting into account. Similarly, the two competing lines of research (meeting structure vs. inspection techniques) have rarely examined the interactions between these two variables. Therefore, it is not clear from the previous research if the benefits that PBR provides over a checklist will still be true in the context of a specific inspection methodology such as N-Fold inspections. In this paper, we address this question.

Section 2 describes the previous related work, followed by Section 3, which discusses the research hypotheses and experimental design. In Section 4 we discuss the results, and provide an interpretation in Section 5. Section 6 provides conclusions and future work.

2. RELATED WORK

To investigate the interaction between PBR and the N-fold inspection method, we must cover related work from both of these areas.

2.1 PBR

PBR is a systematic technique that supports the defect detection in a software artifact (e.g. SRS). The basic idea is that PBR reviewers stand in for specific stakeholders of the SRS to verify its correctness [2]. In this way PBR offers the beneficial attributes such as more effective, systematic, focused, goal-oriented and customizable, transferable via training [26]. Naturally, we need empirical studies that evaluate the performance of PBR and other reading techniques to verify these claims.

The original experiment to test the effectiveness of PBR was conducted with professional software developers from the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) Software Engineering Laboratory (SEL). The main results of the original experiment support the hypothesis that individual using PBR performed better than using the checklist approach, especially when they were less familiar with the domain [2]. Later on, a series of replicated experiments were conducted by different researchers [2, 4, 10, 13, 18, 23]. Their results basically support the findings of the original experiment that individual and teams perform better using PBR than a checklist approach for defect detection. They also analyzed other characteristics of PBR, such as the relationship between PBR and detection of particular defect classes, and the performance of real teams vs. simulated teams. The results of such studies tend to indicate that there is less overlap in the defects that are found by PBR inspections as compared with checklist inspections.

The number of subjects in these experiments is often low, so it is necessary to conduct replications of the original experiment in similar environments to increase the confidence in the results [4]. Furthermore, because the experiment settings of each replication may be slightly different, performing the replication helps to gain an insight of the properties of PBR that have never been investigated before [25].

2.2 N-fold Inspections

The N-fold inspection method [21, 22, 29] is a technique where multiple inspections on the same artifact are carried out in parallel, by some number (N) of teams. Improved inspection performance can be expected based on the hypothesis that a single inspection team can detect only a subset of the total number of defects and that multiple inspection teams can detect (more) unduplicated defects [14]. Team meetings are one way to collate the inspection results from different teams. Studies show that team meetings, particularly the collection meetings, have the benefit of finding new faults, undetected by individuals working separately [5, 29]. Often these meetings result in a *meeting gain* of approximately 5% (i.e. the number of new faults detected during the detection meeting divided by the total number of defects) [8, 12, 13, 14, 21, 22, 24]. Conversely, team meetings have proven costly in terms of coordination overhead - additional time resulted from the scheduling conflicts [1, 22], and lengthened development time [8, 30]. The empirical results from some controlled experiments indicate that meeting-based inspection is not necessarily more effective than meetingless inspection [17, 30].

Based on this previous work, it was not clear what impact using the N-fold inspection process would have on the relationship between PBR and a checklist. To investigate this question, we conducted an experiment with students of Computer Science & Engineering Department at the Mississippi State University to evaluate the effectiveness of PBR and checklist approach in the context of a meeting-based N-fold inspection. In this study, we replicated the original experiment, with two changes. First we used a real SRS document, and second, we used the N-fold inspection process. We first wanted to validate the results of the previous experiments and then investigate the impact of the new N-fold inspection context

3. THE EXPERIMENT

Our experiment was conducted as part of a graduate requirement-engineering course at Mississippi State University in the Fall of 2005. The experiment was run as a course project helping students to gain hands-on experiences of the inspection of SRS. The project was not graded, but the participation and contribution of the students influenced their final grades, so we can be confident that the students participate seriously in the experiment. Twelve subjects participated in this study

3.1 Research Questions and Hypotheses

We use the Goal Question Metric (GQM) approach to define the goals for this study. Staring with some goals and hypotheses from the original study [4], and adding some specific to our replication, we obtained the following goals and hypotheses::

Goal 1: *Analyze* PBR and checklist reading techniques *for the purpose of* their evaluation *with respect to* their effectiveness for individuals

> **Hypothesis 1**: individuals applying PBR perform better than individuals using reading techniques with respect to their mean defect detection rate.
>
> **Hypothesis 2:** The experience of the subjects has no influence on their mean defect detection rate.

Goal 2: *Analyze* PBR and checklist reading techniques *for the purpose of* their evaluation *with respect to* their effectiveness for teams

Hypothesis 3: Teams applying PBR perform better than teams using a checklist with respect to their mean defect detection rate.

Goal 3: *Analyze* PBR and checklist reading techniques *for the purpose of* their evaluation *with respect to* their impact on the effectiveness of team meetings in N-fold inspection

Hypothesis 4: There is a difference between the *meeting gains* in the meeting-based N-fold inspection depending on whether teams used PBR or a checklist.

Hypothesis 5: There is a difference between the *meeting loss* in the meeting-based N-fold inspection depending on whether teams used PBR or a checklist.

Goal 4: *Analyze* PBR and checklist *for the purpose of* evaluation *with respect to* detecting unique defects

Hypothesis 6: The overlap of commonly detected defects among perspectives in PBR teams is lower than the overlap among individuals in checklist teams.

Goal 5: *Analyze* PBR and checklist reading techniques s *for the purpose of* evaluation *with respect to* detecting defects in different parts of the document

Hypothesis 7: Defects detected by PBR are more evenly distributed over the whole SRS document than those detected by checklist.

3.2 Variables

The experiment manipulates two independent variables:

1. The **reading technique** (RTECH). Subjects either applied PBR or checklist approach to review the SRS document to detect defects.
2. The **background experience** of subjects (EXP). Though subjects are all graduate students, they have varied levels of background experiences. Some of them have previous industry experiences, while others only have experiences of class projects.

The reading technique is the treatment variable of our experiment. The other variables allow us to access several potential threats to the experiment's internal validity. We also measure the following dependent variables.

1. The **individual defect detection rate** (IDDR): the number of the real defects reported by individual subjects (see section 3.3 for definition of real defects).
2. The **team defect detection rate** (TDDR): the number of the real defects reported by each team.
3. The **meeting gains** (MG): the number of the defects newly detected during each meeting divided by the total number of real defects reported by this team, represented in percentage.
4. The **meeting loss** (ML): the number of individually detected defects not subsequently recorded during the meeting divided by the total number of real defects reported by this team, represented in percentage.

3.3 Design

The experiment described in this paper is a replication of the original PBR study [2] with some modifications. After describing the overall experimental design, we will recap the differences between our replication and the original study.

3.3.1 Subjects

The subjects of the experiment were 12 graduate students of the Computer Science Department at the Mississippi State University, enrolled in Requirement Engineering course in the fall semester, 2005. The experience of the subjects varied greatly, from many years of industry experience to no project experience at all. In order to balance the checklist and PBR groups, we split the subjects into two groups - a *high experience group* and a *low experience group*, based on the background questionnaire they filled out at the beginning of the experiment. A subject was classified into the high-experienced group if he or she had industrial experience in at least one third of areas covered by the background questionnaire (discussed in Section 3.4). The rest of the students were classified as low-experienced because most of their previous experience was in the classroom and not in industry.

We then randomly split the subjects into 2 Checklist teams of 3 subjects each and 2 PBR teams of 3 subjects each, while ensuring that the high-experienced subjects were evenly split between the two. Within the PBR teams, one person was randomly assigned to each of the three perspectives so that each PBR team had a user perspective, designer perspective, and a tester perspective.

3.3.2 Artifact Inspected

The SRS for this study was a real-world requirement document for an SQL upgrade to an appeals tracking system prepared for National Labor Relations Board. Because this SRS was a real document, unlike the SRS used in previous experiments, it was not seeded with a set of known defects before the study. This fact made it more difficult to compute the defects detection rate of the individuals or teams because we did not know exactly how many defects exist in the document. As a solution to this problem, we collected the final defect lists submitted by each team after the N-fold inspection ended (described in more detail later in this section) and came up with a *master defect list.* We considered the defects on this master list as the *real* defects. In reality, this is what happens in the software development process in real world. Anything in the SRS that stakeholders (e.g. users, designers, or testers, etc.) think as ambiguous, inconsistent or incorrect is marked as a *defect*. Then, the requirement author has the obligation to correct the defect, e.g. adding more details or domain knowledge to clarify the ambiguity.

3.3.3 Experimental Operation

The first step was to train the subjects in their assigned technique. On the day of the training the checklist subjects and the PBR subjects went into different classrooms to receive their training. The training sessions were conducted during one lecture meeting for the course (1 hour). The training for the checklist reviewers included a discussion with the course professor about general requirements quality attributes. From this discussion, the 6 checklist subjects along with the professor developed a checklist to guide their inspection of the SRS. The training for PBR was done by an expert in PBR and included a discussion of the theory behind the techniques as well as a case study that provided an example of their use. After the training, the PBR subjects were provided with a detailed protocol to follow to guide their individual inspection.

After the training, the subjects began the study. The first step was to perform an individual inspection of the SRS using either the checklist of their assigned PBR perspective. During this time, each subject reviewed the document and recorded all of the defects he or she saw on a defect form provided by us. The subjects were given 2 days to perform this task. Once the three members of a team were done with their individual inspections, they met together in the first team meeting. There were four of these meetings (Team 1 and 2 were checklist reviewers and Team 3 and 4 were PBR reviewers). During this meeting, the subjects discussed the defects found by each team member and came up with an agreed on final defect list. During this process the team could add new defects that were not found by any reviewer, and eliminate defects from team members' lists that all did not agree with. Finally, the two checklist teams met together and the two PBR teams met together for a second round of meetings. In these six-person meetings, the reviewers examined the two 3-person team defect lists and came up with a final list that all six members agreed with. The experimental design is summarized in Figure 1

3.3.4 Differences between Replication and Original

There were three main differences in this study as compared with the original. First, in this study the subjects were graduate students in a requirements engineering class, while in the original study, the subjects were NASA professionals. The second difference was the artifact inspected. In the original study, the subjects inspected two generic documents and two domain specific documents, all created by the researchers for the purpose of conducting the study. In our case, the subjects inspected a real SRS prepared for a government agency (that had already undergone some review and correction).

Finally, the major difference was the use of the N-fold inspection process. In the original study, the subjects did not even meet as a team, the team defect lists were simulated. In our study, not only do the team members meet, we have a second round of meetings including all reviewers who used the same approach (checklist or PBR). One reason for investigating the importance of the meeting is due to the mixed results of previous studies. The original study simply ignored the impact the meeting could have, while later studies suggested that the real team meetings may have little impact on the final outcome [13].

3.4 Data Collection

The data collection procedure was designed to fit the experiment procedure. Where possible, we used the same data collection forms as the original experiment [2], with a little modification adjusted to our context. For example, we deleted some background questions in the experience questionnaire (e.g. experiences in coding) because they did apply to subjects in our experiment.

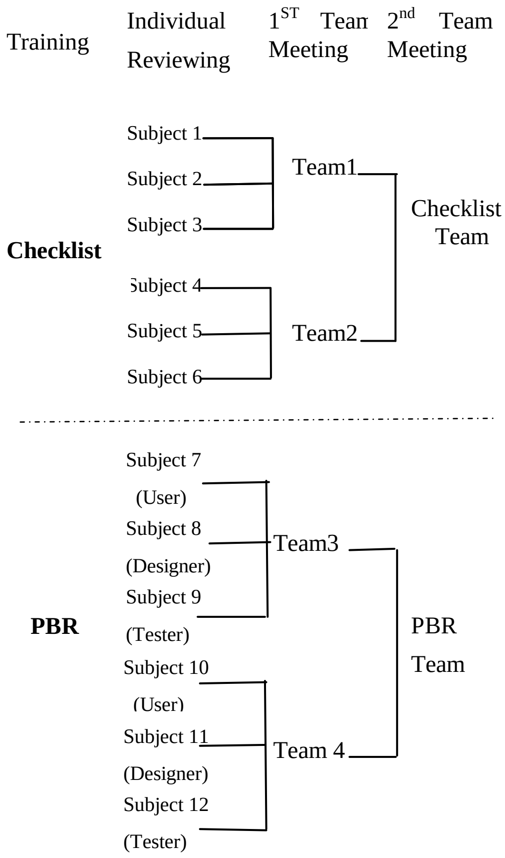

Figure 1 - Experimental Design.

We used a series of forms to collect data. At the beginning of the experiment, the subjects completed the **background and experience questionnaire**, which was used to group the subjects. This questionnaire asked the students about their previous programming experience, their experience working with requirements documents, their experience in design and testing, and their experience with inspections. The next form, **the defect report form**, was used to record information about the defects detected by each subject. This defect form was produced by each individual subjects (12 total), by each 3-person team (4 total) and by each 6-person team (2 total). Finally, the subjects completed a **post-study questionnaire**, to provide feedback on the technique they used and the experiment in general.

4. RESULTS

In this section, we conducted the data analysis to test the hypotheses mentioned in section 3.1. We chose the significant level $\alpha = 0.1$ due to our small sample size.

4.1 Analysis of Individual Performance

Before we analyzed the individual performance with respect to defects detection effectiveness, we first run a 2-way ANOVA test to determine the interaction between two independent variables, i.e. reading techniques (RTECH) and background experiences of subjects (EXP). The results indicate that the interaction between RTECH and EXP is not significant ($F_{12,1} = 2.061$, $p = 0.189$). The results also revealed a significant effect of RTECH ($F_{12,1} = 6.388$, $p = 0.035$) and non-significant effect of EXP ($F_{12,1} = 3.239$, $p = 0.110$).

So we ran an independent-samples t-test and Mann-Whitney test to analyze the effect of reading techniques and subject's experiences on the individual defect detection effectiveness respectively. The advantage of the Mann-Whitney test over t-test r is that the former does not assume normal distribution of the variables. The box plots of the individual detection rate (IDDR) grouped by RTECH and EXP are shown in Figure 2 and Figure 3 respectively. These results basically confirm the results from 2-way ANOVA test. The difference between the mean IDDR of individuals applying PBR and checklist is significant in both tests (t_{10} =-2.286, p=0.045 [t-test]; u=4.5, p= 0.029 [Mann-Whitney]), while the difference between subjects having high experiences and low experiences is not significant in both tests (t_{10} =-1.862, p=0.092 [t-test]; u=7.5, p= 0.101 [Mann-Whitney]).

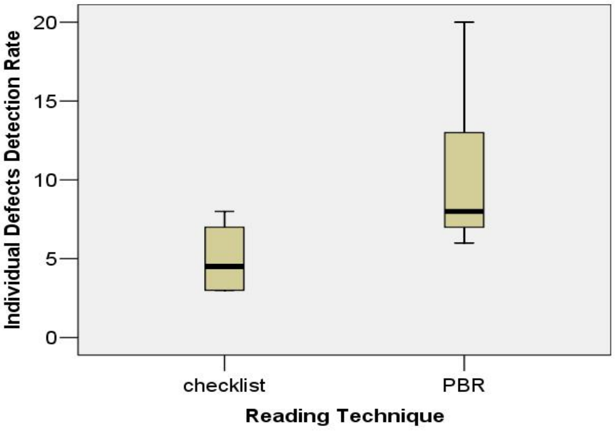

Figure 2. Boxplots of IDDR for Reading Techniques.

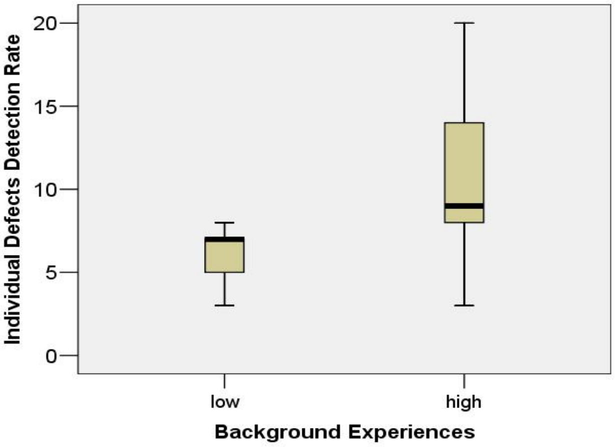

Figure 3. Boxplots of IDDR for Subject's Experiences.

To examine these results in more detail, we analyzed the effectiveness of individuals applying each PBR perspective. The boxplot of IDDR grouped by different perspectives and checklist technique is shown in Figure 4. It shows that each perspective performed better than individuals applying checklist techniques in terms of IDDR. Due to the small number of data points, we did not perform any further statistic analysis.

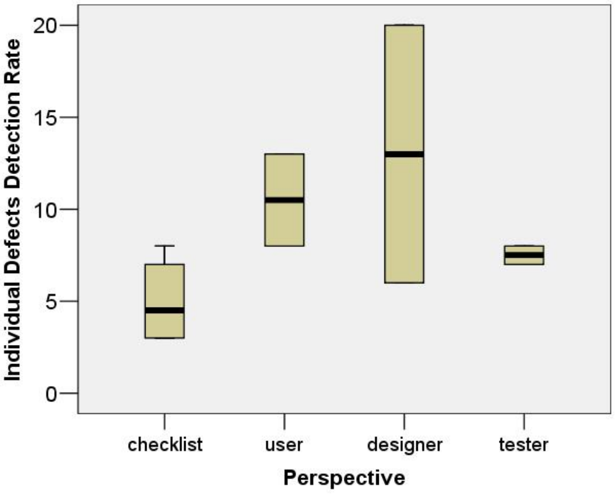

Figure 4. Boxplots of IDDR for PBR perspectives.

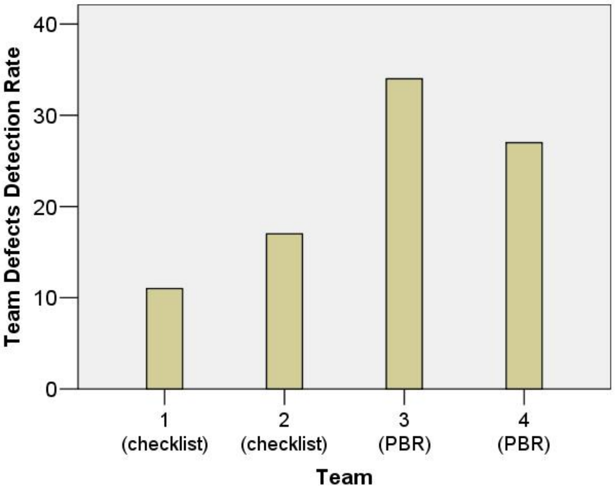

Figure 5. TDDR of Each Team.

4.2 Analysis of Team Performance

To analyze the team performance in defects detection, we also applied the independent-samples t test and Mann-Whitney test to test for significant difference between means of team defect detection rate (TDDR) in two groups (PBR and checklist). Figure 5 presents the bar chart for the TDDR of each team. It shows that both PBR teams detected more defects than checklist teams did. The analysis results (t_2 =-3.579, p=0.070 [t-test]; u=0, p= 0.121 [Mann-Whitney]) reveal that difference between the effectiveness of teams applying PBR and checklist is significant for t-test while not significant for Mann-Whitney test. These results indicate a trend showing that teams applying PBR performed better than the teams applying the checklist technique in terms of defect detection effectiveness, though the statistic result is not as strong as that of individual performance in 4.1.

4.3 Analysis of Team Meetings

We evaluated the effectiveness of team meetings in N-fold inspection in terms of the *meeting gains* (MG) and *meeting loss* (ML), defined in Section 3.2. Table 1 shows the effectiveness of all the team meetings in our N-fold inspection. Same as above, we ran an independent-samples t test and Mann-Whitney test to test our hypotheses. The boxplots of MG and ML grouped by RTECH are shown in Figure 6 and Figure 7 respectively. It shows that the meetings of checklist teams achieved more meeting gains and meeting loss than those of PBR teams. The results indicate that difference between the meeting gains achieved by using PBR and checklist techniques is significant (t_4 =3.031, p=0.039 [t-test]; u=0, p= 0.046 [Mann-Whitney]), while the difference between meeting loss is not significant (t_4 =-1.709, p=0.163 [t-test]; u=1.0, p= 0.127 [Mann-Whitney]).

Table1. Effectiveness of Team Meetings

Team Meeting	3 Person meeting				6 person meeting	
	checklist		PBR		checklist	PBR
	M1	M2	M3	M4	M5	M6
Meeting Gains	6%	8%	3%	0%	3%	0%
Meeting Loss	58%	28%	10%	9%	10%	3%

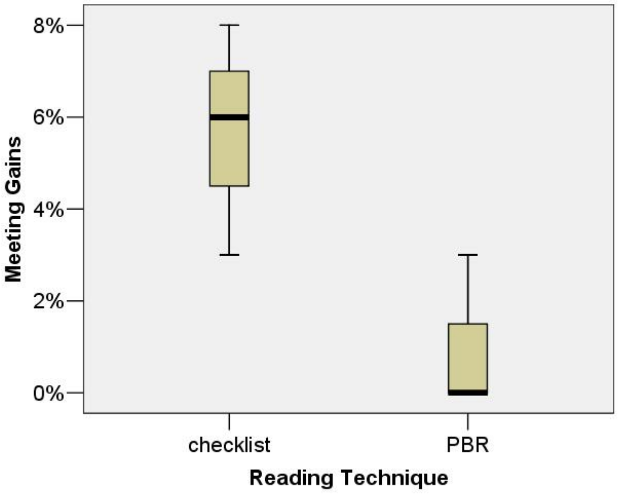

Figure 6. Boxplots of Meeting Gains for Reading Techniques.

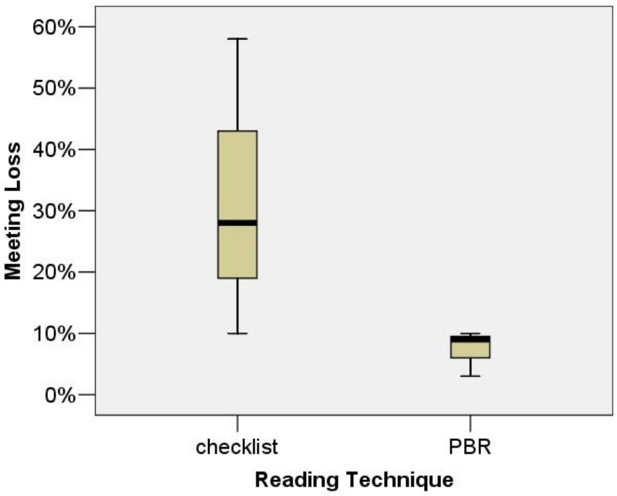

Figure 7. Boxplots of Meeting Loss for Reading Techniques.

4.4 Analysis of Defects Overlap

For each team, we analyzed the overlap of the detected defects among team members, that is, the number of defects found by more than one team member. Figure 8, Figure 9, Figure 10 and Figure 11 show the results of the overlap analysis for Team 1, Team 2, Team 3 and Team 4 respectively. The Venn diagrams show the percentage of defects on the team list that fall into each category along with the actual number of the detected defects in parentheses below. For example, in Figure 8, the same 3 defects, or 30% of the team total, were reported by both S2 and S3. Since the members of PBR team were from different perspectives respectively, Figures 9 and 10 also show the results of overlap analysis for PBR perspectives. These results show that there is little overlap among different PBR perspectives. The bar chart for the overall overlap (the sum of overlaps) of each team is shown in Figure 12. The results indicate that the number of overlapped defects is small in all four teams, but the teams applying checklist technique have a higher percentage of overlapped defects than the teams applying PBR.

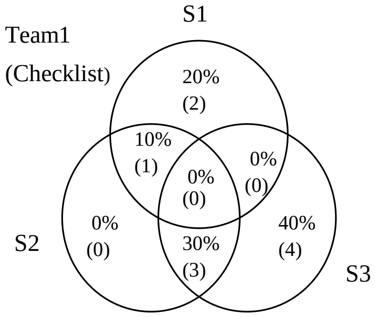

Figure 8. Overlap for Team 1.

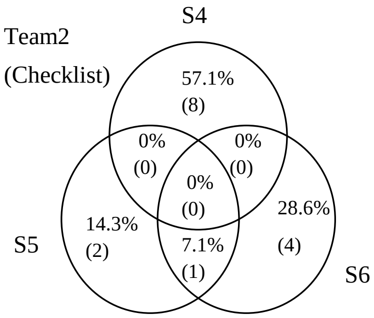

Figure 9. Overlap for Team 2.

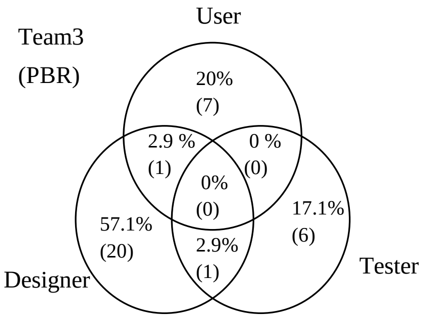

Figure 10. Overlap for Team 3.

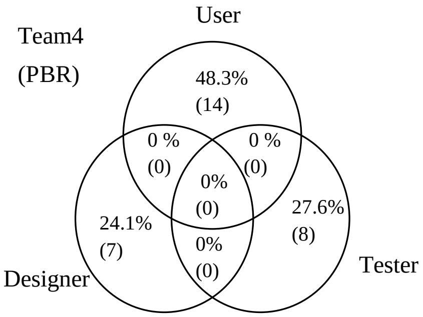

Figure 11. Overlap for Team 4.

4.5 Analysis of Defects Distribution

To determine whether PBR or checklist helps the review team do a better job of finding defects through the document as opposed to focusing on only some sections, we compared the distribution of defects detected by PBR and checklist. Figure 13 shows analysis results of the defect distributions. The X axis shows the section number of requirements in the SRS where the defect was detected e.g. 2.1. The Y axis shows the number of defects detected in each section using each approach. To show the comparison more clearly, we assigned a negative value to the number of defects detected by checklist, so defects distribution line for checklist was shown below the X axis. For example, the checklist technique found 5 defects in section 2.1, so on the figure that became -5 at position 2.1. This approach allows us to more easily compare the number of defects detected by each approach in each section of the SRS. The results indicate that the defects detected by PBR are spread more evenly over the whole SRS document, while those detected by checklist tend to be focused only in some parts of the document.

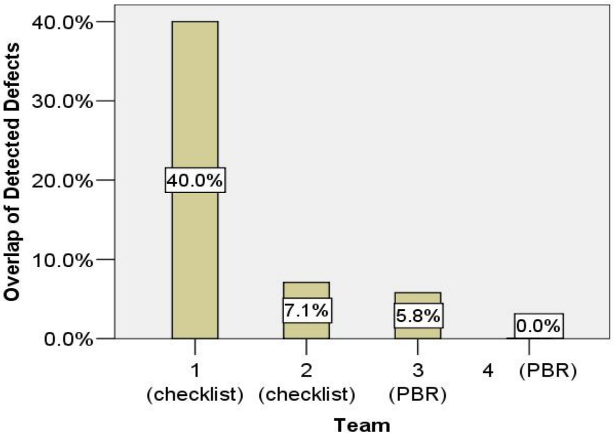

Figure 12. Total Overlap for Each Team.

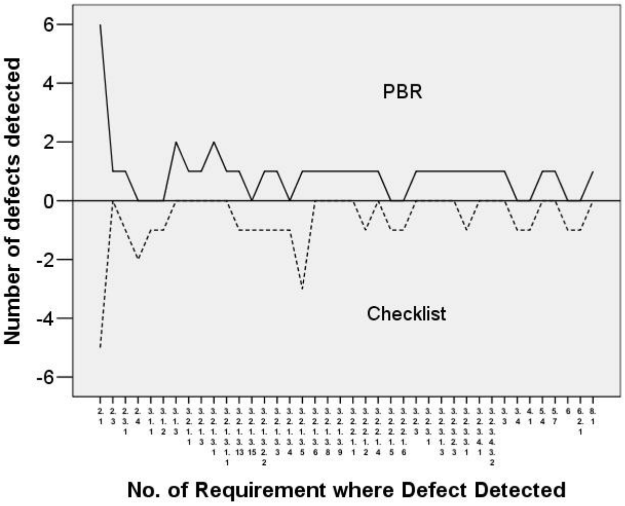

Figure 13. Distribution of Defects Detected in SRS.

5. DISCUSSION

5.1 Threats to validity

A potential problem in any experiment is that some factor may affect the dependent variable without the researcher's knowledge [7]. This possibility must be minimized. We consider two such threats: (1) selection effects and (2) instrumentation effects.

Selection effects are due to natural variation in the performance of the subjects. As we mentioned before, subjects in our experiment have varied level of experience in software development. Therefore, the difference in subjects' natural ability will mask the difference in the reading techniques performance [18]. To limit this effect, we test the interaction of subject's experiences and techniques used in section 4.1. Though the result showed that the interaction is not significant, it was probably due to the way we classify the experiences of a subject as “high” or “low” (see section 3.5). We chose this criterion based on our intuition. Its correctness still needs to be proved by further experiments.

Instrumentation effects may result from the favorability of the SRS to some specific reading technique. Because we had only one SRS, we cannot control this threat in our experiment.

The small sample is also a threat to validity in our experiment. There were only 12 subjects involved in the study, so the number of data points is quite small for some of the statistical analyses. So though the results showed statistic significance, more evidence is needed before fully accepting any of the hypotheses.

5.2 Interpretation of Results

From our analysis of the experimental data, we test the hypotheses listed in section 3.1 and come up with some findings as follows:

Hypothesis 1

> *Individuals applying PBR perform better than individuals using reading techniques with respect to their mean defect detection rate*.

Results from the analysis of individual performance in Section 4.1 show that individuals applying PBR detected more defects than those applying the checklist technique. This data supports the results from some of the previous experiments [2, 4, 13]. In addition, we find that individuals from each perspective all performed better than individuals using checklist.

Hypothesis 2

> *The experience of the subjects has no influence on their mean defect detection rate*.

There is no significant difference between the defect detection effectiveness of subjects who have “high” experiences and “low” experiences. Many factors might contribute to this result. One is the kind of information we collected about the subject’s experiences. Another is the way we classified them (see 3.3.1).

Hypothesis 3

> *PBR teams applying PBR perform better than teams using a checklist with respect to their mean defect detection rate.*

The 2 PBR teams did perform better than the 2 checklist teams. Due to the small number of data points in each group (2), there was not a statistically significant difference between teams applying PBR and checklist with respect to defect detection effectiveness. If we had more data points that followed the same trend, then this result would contradict the results from a previous study that also analyzed the performance of real teams [13].

Hypothesis 4

> *There is a difference between the meeting gains in the meeting-based N-fold inspection using PBR and checklist reading techniques*.

Hypothesis 5

> *There is a difference between the meeting loss in the meeting-based N-fold inspection using PBR and checklist reading techniques*.

Because they are related, we discuss hypotheses 4 and 5 together. There was a difference between the impact of PBR and checklist on the effectiveness of team meetings in N-fold inspection. The checklist teams had a significant increase in meeting gains over the PBR teams. Furthermore, the checklist teams had a sizeable (30% vs. 10%), but non-significant, increase in meeting losses. One likely cause of these results is the different perspectives from which the PBR reviewers approached the SRS. Each PBR reviewer focused on his own perspective and was less concerned with the perspectives of others. While for checklist team, the subjects inspected the SRS using the same checklist and there was more interaction among the team members during the meetings. These results suggest that when using a less procedural technique like a checklist, the team meeting is much more important than when using a technique like PBR.

Hypothesis 6

> *The overlap of commonly detected defects among perspectives in PBR teams is lower than the overlap among individuals in checklist teams*.

The checklist team has more defects overlap among its team members than the PBR team. This result was also likely caused by the fact that each PBR perspective concentrated on its own concern with little overlap in the foci of different perspectives.

Hypothesis 7

> *Defects detected by PBR are more evenly distributed over the whole SRS document than those detected by checklist*.

Defects detected by PBR are more evenly distributed in the SRS document than those detected by checklist techniques. This results suggests that the collection of perspectives from PBR may give a review team a more complete coverage of an SRS than three checklist reviewers can obtain. Because each checklist reviewer is not specifically focused on some aspect of the document, as the PBR reviewers are, this result makes sense. We need further research to determine whether uneven distribution of defects is related to the characteristics of checklist technique itself (preference to positions or defect types) or the specific SRS document used in our experiment.

We did not perform statistical analysis for the last two hypotheses due to the small number of data points. The results support the basic idea of PBR in some sense. The reviewers of each perspective focus on a particular (usually different) aspect of SRS, so there is little overlap among each perspective, and the combination of all the perspectives covers more parts of the document than checklist team does.

6. CONCLUSIONS AND FUTURE WOKR

We tested the effectiveness of PBR in a controlled experiment of the classroom environment. The subjects, graduate students at Mississippi State University, were divided into 4 teams, two of which used checklist approach, and two of which used PBR, to review a real SRS document for defects. Each subject performed an individual review and participated in two N-fold team meetings.

The main goal of our experiment was to test the effectiveness of reading techniques (PBR and checklist) for software inspection in the context of N-fold team meeting. Through the analysis of experimental data, we find that the individuals and teams applying PBR found more defects than those using checklist. Furthermore, checklist teams had more effective team meetings during the N-fold inspection process. The defects detected by PBR teams showed less overlap and were more evenly distributed through the whole SRS document than those detected by the checklist teams.

The most important, and novel, conclusion that we can draw from these results is that the effectiveness and necessity of a team meeting depends greatly on the type of technique used for the individual reviews. In the case of PBR, the team meeting served little purpose, that is very few new defects were found while very few false positives were eliminated from individual team member's lists. Conversely, in the case of a checklist, the meeting is a necessary part of the process. In the team meetings not only were new defects detected, but also a large percentage of defects on the individual defect lists were eliminated as being false positives, thus potentially saving time in the rework phase.

The design and execution of this replication has also inspired our thoughts in the following aspects concerning the improvement of PBR or inspection techniques in general, which leaves space for future work:

(1) How can we improve the effectiveness of meeting-based N-fold inspection when using PBR approach?

From the results of our experiment, we noticed that PBR teams have a less effective team meeting in terms of team gains and loss compared with checklist teams. One possible way to improve the effectiveness of the meetings is to put more than one person from each perspective in the same PBR teams. Thus more than one member of each inspection team will have the same perspective and follow the same inspection procedure which might allow them to discuss more effectively during the team meeting. Future research can be done on the relationship between the organization/structure of the PBR teams and the effectiveness of the team meetings.

(2) Effects of the experience of subjects

The experience of subjects concerns two aspects: their experience in reviewing requirements documents and in participating in software development in the role of user, designer or tester. The findings in our experiment support the results of the original experiment, which showed no significant relationship between PBR defect detection rate and the experience of subjects. In our experiment we recorded the subject's experiences using ordinal metrics and simplified the data by classifying them into two groups. In the future, we could build a more complex model to establish the relationship between the experiences and performance of subjects. The combination of statistical analysis methods and other techniques like genetic programming maybe a possible approach. Other questions that need to be taken into consideration include how to write questions about the subject's experience and in the right format.

7. ACKNOWLEDGEMENT

We gratefully acknowledge Dr. Ray Vaughn for allowing us to conduct the study in his class. We would also thank the students of RE class for their participation.

8. REFERENCES

[1] Ballman, K., and Votta, L.G. Organizational congestion in large-scale software development. In *Proceeding of the Third International Conference on the Software Process.* IEEE CS Press, Los Alamitos, CA, 1994, 123-34.

[2] Basili, V. R., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sørumgård, S., and Zelkowitz, M. V. The Empirical Investigation of Perspective-Based Reading. *Empirical Software Engineering: An International Journal*, 1, 2 (1996), 133-164.

[3] Basili, V. R., Selby, R., and Hutchens, D. Experimentation in Software Engineering. *IEEE Transactions on Software Engineering*, 12, 7 (Jul. 1986), 733-743.

[4] Ciolkowski, M., Differding, C., Laitenberger, O., and Munch, J. *Empirical Investigation of Perspective-Based Reading: A Replicated Experiment*, Technical Report ISERN-97-13, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, 1997.

[5] Fagan, M. Design and code inspections to reduce errors in program development. *IBM System Journal*, 15, 3 (1976), 182-211.

[6] Freedman, D., and Weinberg, G. *Handbook of Walkthroughs Inspections, and Technical Reviews*. Dorset House Publishing, 1990.

[7] Gilb, T. and Graham, D. *Software Inspections*. Addison-Wesley, London, England, 1993.

[8] Humphrey, W. S. *A Discipline for Software Engineering, Reading*. Addison-Wesley, Mass., 1995.

[9] Johnson, P., & Tjahjono, D. Assessing software review meetings: A controlled experiment study using CSRS. In *Proceeding of the 19th International Conference on Software Engineering (ICSE'97)*. ACM Press, Los Angles, CA, 1997, 118-27.

[10] Kamsties, E., and Lott, C. M. An Empirical Evaluation of Three Defect Detection Techniques. *Proceedings of the Fifth European Software Engineering Conference*, (Sitges, Barcelona, Spain, September 6-10, 1995). Springer, 1996, 362-383.

[11] Knight, J. An Improved Inspection Technique. *Communications of the ACM*, 36, 11(Nov. 1993), 51-61.

[12] Laitenberger, O., and DeBaud, J. *An Encompassing Life-Cycle Centric Survey of Software Inspection.* Technical Report ISERN-98-32, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, 1998.

[13] Lanubile, F. and Visaggio, G. *Evaluating Defect Detection Techniques for Software Requirements Inspections*, Technical Report ISERN-00-08, International Software Engineering Research Network.

[14] Martin, J. and Tsai, .W. N-Fold Inspection: A Requirements Analysis Technique. *Communications of the ACM*, 33, 2(Feb. 1990), 223-232.

[15] McCarthy, P., Porter , A., Sih , H., and Votta , L. An Experiment to Assess Cost Benefits of Inspection Meetings and their Alternatives: A Pilot Study. *Proceedings of the 1996 IEEE International Software Metrics Symposium*

(Berlin, Germany, March 25-26, 1996). IEEE Computer Society Press, 1996, 100-111.

[16] Perpich, J. M., Perry, D. E., and Porter, A. A. Anywhere, anytime, code inspections: Using the Web to remove inspection bottlenecks in large-scale software development. *Proceeding of the 19th International Conference on Software Engineering (ICSE'97)*. ACM Press, Los Angles, CA, 1997, 14-21.

[17] Porter, A., and Johnson, P. Assessing software review meeting: Results of a comparative analysis of two experimental studies. *IEEE Transaction on Software Engineering,* 23, 3 (1997), 129-145.

[18] Porter, A., Votta, L. G., and Basili, V. R. Comparing Detection Methods for Software Requirements Inspections: A Replication Using Professional Subjects. *Empirical Software Engineering,* 3(1998), 355-379.

[19] Porter, A., Votta, L. G. Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment. *IEEE Transactions Software Eng.,* 21, 6 (Jun. 1995), 563-575.

[20] Russel, G.W. (1991). Experience with inspection in ultralarge-scale developments. *IEEE Software,* 8, 1 (1991), 25-31.

[21] Sapsomboon, B. Shared Defect Detection : The *Shared Defect Detection: The Effects of Annotations in Asynchronous Software Inspection.* Ph.D. Thesis, University of Pittsburg, Pittsburg, PA, 2000.

[22] Schneider, G. M., Martin, J., and Tsai, W.T. An experimental study of fault detection in user requirements documents. *ACM Transaction on Software Engineering*, 1, 2, 188-204.

[23] Shull, F. *Developing Techniques for Using Software Documents: A Series of Empirical Studies*. Ph.D. Thesis, University of Maryland, College Park, MD, 1998.

[24] Shull, F., Carver, J., and Travassos, G. An Empirical Methodology for Introducing Software Processes. In *Proceedings of the Joint 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9)* (Vienna, Austria, September 10-14, 2001). ACM Press, 2004, 288-296.

[25] Shull, F., Mendonça, M., Basili, V., Carver, J., Maldonado, J., Fabbri, S., Travassos, G., and Ferreira, M. Knowledge-Sharing Issues in Experimental Software Engineering. *Empirical Software Engineering: An International Journal*, 9, 1(Mar. 2004), 111-137.

[26] Shull, F., Rus, I., and Basili, V. How Perspective-Based Reading Can improve Requirements Inspection. *IEEE Software*, 33, 7 (Jul. 2000), 73-79.

[27] Stein, M., Riedl, J., Harner, S.J., & Mashayekhi, V. (1997). A case study of distributed, asynchronous software inspection. *Proceeding of the 19th International Conference on Software Engineering (ICSE'97)*. ACM Press, Los Angeles, CA, 1997, 107-117.

[28] Thompsom, C., & Riedl, J. (1995). *Collaborative asynchronous inspection of software using Lotus Notes*. Technical Report 95-047, Computer Science Department, University of Minnesota.

[29] Tripp, L., Struck, W., and Pflung, B. The application of multiple team inspections on a safety-critical software standard. *Proceeding of 4th Software Engineering Standards Application Workshop*. IEEE CS Press, Los Alamitos, CA, 1991, 106-11

[30] Votta, L. G. Does every inspection need a meeting? *Proceedings of ACM SIGSOFT '93 Symposium on Foundations of Software Engineering (FSE'93)* (Los Angeles, California, December 7-10, 1993).ACM Press, 1993, 107-114.

[31] Wheeler, D. A., Brykczynski, B, and Meeson, R. N. S*oftware Inspection: An Industry Best Practice*. IEEE CS Press, Los Alamitos, CA, 1996.

[32] Zhang, Z., Basili, V., and Shneiderman, B. Perspective-Based Usability Inspection: An Empirical Validation of Efficacy. *Empirical Software Engineering - An International Journal*, 4, 1(1999), 43-70.

.