A Controlled Experiment to Assess the Effectiveness of Inspection Meetings

Alessandro Bianchi, Filippo Lanubile, and Giuseppe Visaggio

Dipartimento di Informatica
University of Bari
Bari, Italy
{bianchi, lanubile, visaggio}@di.uniba.it

Abstract

Software inspection is one of the best practices for detecting and removing defects early in the software development process. In a software inspection, review is first performed individually and then by meeting as a team. In the last years, some empirical studies have shown that inspection meetings do not improve the effectiveness of the inspection process with respect to the number of true discovered defects. While group synergy allows inspectors to find some new defects, these meeting gains are offset by meeting losses, that is defects found by individuals but not reported as a team.

We present a controlled experiment with more than one hundred undergraduate students who inspected software requirements documents as part of a university course. We compare the performance of nominal and real teams, and also investigate the reasons for meeting losses. Results show that nominal teams outperformed real teams, there were more meeting losses than meeting gains, and that most of the losses were defects found by only one individual in the inspection team.

1. Introduction

Software inspection is a structured process for the static verification of software documents, including requirements specifications, design documents as well as source code. From the seminal work of Fagan [5, 6] to its variants [8, 9], the software inspection process is essentially made up of four consecutive steps: planning, preparation, meeting, and rework.

The main changes from the original Fagan's inspection have been a shift of primary goals for the preparation and meeting steps. The main goal for preparation has changed from pure understanding to defect detection, and so inspectors have to individually take notes of defects. Consequently, the main goal of the inspection meeting has been reduced from defect discovery to defect collection, including the discussion of defects individually found during preparation.

In the attempt to shorten the overall cost and total time of the inspection process, the need for a meeting of the whole inspection team has been debated among researchers and practitioners. Parnas and Weiss [15] first dropped the team meeting in their Active Design Reviews, which had another fundamental difference from Fagan's inspections in the separation of concern applied to the preparation step, with individual inspectors using specialized and different checklists as defect detection helpers. Then Votta [21] showed how defect collection meetings lengthened the elapsed time of software inspections at Lucent Technologies's Bell Labs of almost one third, with defects discovered at the meeting (*meeting gains*) matched by defects not recorded at the meeting although found during preparation (*meeting losses*). Further studies [4, 7, 11, 13, 14, 16, 17] have also observed that the *net meetings improvement* (difference between meeting gains and meeting losses) was not positive and then *nominal teams* (teams who do not interact in a face-to-face meeting) are at least equivalent to *real teams* but with lower cost and time. However, meetings have been found useful for filtering out false positives (defects erroneously reported as such by inspectors), training novices, and increasing self-confidence [10, 11, 13].

Among the many sources of variations in software inspections, Porter et al. [18] have shown that changes in the inspection process structure can cut inspection cost and shorten the inspection interval but do not improve the inspection effectiveness (basically measured as the number or density of defects found).

We have further investigated the variation sources in software inspections by means of a controlled experiment in a classroom environment with more than one hundred undergraduate students. In this paper, we focus on the use of meetings and then we report only those aspects of the experiment which are relevant for it (the experiment also included the study of the effects of systematic reading techniques on defect detection and the effects of having distinct roles when composing inspection teams; these issues will be the subjects of future reports).

A real team, i.e., a team who interacts in a face-to-face meeting, can both find new defects because of synergy group, and leave out defects found during preparation because of negative acknowledgement. The main research question was the following:

RQ1: Are there differences in the number or density of defects found (inspection effectiveness) between defect collection by inspection meetings (real teams) and defect collection by merging individual reports (nominal teams)?

Based on findings from previous studies, our hypothesis was the following:

Hyp1: Real teams do not find a higher number of defects than nominal teams.

However, because a nominal team can neither have meeting gains nor meeting losses, the above hypothesis above can be restated as:

Hyp1b: Meeting gains are no more than meeting losses.

We also wanted to investigate the group dynamics that might cause meeting losses, by examining the relations between meeting losses, defects reported by teams, and the overlapping of individual discoveries in a team. So, we explored the following other research questions:

RQ2: Are there differences in the number of meeting losses between defects found (during preparation) by only one reviewer in a team and defects found (during preparation) by more than one reviewer in a team?

RQ3: Are there differences in the number of true defects reported as a team between defects found (during preparation) by only one reviewer and defects found (during preparation) by more than one reviewer in a team?

To our knowledge, these relations have not been previously investigated, and then there are no past findings to be used as hypotheses to be confirmed or rejected.

The remainder of this paper is organized as follows. Section 2 describes the experiment, Section 3 presents the results from data analysis, Section 4 discusses the validity threats to the experiment, and the final section summarizes and discusses our findings.

2. The Experiment

The experiment was conducted as part of a two-semester software-engineering course at the University of Bari. The experiment simulated in a classroom environment, with more than one hundred undergraduates, the preparation and meeting steps of an inspection process for requirements documents.

2.1. Experimental Design

We conducted two runs of the experiment, each run requiring subjects to inspect a requirement document, starting with an individual preparation and finishing with a team meeting step. Some differences between runs were planned in advance while some changes were introduced after the first run was over.

The planned change, which is relevant for the team meeting stage, was the document to be inspected:

- ATM in the first run and PG in the second run (see next section for a brief description of these documents)

Students were randomly assigned to three-person inspection teams, but in some cases we had to create four-people teams because of spare people to accommodate in a team.

The unplanned change, associated to team meetings, was that we had to rearrange the composition of some teams because of some subject withdrawals between the two runs.

For each experiment run, the independent variable is the type of team interaction, with two values: lack of team interaction (nominal team) and face-to-face interaction (real team). Because we have repeated measurements of this same variable (under different conditions) on the same subjects (teams), then the experimental plan of each experiment run is a repeated measure design and the independent variable is a within-subjects factor.

We measured the following dependent variables:

- Nominal team true defects (*NOMTDEF*): the number of true defects obtained by merging individual reports of a same team
- Nominal team defect percentage (*NOMTPCT*): nominal team true defects divided by the total number of known defects in the document
- Real team true defects (*REALTDEF*): the number of true defects reported by a team at inspection meeting
- Real team defect percentage (*REALTPCT*): real team true defects divided by the total number of known defects in the document

- Meeting gains (*GAINS*): the number of true defects first found during an inspection meeting
- Meeting losses (*LOSSES*): the number of true defects reported by an individual inspector but erroneously omitted in the meeting defect report
- Net meeting improvement (*NETIMPR*): the difference between real teams true defects and nominal team true defects
- Defects lost by one inspector (*LOSTBY1*): the number of true defects reported by only one individual inspector but erroneously omitted in the meeting defect report
- Defects lost by many inspectors (*LOSTBYM*): the number of true defects reported by more than one individual inspector but erroneously omitted in the meeting defect report
- Defects collected by one inspector (*COLLBY1*): the number of true defects reported by only one individual inspector and included in the meeting defect report
- Defects collected by many inspectors (*COLLBYM*): the number of true defects reported by more than one individual inspector and included in the meeting defect report
- Time for meeting (*MTNGTIME*): the time spent for meeting, in minutes

The following equations hold among the dependent variables:

$$\begin{aligned} NETIMPR &= REALTDEF — NOMTDEF \\ &= GAINS — LOSSES \end{aligned} \qquad (1)$$

$$LOSSES = LOSTBY1 + LOSTBYM \qquad (2)$$

$$REALTDEF = COLLBY1 + COLLBYM + GAINS \qquad (3)$$

2.2. Experimental Material

The experiment has reused most of the material from a previous experiment [12] which is part of a family of experiments on software reading techniques [3]. The material is available as a lab package on the web [20] but we had to translate everything from English to Italian otherwise many students would not be confident with reading and using it.

The material includes requirements documents, general instructions, instructions and defect detection aids for the preparation step, defect report forms to be used both for the individual preparation and the team meeting, and debriefing questionnaires.

The software requirements specifications were written in natural language and adhered to the IEEE format for SRS (IEEE, 1984). The requirements documents used for the experiment were:

- Automated Teller Machine (ATM), 17 pages long and containing 29 defects
- Parking Garage control system (PG), 16 pages long and containing 27 defects

2.3. Training and Preparation

All subjects taking a course in software engineering for undergraduates were prepared with a set of lectures on requirements specifications and software inspections.

We gave a 2-hour lecture on the IEEE standard for SRS and taught a requirements defect taxonomy. A requirements document for a course scheduler system was presented and an assignment was given for finding defects. The results were discussed in class and a list of known defects was written out according to the schema of defect report forms.

Another 2-hour lecture was given on software inspections, explaining the goals and the specific process to be used in this study. We then introduced a new requirements document for a video rental system, which was available in the experiment lab package for training purposes. As a trial inspection, students were asked to individually read the document and record defects on the defect report forms to be used in this experiment. We then created teams, assigned roles inside the teams (moderator, reader, and recorder) and a trial inspection meeting was conducted. After the trial inspection we discussed with students the list of known defects and what defects they had found.

Finally, we spent one lecture to present the defect detection techniques for the preparation step and the experiment organization. We also communicated the outcomes of randomly assigning subjects to the experimental conditions. Teams were let free to choose team roles as moderator, reader, and recorder.

2.4. Running the Experiment

The experiment was run as a midterm exam of an undergraduate software engineering course. Each experiment run, corresponding to a separate inspection (ATM document first and then PG document), took two consecutive days, one for individual preparation and one for team meeting. The second run was scheduled after one week from the first run.

Subjects always worked in two big rooms with enough space to avoid plagiarism and confusion. We were always present to answer questions and preventing unwanted communication. Each experimental task was limited to four hours and, before leaving, subjects were asked to complete a debriefing questionnaire.

Before each individual preparation step, subjects were given a package containing the requirements document, specific instructions for the assigned reading technique, and blank defect report forms. After each individual preparation step, we collected all the material. This material was returned to subjects before the inspection meeting together with new blank defect report forms. At the inspection meeting, the reader paraphrased each requirement and the team discussed defects found during preparation or any new defect. The moderator was responsible for managing discussions and recorder for filling out the team's defect report forms.

3. Data Analysis

We validated the reported defects by comparing location and description information with those in the master defect list from a former experiment on requirements inspection techniques [1]. All the reported defects that could be matched to some known defect were considered true defects. Real team true defects were collected through team defect report forms, while nominal team true defects were collected through the merge of individual defect report forms in a team. Meeting losses and meeting gains were collected by comparing team defect report forms and individual defect report forms.

In the following, we first present some descriptive statistics for the dependent variables, and answer to the first two research questions. Then, we perform some exploratory analysis by looking at the relationships between dependent variables. Finally, we answer to the remaining research questions by testing for differences between matched dependent variables.

3.1. Descriptive Statistics

Table 1 and Table 2 present some basic information for the two runs of the experiment, such as the number of valid observations, mean, confidence intervals, minimum and maximum values, and standard deviation.

The tables show that real teams detected on average between 39% and 45% of the defects in the ATM document, and between 33% and 39% of the defects in the PG document. Nominal teams detected on average between 44% and 52% of the defects in the ATM document, and between 42% and 49% of the defects in the PG document. These percentages are in line with the ones reported in a former experiment with nominal teams, made up of NASA professionals, applying their usual review technique [1].

The mean values for meeting gains and meeting losses are positive for both runs of the experiment. The confidence intervals for the means give a range of values around the means where we expect the "true" means are located, with a given level of certainty (95% for a p=0.05 confidence interval). The lower limits of the meeting gains mean are 0.95 and 0.79 (respectively for the ATM and PG documents), while the lower limits of the meeting losses mean are 2.5 and 3.3 (respectively for the ATM and PG documents). However, the meeting gains variable does not met normality assumption and so the estimate of confidence intervals may not be valid.

Variable	Valid N	Mean	Confid. -95.000%	Confid. +95.000%	Minimum	Maximum	Std.Dev.
NOMTDEF	37	14.0000	12.7649	15.2351	8.00000	25.0000	3.70435
NOMTPCT	37	.4824	.4396	.5253	.28000	.8600	.12848
REALTDEF	37	12.2432	11.3881	13.0984	7.00000	17.0000	2.56478
REALTPCT	37	.4222	.3925	.4519	.24000	.5900	.08907
GAINS	37	1.4054	.9508	1.8600	0.00000	6.0000	1.36340
LOSSES	37	3.1622	2.4793	3.8451	0.00000	9.0000	2.04822
NETIMPR	37	-1.7568	-2.6746	-.8389	-9.00000	5.0000	2.75283
LOSTBY1	37	2.7297	2.1382	3.3212	0.00000	7.0000	1.77402
LOSTBYM	37	.4324	.1897	.6752	0.00000	2.0000	.72803
COLLBY1	37	5.4865	4.9329	6.0401	2.00000	8.0000	1.66035
COLLBYM	37	5.3514	4.5491	6.1536	1.00000	10.0000	2.40620
MTNGTIME	37	149.8919	140.4992	159.2845	80.00000	210.0000	28.17089

Table 1. Descriptive statistics for the first run (ATM document)

Variable	Valid N	Mean	Confid. -95.000%	Confid. +95.000%	Minimum	Maximum	Std.Dev.
NOMTDEF	35	13.2857	12.2736	14.2978	6.00000	20.0000	2.94630
NOMTPCT	35	.4571	.4220	.4923	.21000	.6900	.10234
REALTDEF	35	10.4571	9.5281	11.3862	5.00000	16.0000	2.70449
REALTPCT	35	.3594	.3276	.3913	.17000	.5500	.09280
GAINS	35	1.2286	.7870	1.6701	0.00000	5.0000	1.28534
LOSSES	35	4.0571	3.3311	4.7832	1.00000	8.0000	2.11358
NETIMPR	35	-2.8286	-3.7003	-1.9568	-8.00000	3.0000	2.53778
LOSTBY1	35	2.7714	2.1877	3.3551	0.00000	7.0000	1.69923
LOSTBYM	35	1.2857	.8988	1.6727	0.00000	4.0000	1.12646
COLLBY1	35	3.4857	2.8330	4.1384	0.00000	8.0000	1.90002
COLLBYM	35	5.7429	5.1175	6.3682	1.00000	10.0000	1.82052
MTNGTIME	35	148.6000	137.7576	159.4424	85.00000	215.0000	31.56338

Table 2. Descriptive statistics for the second run (PG document)

Figure 1 presents the distributions of the two variables for both documents using boxplots. Boxplots graphically show some ordinal descriptive statistics, such as median, quartiles, and quartile range. For meeting losses, the median and quartile values are clearly positive, but for meeting gains, especially for the PG document, there are about 25 percent of the cases with zero meeting gains.

The average meeting time is approximately two hours and a half for both the experiment runs, with a standard deviation of about one half hour. The maximum meeting time is about three hours and a half. Thus, no team was pressed to end the meeting because of the four hours time limit.

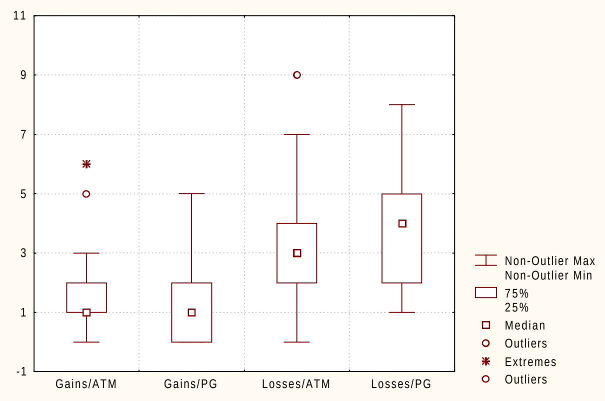

Figure 1. Boxplots of meeting gains and losses on the two documents

3.2. Exploring Relationships between Variables

We first wanted to verify whether the amount of time available for meeting, might have influenced the team interaction. Table 3 shows for both documents the correlation coefficients between meeting time and the dependent variables related to team performance. We use a nonparametric correlation coefficient, Spearman *R*, which only assumes that the variables under consideration were measured on at least an ordinal scale. As can be seen, there is no correlation between time and team performance.

Pair of Variables	Spearman *R*	
	ATM	PG
MTNGTIME & NOMTDEF	-.214081	-.119964
MTNGTIME & REALTDEF	-.309926	.098098
MTNGTIME & LOSSES	.202246	-.291800
MTNGTIME & GAINS	.285493	.117192
MTNGTIME & NETIMPR	-.050361	.287004
MTNGTIME & LOSTBY1	.114978	-.356604
MTNGTIME & LOSTBYM	.099270	-.104289
MTNGTIME & COLLBY1	-.239793	.090383
MTNGTIME & COLLBYM	-.372485	.093668

Table 3 . Correlation between meeting time and team performance variables

Then, we wanted to analyze the relationship between those team performance variables included in equations (1), (2), and (3). Table 4 shows the Spearman rank order correlations between each variable on the left-side part of the equations and the variables on the right-side part. As the parametric Pearson *r*, Spearman *R* can be interpreted in terms of the proportion of variability accounted for, except that Spearman *R* is computed from ranks. As can be seen, there is a strong negative correlation between the net meeting improvement and meeting losses (Spearman *R* are -0.88 and -0.86, respectively for ATM document and PG document) and a strong positive correlation between meeting losses and defects lost by one inspector (Spearman *R* are 0.92 and 0.86, respectively for ATM document and PG document).

Pair of Variables	Spearman *R*	
	ATM	PG
NETIMPR & NOMTDEF	-.679800	-.531094
NETIMPR & REALTDEF	.037028	.415316
NETIMPR & LOSSES	-.880575	-.861857
NETIMPR & GAINS	.643350	.487023
LOSSES & LOSTBY1	.922000	.859384
LOSSES & LOSTBYM	.454820	.597919
REALTDEF & GAINS	.224159	.371076
REALTDEF & COLLBY1	.518403	.593939
REALTDEF & COLLBYM	.566095	.542269

Table 4. Correlation between some team performance variables

3.3. Testing for differences

Because the two groups of observations that are to be compared are based on the same sample of cases (teams), which were measured twice, we might use the *t*-test for dependent samples. However, since the normality assumption was not always respected, we decided to use the Wilcoxon matched pairs test. This nonparametric alternative only assumes that the two variables are on an ordinal scale and that the differences between variables can be rank ordered too.

We run a total of six tests, one for each pair of research question and document. In order to lower the probability of getting a significant result purely by chance, we control the level of significance for a set of tests through the Dunn-Bonferroni procedure [23].

Briefly, an experimenter may obtain the significance level for a single test as $\alpha_{ind} = \alpha_{expw} / m$, where α_{expw} is the desired level of significance for the entire experiment and *m* is the number of tests in the experiment. In our case, if we set α_{expw} to 0.05, we will need a *p*-value less than 0.0083 ($\alpha_{ind} = 0.05 / 6$) to conclude that a single test has found a significant difference.

We first tested the main research question by comparing the nominal team true defects (i.e., the number of true defects obtained by merging individual reports of a same team) and the real team true defects (i.e, the number of true defects reported by a team at inspection meeting). Figure 2 shows boxplots of the two variables for both documents.

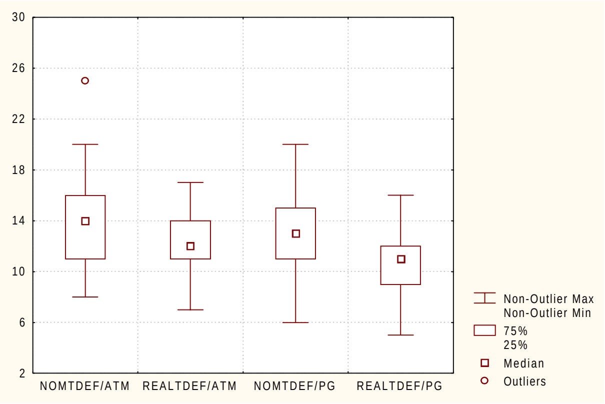

Figure 2. Boxplots of nominal team and real team true defects on the two documents

For the two documents, the null and alternative hypotheses can be formulated as follows:

$H1_0$: There is no difference between nominal team true defects (*NOMTDEF*) and real team true defects (*REALTDEF*)

$H1_a$: There is a difference between nominal team true defects (*NOMTDEF*) and real team true defects (*REALTDEF*)

The analysis found a significant difference between the two variables (*p* = 0.000801 for ATM document and *p* = 0.000009 for PG document), with nominal teams finding defects more often than real teams. This finding can be rephrased saying that there were more meeting losses than meeting gains.

We then tested the second research question by comparing defects lost by one inspector (i.e., the number of true defects reported by only one individual inspector but erroneously omitted in the meeting defect report) and defects lost by many inspectors (i.e., the number of true defects reported by more than one individual inspector but erroneously omitted in the meeting defect report). Figure 3 shows boxplots of the two variables for both documents.

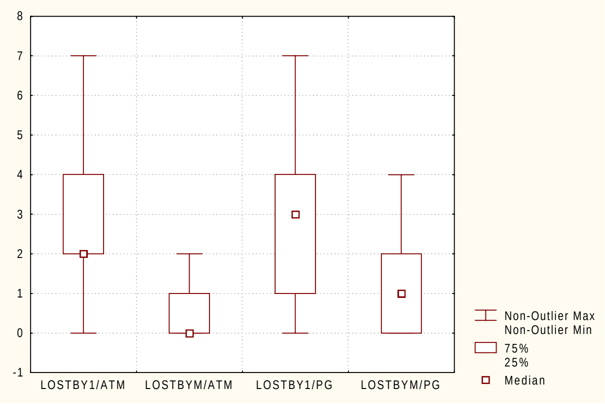

Figure 3. Boxplots of defects lost by one inspector and by many inspectors on the two documents

For the two documents, the null and alternative hypotheses can be formulated as follows:

$H2_0$: There is no difference between defects lost by one inspector (*LOSTBY1*) and defects lost by many inspectors (*LOSTBYM*)

$H2_a$: There is a difference between defects lost by one inspector (*LOSTBY1*) and defects lost by many inspectors (*LOSTBYM*)

The analysis found a significant difference between the two variables (p = 0.000001 for ATM document and p = 0.000079 for PG document), with meeting losses being more frequent for defects found by one inspector than for defects found by more than one inspector.

We finally tested the third research question by comparing defects collected by one inspector (i.e., the number of true defects reported by only one individual inspector and included in the meeting defect report) and defects collected by many inspectors (i.e., the number of true defects reported by more than one individual inspector and included in the meeting defect report). Figure 4 shows boxplots of the two variables for both documents.

For the two documents, the null and alternative hypotheses can be formulated as follows:

$H3_0$: There is no difference between defects collected by one inspector (*COLLBY1*) and defects collected by many inspectors (*COLLBYM*)

$H3_a$: There is a difference between defects collected by one inspector (*COLLBY1*) and defects collected by many inspectors (*COLLBYM*)

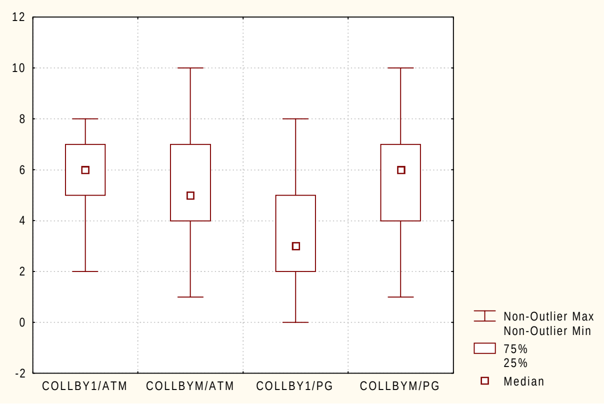

Figure 4. Boxplots of defects collected by one inspector and by many inspectors on the two documents

The results were different between the two documents. In the ATM document, the analysis failed to reveal any significant difference between the two variables (p = 0.687886) while in the PG document the analysis found a significant difference between the two variables (p = 0.000079), with defects reported by a team being less frequent for those collected by one inspector than for those collected by more than one inspector.

4. Threats to Validity

This section discusses the threats to validity that are relevant for our experiment.

Threats to internal validity are factors beyond the experimenter's control, which might affect the dependent variables and then causing problems in the correct interpretation of findings. We identified the following threats to internal validity:

Plagiarism. Because the experimental tasks were part of a midterm exam, the highest risk event is plagiarism, with subjects exchanging information about defects in the intervals between tasks. While plagiarism could not occur between the two experimental runs because the requirements documents were different, it might be the case for the two one-day intervals between individual preparations and team meetings. To reduce this risk, we told students that only individual tasks were subject to grading. Furthermore, the individual defect lists were collected after individual preparation and returned to subjects just before the team meeting.

Learning. We cannot exclude that learning was still in progress during the experiment. We tried to minimize the learning effect by teaching requirements specification and review and having a training session before the experiment itself.

Boredom. As the learning effect, boredom occurs over time, but while learning tends to amplify subjects' performance, boredom tends to degrade the performance. The boredom effect might have affected the second run of the experiment, because subjects had to perform a second complete inspection using the same review technique. This might explain why for the PG document, there were less meeting gains and more meeting losses together with fewer defects collected by one inspector than by more than one inspector.

Threats to external validity are factors that limit the generalization of the experimental results to the context of interest, here the industrial practice of software inspections. For our experiment, we can identify the following threats to external validity:

Representative subjects. Our students may not be representative of the population of software professionals. However, a former experiment with NASA developers [1] failed to reveal significant relationship between inspection effectiveness and reviewers' experience. Probably, being a software professional does not imply that the experience matches with the skills that are relevant to the object of study. Based on the behavioral theory of group performance, Sauer et al. [19] state that task expertise is the dominant determinant of review performance and recommend training to increase to develop reviewers' skills. Since this experiment was part of a software engineering course, we had a chance to train students on both defect detection techniques and inspection process.

Representative artifacts. The requirements documents inspected in this experiment may not be representative of industrial requirements documents. Our documents are smaller and simpler than industrial ones although in the industrial practice long and complex artifacts are inspected in separate pieces. Furthermore, we cannot exclude that meeting losses and meeting gains would occur with the same frequency also for other software artifacts, such as design documents and code.

Representative processes. The inspection process in this experiment may not be representative of industrial practice. Although there are many variants of the inspection process in the literature and industry, we conducted inspections on the basis of a widely spread inspection process [22]. However, our inspections differ from industrial practice of inspections because inspection meetings occurred simultaneously in big rooms, and did not include the document's author.

All these threats are inherent to running classroom experiments and can only be overcome by conducting replications with people, products, and processes from an industrial context.

5. Conclusions

In this paper we have investigated the contribution of meetings in software inspections. We have considered only the main expected benefit of inspection meetings, that is increasing the number of defects discovered with respect to merging the individual preparation logs. Although inspection meetings have other benefits, it is the improvement in defect discovery that usually justifies the meeting costs.

We tested the effectiveness of inspection meetings in two runs of a controlled experiment in a classroom setting, where we compared real teams vs. nominal teams. While a real team reports defects during a face-to-face meeting, defects are attributed to a nominal team by merging the preparation logs of the team individuals. Our finding was that nominal teams were more effective than real teams because meeting losses outperformed meeting gains. We also showed that the meeting duration was not related to team performance.

Previous studies had found no differences between real and nominal teams, and this was our initial hypothesis. Although a null meeting improvement might be considered a sufficient reason to drop out team meetings, in our case the team meetings had a negative effect on defect discovery. The real teams did not produce a substantial amount of group synergism: only 5% of defects were found for the first time during a meeting. Furthermore, real teams erroneously left out more defects than those newly gained.

Our goal was also to provide additional insight into the reasons behind meeting losses.

We tested the differences between defects found by only one reviewer but lost in the meeting, and defects found by more than one reviewer (duplicates) but lost too. We found that most of the meeting losses were not duplicates but defects found by just one reviewer. Perhaps, reviewers who were responsible for the discovery were not able to get the consensus of the other team members. This finding poses a new question of whether interactive meetings are the right process component when the reviewers in a team have separate and distinct detection responsibilities, such as in Scenario-based reading techniques [1, 2, 16].

We also tested the differences between defects found by only one reviewer and collected in the meeting, and defects found by more than one reviewer (duplicates) and collected too. We got contradictory findings between the two experimental runs. With the first document, most of the defects collected during the meetings were duplicates but with the second document, there were no significant differences between duplicates and unique defects with respect to being collected. Then, we are not able to conclude that interactive teams more easily accepted duplicates.

We are conscious that these findings originate from a classroom experiment with inherent threats to the external validity. However, they provide a set of hypotheses to be confirmed or rejected by conducting replications with people, products, and processes from an industrial context.

As future work, we intend to assess the contribution of systematic reading techniques to defect discovery and the effects of combining different or identical perspectives at inspection meetings.

Acknowledgments

We gratefully acknowledge the collaboration of Nicola Barile in the execution and data collection phases of the experiment. Our thanks to all the students of SE class for their hard work. Thanks also to Forrest Shull for having set up the lab package from which the experimental material has been taken.

References

[1] V. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sorumgard, and M. Zelkowitz, “The Empirical Investigation of Perspective-based Reading”, *Empirical Software Engineering*, 1, 133–164, 1996.

[2] V. R. Basili, “Evolving and packaging reading technologies”, *Journal of Systems and Software*, 38 (1): 3-12, July 1997.

[3] V. R. Basili, F. Shull, and F. Lanubile, “Building Knowledge through Families of Experiments”, *IEEE Transactions on Software Engineering*, 25(4):456–473, July/August 1999.

[4] M. Ciolkowksi, C. Differding, O. Laitenberger, and J. Munch, “Empirical Investigation of Perspective-based Reading: A Replicated Experiment”, ISERN Report 97-13, 1997

[5] M. E. Fagan, “Design and Code Inspections to Reduce Errors in Program Development”, *IBM Systems Journal*, 15(3):182–211, 1976.

[6] M. E. Fagan, “Advances in Software Inspections”, *IEEE Transactions on Software Engineering*, 12(7):744–751, July 1986.

[7] P. Fusaro, F. Lanubile, and G. Visaggio, “A Replicated Experiment to Assess Requirements Inspection Techniques”, *Empirical Software Engineering*, 2, 39–57, 1997.

[8] T. Gilb and D. Graham, *Software Inspection*, Addison-Wesley Publishing Company, 1993.

[9] W. S. Humphrey, *Managing the Software Process*, Addison-Wesley Publishing Company, 1989.

[10] P. M. Johnson, and D. Tjahjono, “Does Every Inspection Really Need a Meeting?”, *Empirical Software Engineering*, 3, 9-35, 1998.

[11] L. P. W. Land, R. Jeffery, and C. Sauer, “Validating the Defect Detection Performance Advantage of Group Designs for Software Reviews: Report of a Replicated Experiment”, Caesar Technical Report 97/2, Univ. of New South Wales, 1997.

[12] F. Lanubile, F. Shull, and V. Basili, “Experimenting with Error Abstraction in Requirements Documents”, in *Proc. of METRICS ’98*, 1998.

[13] L. P. W. Lau, C. Sauer, and R. Jeffery, “Validating the Defect Detection Performance Advantage of Group Designs for Software Reviews: Report of a Laboratory Experiment Using Program Code”, Caesar Technical Report 96/8, Univ. of New South Wales, 1996.

[14] J. Miller, M. Wood, and M. Roper, “Further Experiences with Scenarios and Checklists”, *Empirical Software Engineering*, 3, 37–64, 1998.

[15] D. L. Parnas and D. M. Weiss, “Active Design Reviews: Principles and Practice”, *Journal of Systems and Software*,7:259–265, 1987.

[16] A. Porter, L. G. Votta, and V. R. Basili, “Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment”, *IEEE Transactions on Software Engineering*, 21(6):563–575, June 1995.

[17] A. Porter, and L. Votta, “Comparing Detection Methods for Software Requirements Specification: A Replication Using Professional Subjects”, *Empirical Software Engineering*, 3, 355-379, 1998.

[18] A. Porter, H. Siy, A. Mockus, and L. Votta, “Understanding the Sources of Variation in Software Inspections”, *ACM Transactions on Software Engineering and Methodology*, 7(1): 41-79, January 1998.

[19] C. Sauer, D. R Jeffery, L. Land, and P. Yetton, “The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research”, *IEEE Transactions on Software Engineering*, 26(1):1–14, January 2000.

[20] F. Shull, “Procedural Techniques for Perspective-Based Reading and Error Abstraction”, http://www.cs.umd.edu/projects/SoftEng/ESEG/manual/error_abstraction/manual.html, 1998.

[21] L. G. Votta, “Does Every Inspection Need a Meeting?”, *ACM Software Engineering Notes*, 18(5):107–114, December 1993.

[22] D. A. Wheeler, B. Brykczynski, and R. N. Meeson, Jr. (Eds.), *Software Inspection: An Industry Best Practice*, IEEE Computer Society Press, 1996.

[23] B. J. Winer, D. R. Brown, K. M. Michels, *Statistical Principles in Experimental Design*, third edition, McGraw-Hill, New York, 1991.