Empirical Software Engineering, 2, 39–57 (1997)

A Replicated Experiment to Assess Requirements Inspection Techniques

PIERFRANCESCO FUSARO
Fraunhofer Institute for Experimental Software Engineering (IESE)

FILIPPO LANUBILE*
University of Maryland, College Park, USA

GIUSEPPE VISAGGIO
University of Bari, Italy

Abstract. This paper presents the independent replication of a controlled experiment which compared three defect detection techniques (Ad Hoc, Checklist, and Defect-based Scenario) for software requirements inspections, and evaluated the benefits of collection meetings after individual reviews. The results of our replication were partially different from those of the original experiment. Unlike the original experiment, we did not find any empirical evidence of better performance when using scenarios. To explain these negative findings we provide a list of hypotheses. On the other hand, the replication confirmed one result of the original experiment: the defect detection rate is not improved by collection meetings.

The independent replication was made possible by the existence of an experimental kit provided by the original investigators. We discuss what difficulties we encountered in applying the package to our environment, as a result of different cultures and skills. Using our results, experience and suggestions, other researchers will be able to improve the original experimental design before attempting further replications.

Keywords: replicated experiments, technique and process evaluation, inspection, reading technique

1. Introduction

Software inspection is the best known way of detecting defects in software requirements specifications (SRS). Initially created for source code (Fagan, 1976), software inspections have been extended for the intermediate products of the earlier phases of the software life cycle, such as design and requirements specifications (Humphrey, 1989).

The full software inspection process is made up of three consecutive steps: preparation, meeting, and post-inspection activities (Humphrey, 1989). In the preparation step, each member of the inspection team familiarizes with the material and individually reviews the product to find defects. In the meeting, the members of the team discuss the defects from the individual reviews, briefly review the product to find other defects, and prepare a team defect report. In the post-inspection activities, the author of the product fixes the defects contained in the team defect report and the corrections are then reviewed either in new inspection or just together with the moderator. This research addresses issues related to the preparation and meeting steps only.

* Filippo Lanubile is on sabbatical from the University of Bari, Italy

1.1. Detection Techniques

Ad Hoc and Checklist are two popular defect detection techniques which can be applied to the first step of the software inspection process, i.e., the individual detection of defects (Porter, 1995). With the Ad Hoc technique, no guidance is provided during the inspection, but the reviewers use their own knowledge and experience to identify defects in the document. With the Checklist technique, reviewers must answer a list of questions which capture the knowledge from previous inspections. Questions can be general or specific for some classes of defects. The purpose of the questions is to provide some direction to reviewers in looking for defects. Both the Ad Hoc and Checklist approaches are nonsystematic, i.e. there is not a defined and clear procedure to follow. However, on an ordinal scale, Checklist can be considered more systematic than Ad Hoc. Another dimension to characterize these approaches is the reviewer's responsibility; with both Ad Hoc and Checklist approaches, reviewers have general (i.e. they look for all kinds of defects) and identical (i.e. no division of work within a team) responsibilities. Although in a different context, such as design reviews, Parnas and Weiss (Parnas, 1985) criticized the overlapping of responsibilities and proposed active design reviews, a new form of inspection process where the reviewers had specific responsibilities defined by their use of different checklists.

Based on the same concern about the separation of responsibilities, the Scenario approach has been proposed (Porter, 1995) as a systematic defect detection technique with a specific and distinct responsibility for each reviewer. Defect based scenarios are a set of procedures for detecting particular classes of defects. A defect-based scenario requires the reviewer to create a model for a specific class of defects (e.g. missing functionality) and then answer a list of model-based questions. Each reviewer follows a scenario for a specific defect class (i.e., the target defect class) and a team combines distinct scenarios.

Running a controlled experiment, Porter, Votta, and Basili (Porter, 1995) found that:

- the defect detection rate increased about 35% using a Scenario approach as compared to Ad Hoc and Checklist;
- Ad Hoc and Checklist techniques were equivalent with respect to the performance of reviewers;
- reviewers using scenarios found more defects in their target defect class as compared to reviewers using Ad Hoc or Checklist;
- reviewers using scenarios were no less effective at finding defects in other classes than reviewers using Ad Hoc or Checklist;

1.2. Inspection Meetings

As regards the second step of the inspection process, a common assumption is that meetings allow the reviewers to detect more defects than individual reading only. However, an unexpected result from the experiment of Porter, Votta, and Basili put in doubt the contribution of collection meetings as regards the defect detection effectiveness. They found that the

number of new defects discovered during a collection meeting (meeting gain) is equivalent to the number of true defects detected by individual inspection but not included in the team defect report compiled during the meeting (meeting loss).

1.3. Motivation

We found these experimental results, and their implications on the inspection process, very interesting. However, since it is not possible to draw final conclusions from a single experiment, we conducted a replication of the experiment of Porter, Votta, and Basili.

A comprehensive definition of replication is in (Judd, 1991):

> "Replication means that other researchers in other settings with different samples attempt to reproduce the research as closely as possible. If the results of the replication are consistent with the original research, we have increased confidence in the hypothesis that the original study supported."

Software engineering, as a scientific discipline, needs research whose primary purpose is replication. Such research is especially concerned with external validity, i.e. the extent to which we can generalize the results to the population of interest in the hypothesis. Frequently, in software engineering research we are not able, or it is not practical, to use random samples from a population in order to increase our ability to generalize. Generalization must then be done by running multiple experiments in different settings and times. However, replications conducted by the same researchers are not sufficient because the empirical observations in support of a hypothesis may be in error or biased by the original researchers. A scientific hypothesis gains increasing acceptance when independent replications conducted by different researchers arrive at the same conclusions.

Our strict replication of the Porter, Votta, and Basili experiment was made possible by the availability of the experimental material, prepared by the original experimenters in the form of a laboratory package.

2. The Replication Study

Our main interest was in replicating the original research as closely as possible. We decided to make only those minimal changes that were necessary to adapt the first experiment to our environment or to get additional useful information. In the following we describe the design, preparation and execution of the experiment, including the differences between our replication and the original experiment.

2.1. Experimental Design

To evaluate these hypotheses we replicated the controlled experiment. We ran our experiment in early 1995.

The replication of the experiment manipulates five independent variables:

1. the detection technique: Ad Hoc (AH), Checklist (CH), or Scenario (SC) are used during the individual inspection;

2. the inspection round: two inspections (R1, R2) are performed by each reviewer;

3. the specification: two SRS (CRUISE, WLMS) are inspected by each reviewer;

4. the order of inspection: first CRUISE (CW) or first WLMS (WC) can be inspected;

5. the team composition: ten 3-people teams perform the inspection tasks.

The detection technique is the treatment variable. The other variables are used to assess potential threats to the internal validity of the experiment. The original experiment included two separate, internal replications. Thus, the original design used the experimental replication as another independent variable. Since our replication was performed once, we do not include this independent variable.

There are five dependent variables:

1. the team defect detection rate, i.e. the number of defects detected by a team divided by the total number of defects known to be in the specification;

2. the individual defect detection rate, i.e. the number of defects detected by individuals divided by the total number of defects known to be in the specification;

3. the meeting gain rate, i.e. the percentage of defects first identified in the meeting;

4. the meeting loss rate, i.e. the percentage of defects first identified by an individual but not included in the report from the meeting;

5. the time spent by each participant for the individual inspections.

This last measure was collected but not analyzed by the original experimenters, and thus it may be considered an additional dependent variable.

Table 1 shows the organization of both the original experiment (with two internal replications) and our independent replication. The experimental plan is a partial factorial design in which each team inspects two specifications, one per inspection round, using one detection technique, but not all the combinations of the treatment levels are present.

In the first inspection round of the first internal replication of the original experiment, all the teams used the Ad Hoc technique. In the second internal replication, the teams that used the Scenario technique in the first inspection round (2C, 2F, 2H) were constrained to use it again in the second round. This was based on the assumption that the use of a systematic technique by a reviewer could affect future performance using nonsystematic techniques. However, this concern about the possibility of a "carryover effect" can be generalized so that the inspection order conforms to an increasing scale of prescriptiveness. Since the

Table 1. Experimental Plan. Each replication consists of two inspection rounds. In each round the subjects perform first an individual review and then an inspection meeting. Teams from the first internal replication are denoted as 1A–1H; teams from the second internal replication are denoted 2A–2H. Teams from our replication are in bold and denoted A–K.

Detection Technique	Round 1		Round 2	
	WLMS	**CRUISE**	**WLMS**	**CRUISE**
Ad hoc	1B, 1D, 1G, 1H, 2A **A, K**	1A, 1C, 1E, 1F, 2D **D, J**	1A **J**	1D, 2B **K**
Checklist	2B **B**	2E, 2G **E, G**	1E, 2D, 2G **D, G**	1B, 1H **B**
Scenarios	2C, 2F **C, F**	2H **H**	1F, 1C, 2E, 2H **E, H**	1G, 2A, 2C, 2F **A, C, F**

Checklist approach is more systematic than the Ad Hoc approach, we slightly modified the original plan by extending the constraint to Checklist too. As a result, unlike team 2B in the original plan, our team B was not permitted to use the Ad Hoc technique in the second round.

2.2. *Participants*

Our subjects were third and fourth year undergraduates taking an advanced course in software engineering. One sixth of the subjects had at least two years of industrial experience. All of them had experience from a previous software engineering course in SRS reading, but only in the information systems domain. Our students were certainly less experienced than those of the original experiment, who were graduate students, more than half of them with more than two years of industrial experience. However, the subjects of the original experiment cannot be considered software professionals either.

We used 10 teams of three subjects for a total of 30 participants, while the original experiment used 8 teams of three subjects for each of the two replications, for a total of 48 participants. Based on a survey regarding background and experience, participants were ranked as low, medium, and high, and then each team was composed by randomly taking a subject from each of the three categories. The reasons for this choice are summarized in the following:

1. In the two replications of the original experiment, there were two different selection mechanisms to create inspection teams. The former created teams by randomly selecting subjects from three different classes of background-experience (low, medium, high); the latter created teams by pure randomization. Since we performed just one replication, we had to choose just one of these two options.

2. Our independent replication was part of an academic course and thus could not depart from our teaching goals. One of these was to create teams of mixed experience levels

so that less experienced students could learn from more experienced ones. The first option was consistent with this goal.

3. Both Judd (1991) and Campbell (1966) recognize that pure randomization works well on average, given a large enough number of subjects. Having only 30 subjects, we had a reasonable doubt that pure randomization could assure the initial equivalence of the 10 teams. Thus, we decided to exercise some degree of control in forming teams. However, we used matching as an adjunct to randomization and not as a substitute for it. This practice, known more generally as blocking, is recognized as effective by Campbell and Stanley.

2.3. *Experimental Material*

All the material for this replication was present in the experimental kit. However, we needed to translate all the material from English to Italian, otherwise we could have introduced a threat to internal validity represented by the difficulty or slowness in understanding the documents.

The material for the experiment included three small SRSs, support documents for the detection techniques, and data collection forms.

Each of the three SRSs described an embedded real-time system: a home temperature control system (HTCS, 12 pages), a water level monitoring system (WLMS, 24 pages), and an automobile cruise control system (CRUISE, 31 pages). The HTCS SRS was used for training in a trial inspection (the original experiment used an elevator control system for training purposes, but the HTCS SRS was included in place of it in the experimental kit). All the three SRSs adhered to the IEEE format (IEEE, 1984), with an overview written in natural language and the detailed sections specified using the SCR tabular notation (Heninger, 1980). A complete list of the defects for WLMS and CRUISE, but not for the training SRS, was included in the experimental kit. Defects appeared both in the overview and detailed sections of the SRSs.

As a guide for detecting defects, all Ad Hoc reviewers used the same general defect taxonomy. All Checklist reviewers used the same checklist, refined from the taxonomy by adding detailed questions. Scenario reviewers used three different scenarios, derived by refining separate sections from the checklist: data type inconsistencies (DT), incorrect functionalities (IF), and missing or ambiguous functionalities (MF). The experimental kit did not include a classification of the defect list according to the general taxonomy. We classified the defects ourselves but, in order to be sure we were using the same criteria, we also requested the classification directly from the original investigators. Our classification of the defects was identical to the original. Although the classification was a time-consuming activity, this experience can testify to the independence of the defect taxonomy from subjective evaluations. See the appendices in Porter (1995) for the taxonomy, the checklist and the three scenarios.

The defect forms, to be filled out by the subjects during the inspections, included various identifiers (SRS, reviewer, and team), the date, the initial and finish time, the kind of activity (detection or collection), the defect location (page and line number), and a textual

description of the defect. An entry was also provided to record the defect disposition (true or false positive). However, only true defects were analyzed in the original experiment and we did the same.

2.4. *Preparation and Training*

We gave five 2-hour lectures on the IEEE standard for SRS, the SCR tabular notation, the inspection process, the defect taxonomy and checklist, and the data collection forms. Lecture references (IEEE, 1984; Heninger, 1980; Fagan, 1976) were found in the experimental kit. However, the Heninger's style for SCR tabular notation was partially different from that used in the experimental kit. Thus, we adapted our lectures to the style effectively used for the three SRSs.

After these lectures, we created teams and randomly assigned teams to detection techniques, and subjects to team roles (moderator, recorder, and reader). A trial inspection was performed as a simulation of an experimental session. All the subjects inspected the HTCS SRS using the Ad Hoc technique, as performed in the original experiment and suggested by the guide in the experimental kit. After a two-hour individual inspection and a two-hour collection meeting, another lecture was given both to get feedback from the participants and to answer their questions.

Afterwards, we gave a lecture on defect-based scenarios but only for those participants who had to use the Scenario technique in the first inspection round. The lecture described the scenarios and showed how they could be applied to the training SRS. After the first inspection round, the lecture on scenarios was repeated for those subjects who had to use the Scenario technique for the first time in the second inspection round.

2.5. *Execution*

The two inspection rounds were conducted in a two consecutive days while the original experiment used one week. The participants worked in a big room with enough space so that they would not be disturbed. After they received the experimental package (instructions, SRS, and defect report forms), the students had 30 minutes to read and understand the SRS, and to ask us questions for clarification. Then, they had 2 hours to do the individual inspection and fill out the defect report forms. After they finished they completed a debriefing questionnaire and returned the material to us.

Fifteen minutes after all the subjects had finished their individual inspections, they began the inspection meetings. The meeting process is independent of the defect detection technique used for the individual inspection: the reader calls for defects, sequentially scanning the document; the recorder fills out the team defect report form; and the moderator manages the discussion within the team. The meetings had a two-hour limit. At the end, the team defect report forms were given back to us.

We preferred to shorten the interval period between the two inspection rounds (one day), and the individual-collective inspections (15 minutes) to reduce the threat of internal validity caused by subjects talking to others about their experience.

Table 2. List of new defects.

SRS	Id	Page	Line	Description
WLMS	43	3	22	The Station is not depicted in figure 1 but in figure 2.
WLMS	44	18	8	The input hardware port/clkpulse/ is never used.
WLMS	45	14	9	The update involves //low window// and not //high window// (the same defect is at page 15, 16 where the update involves, respectively, //level display// and //allarm//).
WLMS	46	26		Missing reference to figure 3.
CRUISE	27	4	14	The gas tank belongs to standard hardware which interacts with the software system.
CRUISE	28	9	5	It is not specified whether the system knows the mileage.
CRUISE	29	9	22	The event should be "Event(4750 < !oil miles! < 4950)".
CRUISE	30	16	25	The value is not a non-negative fixed point but a floating-point number in the range [1, 100] as specified on page 25.

3. Experimental Results

Looking at the individual and team defect report forms, we validated the defects detected by the participants during the inspections. We discarded false positives and minor errors, such as stylistic inconsistencies or obviously typographical errors. Our effort was devoted to determining if a defect description, possibly unclear, could be matched to some known defect or could lead to future system faults if remaining uncorrected in the SRS.

While almost one half of the original defects were not discovered, our students found some new defects not present in the list of the experimental kit. The new defects, initially found in the Italian translation of the SRSs, were also present in the original English SRSs. The new defects are shown in table 2.

We did two parallel analyses of the experimental data. The first analysis used only the defects that were present in the experimental kit. The second analysis used both the defects in the original experiment and the new defects discovered for the first time by our students. The results of both the analyses bring to the same conclusions. Here, we will show only the results from the first analysis because more directly comparable to those of the original experiment.

3.1. Team Inspection Performance

Figures 1 and 2 are frequency diagrams showing the number of teams that found each defect respectively in the WLMS and CRUISE SRSs. With respect to the 42 defects of the WLMS SRS, 3 defects were found by all the teams, 13 defects by at least one half of the teams, and 17 defects were not found by any team. For the 26 defects of the CRUISE SRS, 2 defects

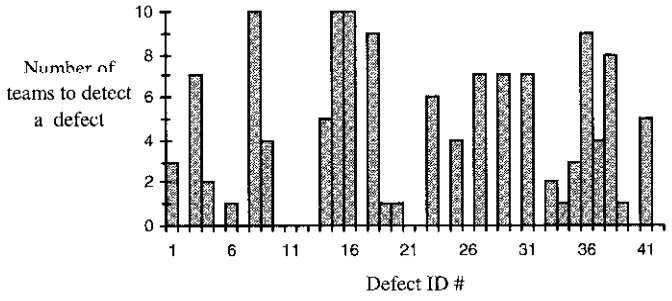


Figure 1. Number of teams that found each WLMS defect during inspection.

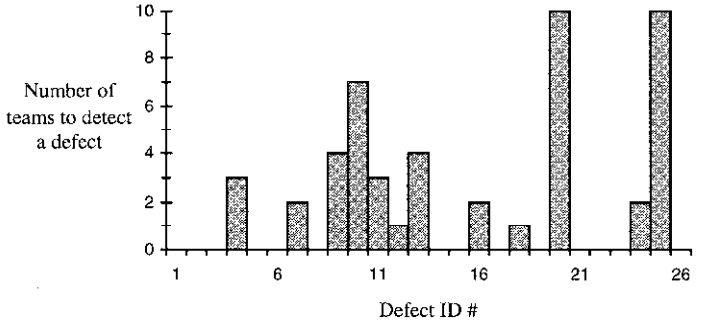


Figure 2. Number of teams that found each CRUISE defect during inspection.

were found by all the teams, just 3 defects by at least one half, and 14 defects were not found by any team.

The data were analyzed by repeating the analysis strategy of the original experiment. First, data were analyzed with a one-way analysis of variance for each of the five independent variables, to identify the individual variables that could explain a significant amount of variation in the team detection rates. For each independent variable, we can formulate the null and alternative hypotheses as follows:

H_0: There is no difference between the various treatment conditions with respect to the team scores on the defect detection rate.

H_a: There is a difference between the various treatment conditions with respect to the team scores on the defect detection rate.

The results, summarized in table 3, revealed a significant effect only for Specification ($p < 0.05$). Detection Technique, Inspection Round, Inspection Order, and Team Composition did not show significant differences. On the other hand, in the original experiment,

Table 3. Analysis of variance for each independent variable. df represents the degrees of freedom, SS the sum of squares, MS the mean square, F the value of the F statistic used to test the null hypothesis, and p the probability of incorrectly rejecting the null hypothesis. A p value of 0.05 is the most commonly accepted threshold, when a p value is less than 0.05, the null hypothesis may be rejected and thus there is a significant effect for the independent variable.

Independent Variable	df	SS	MS	F	p
Detection Technique—main treatment	2	0.00499	0.00250	0.27	0.77
Inspection Round—maturation effect	1	0.00039	0.00039	0.04	0.84
Specification—instrumentation effect	1	0.06475	0.06475	11.93	*0.003*
Inspection Order—presentation effect	1	0.01959	0.01959	2.47	0.13
Team Composition—selection effect	9	0.09158	0.01018	1.44	0.29

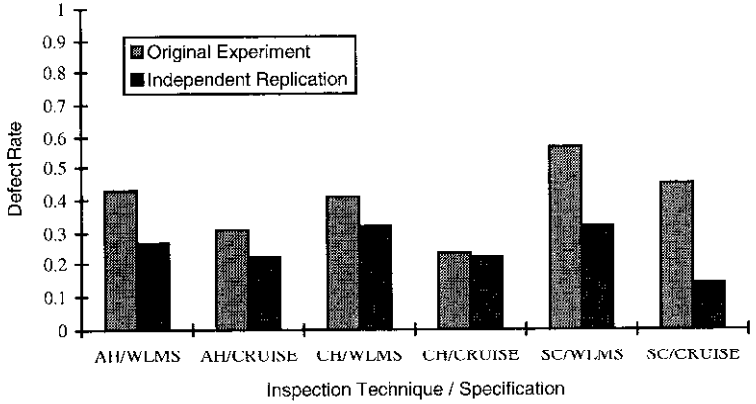


Figure 3. Average team defection detection rates. Each vertical bar is labeled according to the detection technique and the specification. For example, AH-WLMS represents the average defect detection rate obtained with the Ad Hoc technique when inspecting the WLMS specification.

Detection Technique was the most significant independent variable ($p < 0.01$), together with Specification ($p < 0.01$).

Figure 3 shows the average defect detection rates for both the original experiment and the independent replication. Inspection teams in our independent replication generally found less defects than teams of the original replication. While the difference between the defect rates in the two experiments is moderate for teams using the Checklist technique, it is considerable for the Defect-based Scenario technique. Both the lowest and the highest differences are with the CRUISE SRS using, respectively the Checklist and the Scenario techniques.

Figure 4 graphically depicts the variance of the team defect detection rates by showing

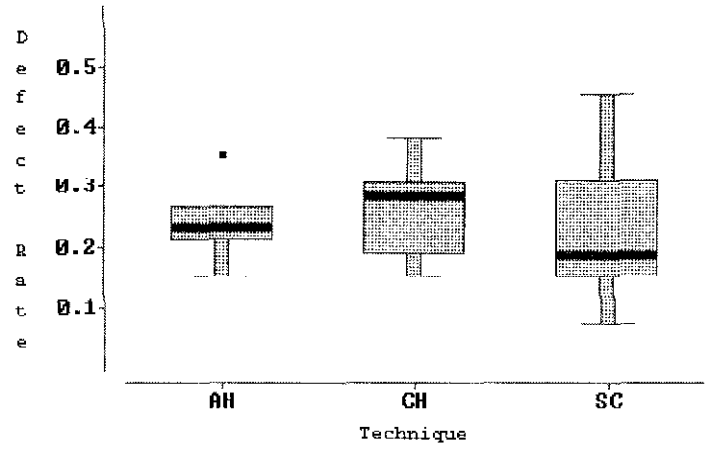


Figure 4. Box plots for the team defect detection rates. The rectangles include the data points in the first to the third quartiles. The median for each method is identified by a bar within the boxes. Vertical lines outside the boxes extend up from the third quartile to the 95th percentile and down from the first quartile to the 5th percentile. Outliers are represented as isolated points.

box-plots for each of the defect detection methods in our replication. Scenario, the most systematic technique, has a higher variance than Ad Hoc and Checklist techniques, with respect to team defect detection rates.

As a second step in the analysis of the team performance, we repeated the analysis of the original experiment to see if the main treatment variable, Detection Technique, varied significantly with the different levels of the Specification variable. Data were analyzed using a two-way analysis of variance. As in the original experiment ($p = 0.81$), the interaction between Detection Technique and Specification was not significant ($p = 0.36$).

3.2. *Individual Inspection Performance*

The analysis of the individual defect report forms helps us to verify if specializing the responsibility of a reviewer results in finding more defects in the related specific class.

Given a specific defect-based scenario X (data type inconsistencies, missing functionality, or incorrect functionality), the null and alternative hypotheses can be formulated as follows:

$H1_0$: Scenario X reviewers do not differ from non-scenario reviewers with respect to the number of X defects found

$H1_a$: Scenario X reviewers find more X defects than non-scenario reviewers

Furthermore, we also see if the scenario reviewers are equally effective at finding defects for which their scenarios were not designed to detect (these defects are labelled as "others"). If not, reviewer specialization could have an undesired effect when scenarios only have a

Table 4. Individual defect detection rates for WLMS SRS. DT is a data type inconsistency, MF a missing functionality, IF an incorrect functionality, "others" are defects of any other class.

Defect Type	Number Present	Defects found by specific SC reviewers	Defects found by non-SC reviewers	p-value
DT	14	6.00	3.75	*0.02**
MF	5	0.50	0.22	0.21*
IF	5	0.50	0.22	0.21*
others	18	1.41	2.33	0.06**

* The null hypothesis is $H1_0$; the alternative hypothesis is $H1_a$
** The null hypothesis is $H2_0$; the alternative hypothesis is $H2_a$

Table 5. Individual defect detection rates for CRUISE SRS.

Defect Type	Number Present	Defects found by specific SC reviewers	Defects found by non-SC reviewers	p-value
DT	10	1.50	1.77	0.35*
MF	3	0.00	0.00	NA
IF	1	0.25	0.16	NA
others	12	0.41	0.66	0.15**

* The null hypothesis is $H1_0$; the alternative hypothesis is $H1_a$
** The null hypothesis is $H2_0$; the alternative hypothesis is $H2_a$

partial coverage of the defect population, like in this experiment (about 50% of the defects covered by the checklist). The null and alternative hypotheses can be formulated as follows:

$H2_0$: Scenario reviewers do not differ from non-scenario reviewers with respect to the number of "other" defects found

$H2_a$: Scenario X reviewers find less "other" defects than non-scenario reviewers

The individual detection rates of Scenario reviewers are compared in tables 4 (WLMS) and 5 (CRUISE) with those of all other reviewers. The right columns in the two tables correspond to the results of one-tail t tests performed to verify the hypotheses.

The results were very different between the WLMS and CRUISE specifications. In the WLMS SRS, the analysis revealed that the number of DT defects found by DT reviewers was significantly higher than the number of DT defects found by non-scenario reviewers ($p < 0.05$). On the other hand, in the CRUISE SRS, the analysis failed to reveal any significant difference for DT defects.

With respect to the second hypothesis, we did not find any significant difference. However, for the WLMS SRS the probability value ($p = 0.06$) is very close to the common accepted threshold of 0.05 for rejecting a null hypothesis.

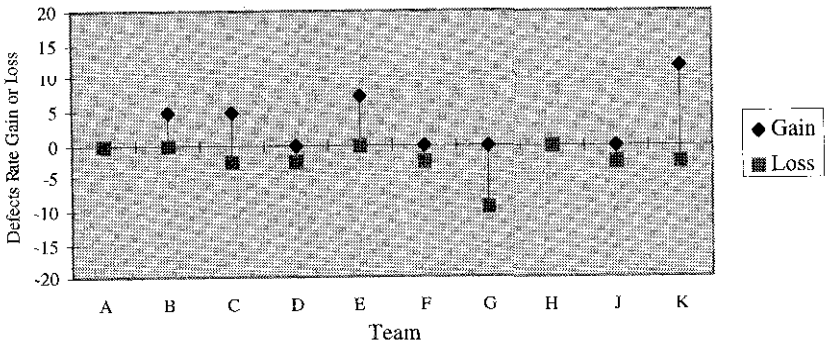


Figure 5. Defect gain and loss rate for WLMS. Each diamond symbol represents the meeting gain rate for a single inspection, i.e., the percentage of defects found in the collection meeting for the first time. Each square symbol represents the meeting loss rate for a single inspection, i.e., the percentage of defects first identified by an individual reviewer but lost during the collection meeting.

For both WLMS and CRUISE, the number of MF and IF defects found by MF and IF reviewers, respectively, was slightly higher than the number of MF and IF defects found by non-scenario reviewers. However, we did not find significant differences for the WLMS SRS, and we could not perform any statistical analysis for the CRUISE SRS, since there were too few data points.

3.3. *Meeting Performance*

We tested the defect detection effectiveness of meetings by comparing the meeting gain rates (i.e. the percentages of defects first identified at the meeting) and the meeting loss rates (i.e. the percentage of defects first identified by an individual but not included in the report from the meeting). We formulated the null and alternative hypotheses as follows:

H_0: There is no difference between gain rate and loss rate within a meeting

H_a: There is a difference between gain rate and loss rate within a meeting

Results were analyzed using a paired-samples t test. The analysis failed to reveal a significant difference between the meeting gain rates and the meeting loss rates ($p = 0.78$ for WLMS and $p = 0.82$ for CRUISE), confirming the results of the original experiment.

Figures 5 and 6 show the meeting gain and loss rates for WLMS and CRUISE, respectively.

3.4. *Time Performance*

The time limit given to the subjects was two hours as in the original experiment. Table 6 shows a quartile table of the time spent in individual reviews. The second column provides

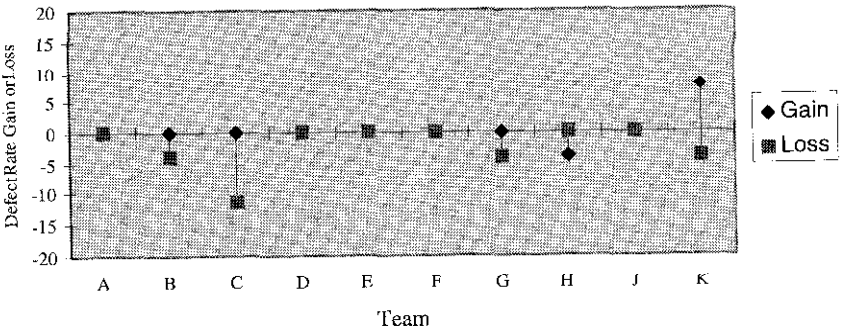


Figure 6. Defect gain and loss rate for CRUISE. Each diamond symbol represents the meeting gain rate for a single inspection, i.e., the percentage of defects found in the collection meeting for the first time. Each square symbol represents the meeting loss rate for a single inspection, i.e., the percentage of defects first identified by an individual reviewer but lost during the collection meeting.

Table 6. Quartiles for time spent in individual reviews. The second column provides the percentage of observations which are below the value shown in the third column (in minutes) of the same row.

Maximum	100%	120
3rd quartile	75%	120
Median	50%	120
1st quartile	25%	115
Minimum	0%	70

the percentage of observations which are below the value shown in the third column (in minutes) of the same row. Most of our students finished their individual reviews right at the deadline (more than 50% after exactly 120 minutes and only 25% under 115 minutes). We do not know if the same occurred in the original experiment because time was not included among the dependent variables.

3.5. *Debriefing Questionnaire and Interviews*

After the experiment the participants completed a debriefing questionnaire built with the purpose of helping us in interpreting the quantitative results. The debriefing questionnaire was validated by means of informal interviews performed by a senior student who helped us in running the experiment. Having a colleague as a direct interface enabled the students to

feel comfortable in answering our questions. The main results from the qualitative analysis are the following.

- *Participants felt they had received too little training in the Scenario approach;* some students stated that they really understood how to use a scenario only after the first inspection round. Remember that, as in the original experiment, students performed a training inspection only with the Ad Hoc approach.

- Students complained about the strict time limit; many students declared that they could have done a better job with more time available.

- Participants suffered from an unfamiliar domain, like embedded real-time systems; during their courses, they had been trained mainly on applications in the information systems domain which focuses more on data management and user requests. Furthermore, in Italy there are no cars with embedded cruise systems.

- Participants were not able to develop the invariants required by the IF scenario: the state-event model in the training SRS was too simple compared to those in the WLMS and CRUISE SRS.

4. Discussion

With respect to the original experiment, our independent replication confirms only that meeting gains are offset by meeting losses during collective inspections. We cannot confirm the results from the assessment of the defect detection techniques. Controlled experiments work well in establishing a cause-effect relationship when they get positive findings, that is when the null hypothesis is rejected. When there are negative findings, as in this case where no significant differences are found among detection techniques, only hypotheses can be drawn. In the following we give a list of hypotheses which can explain why the Scenario approach did not increase the overall effectiveness of the inspection process. They can be considered lessons learned that must be carefully considered for future replications of the experiment.

- *Subjects were asked to learn too many new things.* Our students were not familiar with real-time embedded systems neither had they ever driven a car with cruise control. They were not used to modeling control; they learned the SCR tabular notation, and they performed a formal inspection for the first time. From a teacher's point of view, practice with so many issues in a single course can be considered a good result. However, from an experimenter's point of view, uninterpretable results can come from underestimating the subject's learning curve.

- *Defects in the introductory parts create confusion.* Our subjects needed to first understand the problem but they had no error-free introduction. When subjects are not expert of the problem domain, the seeded defects should be contained only in the detailed specifications, leaving the general descriptions free of errors.

- *Training was unfair.* The trial inspection was conducted only with the Ad Hoc approach, while participants could not get confidence with the Checklist and Scenario approaches before the experimental trials. Participants who had the Scenario technique in both inspection rounds said they understood their scenarios only after the first inspection. We cannot verify these affirmations by testing for significant differences in performance, because there are too few observations. However, this can be considered as a threat to the internal validity of the experiment.

- *Subjects who had trouble with the Scenario approach used different techniques to execute the task.* Our students had some difficulties in following the scenarios, especially with the incorrect functionality scenario which required writing invariants from the event and condition tables. A possible interpretation of the greater variance with the Scenario approach could be that some students were successful in applying the scenario, some turned back to Ad Hoc or Checklist, and some continued until the end to apply the assigned scenario but with negative results. The real process should be recorded to prove its conformance to the ideal process.

- *Time limit was too short.* Analyzing human performance in programming activities, Weinberg and Schulman concluded that unreasonably short deadlines would result in erroneous programs, and warned experimenters against mixing the results of subjects who have easily finished with those of subjects who were pressed for time (Weinberg, 1974). We do not know if this happened to the original experiment. In our case, if we discard all the data from subjects pressed for the deadline there will not be enough observations to analyze.

5. Summary and Conclusions

The results of our independent replication differed from the original experiment with respect to the comparison of the defect detection techniques, while the results were identical with respect to the evaluation of the defect detection effectiveness of the collection meetings. The following may be considered a summary of the results from our study:

1. The team defect detection rate when using the Scenario technique was not significantly different from those obtained with Ad Hoc or Checklist techniques. The average defect detection rate with Scenario was slightly higher than Ad Hoc and Checklist but only with one specification (WLMS). On the contrary, in the original experiment the defect detection rate when using Scenario was significantly superior (about 35%) to that obtained with Ad Hoc or Checklist techniques. Both the original experiment and our replication found that there were significant differences between specifications (CRUISE more difficult than WLMS) and that there was no significant interaction between detection techniques and specification.

2. During individual inspections of WLMS, the DT scenario was more effective than Ad Hoc and Checklist at finding its specific class of defects. However this is not true for the individual inspection of CRUISE, and we have no evidence for the other two scenarios.

On the other hand, in the original experiment all the three scenarios helped to find more scenario-specific defects than Ad Hoc or Checklist. Furthermore, for the WLMS specification, the Scenario technique showed a lower ability in finding non-targeted defects than Ad Hoc and Checklist, while for the CRUISE there were no significant differences. In the original experiment, the Scenario technique was as effective in finding other classes of defects as Ad Hoc and Checklist.

3. Both in the original experiment and in our replication, defects found for the first time during the inspection meeting were offset by true defects lost in the meeting.

We discovered problems, some related to the change in experimental environment but others related to the experiment itself, that could be considered threats to internal validity. These include learning curve, unfair training, task conformance, and time limit. Although we are not able to explain exactly why we got partially different results, we have offered some reasonable hypotheses and suggest improvements to enable other researchers to attempt new independent replications. Further replications are needed to understand better under which conditions scenario-based reading is effective.

In their survey on empirical research, Campbell and Stanley wrote (Campbell, 1966):

> "*...we must increase our time perspective, and recognize that continuous, multiple experimentation is more typical of science than once-and-for-all definitive experiments. The experiments we do today, if successful, will need replication and cross-validation at other times under other conditions before they can become an established part of science, before they can be theoretically interpreted with confidence*".

Acknowledgments

Our thanks to all the students participating in the experiment for their hard work. We would like to thank Vic Basili, Adam Porter and Larry Votta for their useful technical comments, and all ISERN (International Software Engineering Research Network) members for having encouraged this replication. Thanks also to John Daly, Carolyn Seaman, Forrest Shull, Sivert Sorumgard, and the anonymous reviewers for having improved a draft version of this paper.

This work has been partially supported by the "40%" funds of the Italian M.U.R.S.T. under the project "V&V in software engineering".

References

Campbell, D. T., and Stanley, J. C. 1966. *Experimental and Quasi-Experimental Designs for Research.* Boston: Houghton Mifflin Co.

Fagan, M. E. 1976. Design and code inspections to reduce errors in program development. *IBM Systems Journal* 15(3): 182–211.

Heninger, K. L. 1980. Specifying software requirements for complex systems: new techniques and their application. *IEEE Trans. Soft. Eng.* SE-6(1): 2–13.

Humphrey, W. S. 1989. *Managing the Software Process*. New York: Addison-Wesley.

IEEE Std.830-1984. *IEEE Guide to Software Requirements Specification*. Soft. Eng. Tech. Comm. of the IEEE Computer Society.

Judd, C. M., Smith, E. R., and Kidder, L. H. 1991. *Research Methods in Social Relations*, 6th edition. Orlando: Holt Rinehart and Winston, Inc.

Parnas, D. L., and Weiss, D. M. 1985. Active design reviews: Principles and practices. *Proc. 8th Int. Conf. Soft. Eng.* 215–222.

Porter, A. A., Votta, L. G., and Basili, V. R. 1995. Comparing detection methods for software requirements inspections: A replicated experiment. *IEEE Trans. Soft. Eng.* 21(6): 563–575.

Weinberg, G. M., and Schulman, E. L. 1974. Goals and performance in computer programming. *Human Factors* 16(1): 70–77.

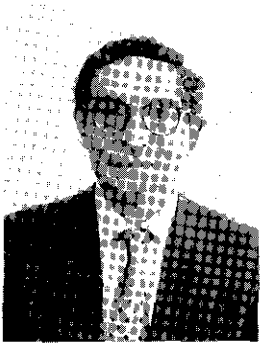
Pierfrancesco Fusaro received his Laurea degree in Computer Science (cum laude) from the University of Bari, Italy, in 1995.

He is currently a researcher of the Quality and Process Engineering Department (Quantitative Methods group) at the Fraunhofer Institute for Experimental Software Engineering (FhG IESE), an industry-oriented research centre located in Kaiserslautern, Germany.

His current research interests include: goal-oriented measurement based on the Goal Question Metric approach, experimentation, evaluation and improvement of the reliability and validity of SPICE-based assessments.

He is also a joint researcher of the Software Engineering Research Laboratory of the Computer Science Department, Univeristy of Bari.

Filippo Lanubile is a research associate of computer science at the University of Maryland, College Park. He is currently in sabbatical from the University of Bari, Italy, where he is an assistant professor of computer science. His research interests include experimental software engineering, reading techniques, validation of software measures and prediction models, evaluation and reuse of software architectures, reengineering, and program slicing. He received a Laurea degree in computer science from the University of Bari. He is a member of the ACM and IEEE Computer Society.



Guiseppe Visaggio is professor in Informatic Department at the Univeristy of Bari. He received the degree in physics from University of Bari. His research interests are quality improvement, measurement of software processes and resulting products, process-sensitive software development environments, reverse and reengineering of existing software. He has published more than 60 papers. He is the head of Software Engineering Research Laboratory (SER-Lab). SER-Lab hosts several basic research projects and executes experiments controlled and in field. He has served for many years as a member of the program committee of I.E.E.E. International Conference on Software Maintenance (ICSM), Workshop on Program Comprehension and Workshop on Empirical Studies of Software Maintenance. He will serve in 1997 as program chair of th ICSM. He is member of the I.E.E.E. Computer Society, A.C.M. and A.I.C.A. (the Italian Computer Society).