Empir Software Eng (2014) 19:241–266
DOI 10.1007/s10664-012-9221-0

Investigation of individual factors impacting the effectiveness of requirements inspections: a replicated experiment

Özlem Albayrak · Jeffrey C. Carver

Published online: 4 August 2012

Editor: Filippo Lanubile

Abstract This paper presents a replication of an empirical study regarding the impact of individual factors on the effectiveness of requirements inspections. Experimental replications are important for verifying results and investigating the generality of empirical studies. We utilized the lab package and procedures from the original study, with some changes and additions, to conduct the replication with 69 professional developers in three different companies in Turkey. In general the results of the replication were consistent with those of the original study. The main result from the original study, which is supported in the replication, was that inspectors whose degree is in a field related to software engineering are less effective during a requirements inspection than inspectors whose degrees are in other fields. In addition, we found that Company, Experience, and English Proficiency impacted inspection effectiveness.

Keywords Software inspections · Software engineering · Empirical studies · Replication · Requirements

1 Introduction

An inspection is the static review of a software artifact to achieve a particular goal, e.g. fault detection. Inspections can be performed on many software artifacts, as opposed to testing which must be performed on executable code. According to IEEE Standard 1028–1997, a software inspection is the visual examination of a software product to detect and identify software anomalies, including errors, violation of development standards, and other

Ö. Albayrak
Department of Computer Technology and Information Systems, Bilkent University, Ankara, Turkey
e-mail: ozlemal@bilkent.edu.tr

J. C. Carver (✉)
Department of Computer Science, University of Alabama, Tuscaloosa, AL, USA
e-mail: carver@cs.ua.edu

problems. Since their introduction in the mid-70s at IBM (Fagan 1976), inspections have been performed on artifacts from all phases of the software lifecycle, including requirements (Basili et al. 1996; Carver et al. 2006; Martin and Tsai 1990; Schneider et al. 1992; Aceituna et al. 2011), architecture (Carver and Lemon 2005), design (Laitenberger et al. 2000; Parnas and Weiss 1985; Travassos et al. 1999a, b), code (Laitenberger and DeBaud 1997) and user interfaces (Zhang et al. 1999). Researchers who have widely studied software inspections agree that they are an effective method of identifying faults and improving software quality (Aurum et al. 2002; Fagan 1986; Kollanus and Koskinen 2009). One researcher has even developed a maturity model for software inspections (Kollanus 2009, 2011). In other cases, faculty members are including requirements inspections as part of their software engineering courses (e.g. (Garousi 2010)).

The traditional software inspection process consists of four steps. First, the inspection team leader forms a team of inspectors. Next, those inspectors individually review an artifact to prepare for the third step, the team meeting. During the team meeting, the members of the inspection team discuss the artifact and identify faults. Finally, the inspection team provides the document author with the list of faults for repair. The original Fagan inspection method places the fault detection emphasis on the team meeting. Researchers who question whether a formal team meeting is needed or beneficial in all cases have investigated the impact of emphasizing fault detection during the individual preparation step rather than during the team meeting (Johnson and Tjahjono 1997; Laitenberger et al. 2001; Votta 1993; McCarthy et al. 1996; Olalekan and Adenike 2008). In this type of inspection, the inspectors spend their preparation time identifying as many faults as possible. Then, the team meeting, if one occurs, focuses on ensuring that the individually identified faults are correct and on identifying additional faults missed by the individual inspectors.

Regardless of whether fault detection occurs primarily during team meetings or primarily during individual preparation, the fault detection activity is an individual activity (Laitenberger et al. 2001). Therefore, the overall effectiveness of an inspection team depends largely on the effectiveness of the individual team members. But, even when different inspectors use the same inspection technique, their effectiveness is quite varied (Carver et al. 2006, 2008; Carver 2003, 2004; McMeekin et al. 2009; Winkler et al. 2007). Other studies suggest that in addition to the specific inspection process followed, technical and non-technical characteristics of individual inspectors may influence individual effectiveness (Land et al. 2003; Sauer et al. 2000). There is a need to better understand the sources of this variation (Aurum et al. 2005; Wong 2011).

Other factors like the choice of inspection technique may also impact inspection effectiveness. Previous work in this area is inconclusive about which techniques are better, e.g. (Ciolkowski 2009; Winkler et al. 2010). An early study by Porter et al. found that scenario-based reading techniques were more effective than either a checklist or an ad hoc approach (Porter et al. 1998). Later studies contradicted this result by showing no benefit for scenario-based techniques (Sandahl et al. 1998; Fusaro et al. 1997). In their review of the impact different scenarios have on inspection effectiveness, Regnell et al. found the results to be inconclusive. Furthermore, their own replication of an earlier study showed that specific perspectives do not seem to affect effectiveness (Regnell et al. 2000). This series of conflicting results again highlights the need to understand how other factors, like individual differences, affect an inspector's performance during a requirements inspection.

While it is clear that the specific inspectors chosen for an inspection team impact the outcome of the inspection, it is not clear which characteristics of those inspectors are important to consider when choosing the best inspectors. Researchers have identified a number of candidate characteristics, including: *development skills*, *quality assurance* *experience*, *performance on a practice inspection* (Biffl 2000), *participant experience level (student vs. professional)* (McMeekin et al. 2009; Wong 2011), and *level of training on inspections* (Dillon et al. 1988). There are no definitive results available about these characteristics. Because inspections are a cognitively intense process (Robbins and Carver 2009), the way an individual processes information could also impact their performance during an inspection (Hungerford et al. 2004).

The remainder of this section first describes a previous study that investigated this question in more detail. Next, it discusses experimental replications in software engineering and motivates the need for the replication described in this paper.

1.1 The Original Study

In 2008, Carver, Nagappan and Page conducted a study to understand the impact of an inspector's educational background on their effectiveness when conducting a requirement inspection (Carver et al. 2008). Their main research question was "Are inspectors who have a degree in computer science more effective during a requirements inspection than inspectors with non-computer science degrees?" To gain insight into the impact of an inspector's background and experience, Carver et al. tested five hypotheses (hypothesis numbers are prefixed with an "O" to distinguish them from the replication hypotheses:

- H_{O1}: The effectiveness of an inspector is affected by their educational background (computer or non-computer).
- H_{O2}: Experience with requirements affects the effectiveness of an inspector.
- H_{O3}: The effectiveness of an inspector is affected by their level of education (bachelor's vs master's degree).
- H_{O4}: The effectiveness of an inspector is affected by whether or not they have industrial software development experience.
- H_{O5}: Inspection experience affects the effectiveness of an inspector.

The dependent variable was **Inspection Effectiveness**, defined as the number of correct faults detected during the inspection. The independent variables, along with their measurement levels in parentheses were:

1) **Educational Background** (*computer related* or *non-computer related*)—the field in which the participant earned their most advanced degree;
2) **Requirements Experience** (*none* or *some*)—experience writing requirements;
3) **Educational Degree** (*bachelor's* or *master's*)—the highest degree earned;
4) **Industrial Experience** (*none* or *some*)—experience in industry; and
5) **Inspection Experience** (*none* or *some*)—experience reviewing requirements or with inspections in general.

The researchers trained 75 Microsoft developers to perform a checklist-based inspection (the typical method used at Microsoft). Those developers had 70 minutes to identify as many faults as possible in the Loan Arranger requirements document (described in Section 2.4). The researchers compared the fault lists submitted by each participant against a master fault list to determine how many of the reported faults were true faults.

The results of this study showed that inspectors with non-computer backgrounds were significantly more effective than inspectors with computer backgrounds (H_{O1}). Further analysis showed that, of the inspectors with computer backgrounds, those who had a computer science or a software engineering background were the least effective. The results also showed that inspectors with requirements experience were significantly more effective than those without requirements experience (H_{O2}). The study did not provide concrete results for H_{O3}–H_{O5}.

As with any empirical study, the researchers identified a number of limitations. First, the fact that all participants were drawn from the same company led to a potential for selection bias and a threat to external validity. Second, the representativeness of the artifact was a threat to external validity. Third, because two of the authors worked for Microsoft, there was a potential that they may have influenced the performance of the participants, resulting in an instrumentation bias.

To better understand the extent of these results and address some of the limitations, we replicated this study in a different industrial context (as described in Section 2).

1.2 Motivation for the Replication

Experimental replication is a key feature of empirical software engineering (Juristo and Vegas 2009) and is receiving increased attention within the empirical software engineering community (Andersson 2007; Kitchenham 2008; Shull et al. 2002, 2004, 2008). Replications performed in a variety of environments are the basis for obtaining robust and generalizable results (Andersson 2007). Due to many uncontrollable sources of variation between different environments, the results of any one study cannot simply be extrapolated to all environments (Shull et al. 2002). Juristo and Vegas point out that variations in experimental conditions between replications have caused many replications to be unsatisfactory because they did not increase the credibility of the original study's results. In addition, they note that a replication tends to produce similar results only when the original researchers also conduct the replication at the same site (Juristo and Vegas 2009).

Software engineering researchers use replications to test the reproducibility of a result and to understand the sources of variability that influence that result. A replication can provide useful insight regardless of whether its results are similar to or different from the results of the original experiment. A replication that produces similar results strengthens the generality of those results. Conversely, a replication that produces contradictory results suggest the presence of additional context variables that need to be identified and studied (Shull et al. 2008).

It is not simple to conduct an effective replication. Thus, to help software engineering experimenters, Shull et al. adapted the concept of replication packages from other experimental disciplines and developed a framework for knowledge sharing between the original experimenters and the replicators (Shull et al. 2004). The goal of this framework is to facilitate communication to increase the chances that the replicators will understand the original experiment and be able to conduct a successful replication.

Our primary motivation for conduct this replication was to address some of the threats to validity present in the original study and to further investigate relevant background factors. Our goal was to conduct a dependent replication by following the procedures of the original experiment as closely as possible (Shull et al. 2008). To support this goal, Carver was involved in the planning and analysis of both studies. In addition, the laboratory package from the original study guided the replication planning. Even so, we introduced some changes during the replication.

First, we made changes to specifically address validity threats from the original experiment:

- To address the selection threat and the external validity threat that resulted from all participants being drawn from the same US-based company, we drew participants from three Turkish-based companies for the replication.

- To address the potential instrumentation bias, the replicating researchers were not affiliated with any of the participating companies.
- To address some construct validity threats (not noted by the original authors) we introduced new definitions for the Educational Background and Experience variables and introduced two new variables: English Proficiency and Artifact Format.

Second, the change of setting allowed us to further investigate the main result from the original study. As Section 2.2 describes, most of the participants in the replication would have been classified as computer related in the original study. So, in the replication we further investigated the impact of a software engineering background rather than a general computing background.

Finally, the change of setting for the replication resulted in additional minor changes:

- Because the participants in the replication were not native English speakers, we gave them 5 extra minutes for the inspection (75 minutes total).
- Some of the training documents used in the original study were proprietary. In the replication we substituted different inspection training materials. Carver helped to ensure that the training materials used in the replication were comparable to those used in the original study.

The remainder of the paper is organized as follows. Section 2 describes the replication. Section 3 provides an analysis of the data. Section 4 compares the results of the replication with those from the original study. Section 5 discusses the implications of the results. Section 6 describes the threats to validity. Section 7 presents contributions and limitations. Finally, Section 8 contains the conclusions and future plans.

2 The Replication

This section provides details about the design and execution of the replication.

2.1 Interaction with Original Experimenters

The level of involvement that the original experimenters should have in a replication is a controversial topic. There is an ongoing debate about the merits and demerits of the involvement of the original experimenters in a replication (Juristo and Vegas 2009; Kitchenham 2008; Shull et al. 2008; Lung et al. 2008). In the spirit of full disclosure, Carver was involved with both the original study and the replication. In the replication, Carver acted as a consultant by helping with the study design and data analysis. The actual execution of both the original study and the replication, including the training and data collection, was carried out by the other authors (Nagappan and Page in the original study and Albayrak in the replication). In addition, for the replication we reused the lab package from the original experiment containing the artifacts and data collection instruments.

2.2 Experimental Design

This section describes the study variables and hypotheses along with a justification for any specific changes made to the original study design.

2.2.1 Variables

The replication has one dependent variable and eight independent variables. The dependent variable is **Inspection Effectiveness (INS)**, defined as the *number of correct faults reported by a participant*. A correct fault is a fault that appears on the list of known faults in the Loan Arranger document (described in Section 2.4).

We reused all of the independent variables from the original study, with slightly modified definitions, and added new variables. The remainder of this section defines each of these variables.

Educational Background (EDB) In the original study, the levels of this variable were *computer related* and *non-computer related*. Using this definition, 93 % of the participants in the replication would have been in the *computer-related* group. Furthermore, in the original study, the researchers divided the computer related group into two subgroups (*computer science/software engineering and other computer related*). Following this example, and because an inspection is a software engineering task, in the replication we defined the two levels of this variable to be *software engineering related* and *non-software engineering related*.

In the original study, the researchers classified the participants based solely upon the researchers' subjective opinion of how close each participant's major was to computing. In the replication we sought to use a more objective and systematic method for defining the levels of this variable. Because there is an overlap among the courses taught in various majors, we do not believe that educational background is a discrete variable based solely on the title of the major. Therefore, to determine the closeness of each major to software engineering, we compared the content of each major to the Software Engineering Education Knowledge (SEEK) (The Joint Task Force on Computing Curricula 2004). The more topics from the SEEK that are covered by courses in the major, the closer that major is to software engineering.

Most participants graduated from one of three Turkish universities (Middle East Technical University (METU), Bilkent University or Hacettepe University). To calculate how close each major is to the SEEK, we had three individuals (a PhD student in computer education and instructional technology and two professors teaching computer and software engineering for more than six years) independently compare the current curricula of each major, using essential course syllabi, against the SEEK's ten knowledge areas and key units. Each reviewer scored each major from each university on a scale from zero (no coverage of knowledge area) to one (full coverage of knowledge area) in each of the ten knowledge areas. If the major partially covered an area, the reviewers assigned it a score between 0 and 1 based on their comparison of the syllabi with the key units of the knowledge area. For each knowledge area, we averaged the nine scores (three reviewers for each of the three universities). Then we summed the averages from each of the ten areas to obtain the final score for each major. Table 1 provides an overview of the SEEK rating for each major along with the number of participants from each company who earned each degree in that major.

A second factor affecting **Educational Background** is **Educational Degree (DEG)**. In the original study, the participants only had bachelor or master's degrees. In the replication, some participants have PhDs. Therefore, we added a third level to this variable (*Undergraduate*, *Master's*, and *PhD*). The degree programs in Turkey and the US are equivalent.

In deciding whether a participant belonged to the *software engineering related* or *non-software engineering related* group we accounted for all his degrees. We conducted a small

Table 1 Majors along with their seek rating and the distribution of participants by company

Major	SEEK Rating	# Bachelors			# Masters			#PhD		
		A	B	C	A	B	C	A	B	C
Software Engineering	10	–	–	–	–	–	3	–	–	–
CTIS	9	1	–	–	–	–	–	–	–	–
Information Systems	8	–	–	–	1	–	2	–	–	–
Software Management	8	–	–	–	1	1	1	–	–	–
Statistics	4	1	–	1	–	–	–	–	–	–
Computer Engineering	3.4	8	7	18	1	4	3	–	–	1
MBA	3.2	–	–	–	1	–	–	–	–	–
Electrical Engineering	2	1	11	16	–	6	5	–	–	1
Physics	2	1	1	–	1	–	–	1	–	–
Chemical Engineering	2	1	–	–	1	–	–	–	–	–
Industrial Design	1	–	1	–	–	–	–	–	–	–
Aerospace Engineering	1	–	–	1	–	–	1	–	–	–
Industrial Engineering	0	–	–	–	–	1	–	–	–	–

survey of our industry contacts in Turkey (21 respondents from 7 companies) to determine how they weight various degrees (undergraduate, master's and PhD) when choosing inspectors. We asked them the following question: "Assuming all other characteristics are equal, if you are hiring someone to be an inspector and that person has Bachelors, Masters and PhD degrees, how much does the field in which each degree was earned affect your decision?". The results of the survey indicate that the relative importance of the undergraduate, master's and PhD degrees is 10, 4, and 1 respectively.

We used these two pieces of information (closeness to the SEEK and importance of various levels of degrees), to redefine the **Educational Background** variable. First, for each of a participant's degrees, we used the SEEK score for the field in which that degree was earned. As an example, take a participant who had an undergraduate degree in Physics, a Master's degree in electrical engineering and a PhD in computer engineering. The SEEK scores for each of those degrees are 2, 2, and 3.4 respectively. Then, we weighted each of those scores based on the formula derived from the survey ($10 * \text{undergraduate} + 4 * \text{Master's} + 1 * \text{PhD}$) to get their final score (10*2+4*2+1*3.4=31.4). While this variable is a continuous variable theoretically ranging from 0 to 150, its relative value among the participants is more meaningful than its absolute value.

Across all participants, the value of **Educational Background** ranged from 10 to 90. To create two groups (*software engineering related* and *non-software engineering related*) we used the average value of the variable as the dividing point (equal to or greater than average = *software related*; less than average = *non-software related*). The *software related* group contained 55 % of the participants and the *non-software related* group contained 45 % of the participants (the most even split possible given the data).

Industrial Experience (IND) This variable was measured in years. Unlike the original study, where 43 % of the participants had no industrial experience, the participants in the replication were more experienced. Approximately one-half of the participants had 4 years or less industrial experience and one-half had more than 4 years industrial experience. So, for the replication, the levels of this variable are <= *4 years* and >*4 years*.

Requirements Experience (REQ) and Inspection Experience (INS) Again, unlike the original study in which many participants lacked these types of experience, in the replication most participants possessed these types of experience. The variables were measured on a five-point scale: *1-none, 2-learned in class, 3-used in class, 4-used once industry,* and *5—used multiple times in industry.* Based on the distribution of the data, we defined two levels for each variable, *high* (those who answered 5) and *low* (everyone else). These definitions provided a fairly even division of the participants into the two levels for each variable.

The first new variable is **English Proficiency (ENG)**. Because none of the participants were native English speakers, this variable is important. On the background survey the participants rated their English reading ability and English writing ability on a five-point scale (from beginner to advanced). We used the maximum of these two values as the **English Proficiency** value. Then we collapsed the five-point scale into two groups for analysis: *low* (less than 5) and *high* (equal to 5).

The second new variable is **Artifact Format (ART)**, referring to whether the participants inspected were inspected using a *hardcopy* of the requirements (as in the original study) or a *softcopy* of the requirements (i.e. an online version). Reading is one of the major tasks of requirements inspection. Considerable research has been conducted to observe and understand the differences between reading hardcopy documents and online documents (Dillon et al. 1988). The results of these studies are not consistent (Noyes and Garland 2008; O'Hara and Sellen 1997). We included **artifact format** as a variable in this study to explore its impact on a requirements inspection.

The last new variable is **Company**, which has three levels, A, B, and C, each representing one of the three companies. Its addition was necessitated by the fact that we conducted the study in multiple locations.

Table 2 summarizes the variables and levels used in the original study and the replication.

Table 2 Comparison of independent variables between studies

Variable	Levels in Original Study	Levels in Replication	# of Participants
Educational Background (EDB)	Computer	Software Engineering	38
	Non-Computer	Non-Software Engineering	31
Educational Degree (DEG)	Undergraduate (Bachelor's)	Undergraduate	36
	Graduate (Master's)	Master's	30
		PhD	3
Industrial Experience (IND)	None	<= 4 years	39
	>0	>4 years	30
Requirements Experience (REQ)	None	Low	49
	>0	High	20
Inspection Experience (INS)	None	Low	31
	>0	High	38
English Proficiency Level (ENG)	N/A	Low	28
		High	20
Artifact Format (ART)	N/A	Hardcopy	27
		Softcopy	42
Company	N/A	A	13
		B	20
		C	36

2.2.2 Hypotheses

We investigated slightly modified versions of the five hypotheses used in the original study. The modifications resulted from the redefinition of the variables in Section 2.2.1. For the sake of space, we have omitted the null hypotheses.

- H_1: The effectiveness of an inspector is affected by their educational background (software engineering related or non-software engineering related).
- H_2: The effectiveness of an inspector is affected by their level of education (undergraduate, master's, PhD).
- H_3: The effectiveness of an inspector is affected by the amount of industrial software development experience they possess.
- H_4: Experience with requirements affects the effectiveness of an inspector.
- H_5: Inspection experience affects the effectiveness of an inspector.

The new variables we added in the replication allowed us to pose the following new hypotheses.

- H_6: English proficiency level affects the effectiveness of an inspector.
- H_7: Artifact format affects the effectiveness of an inspector.

In all cases, we used an alpha value of 0.05 for judging statistical significance.

2.3 Participants and Companies

The replication included 69 participants drawn from three companies in Turkey. We refer to these companies as A, B, and C according to the order of the experiments conducted. We conducted the replication at the Ankara offices of these companies in March–April 2009. The software development managers in each company selected the participants. We requested participants who were software developers and who had different educational backgrounds and different levels of experience. We did not inform the managers of the study hypotheses before they chose the participants. There were 13 participants from company A, 20 from company B, and 36 from company C. As part of the study, the participants received a training course on inspections.

The move from participants in a training course in a US company to regular developers in Turkish companies allowed us to study new variables and gain additional insight about the original variables. First, we could investigate the specific impact of a software engineering background rather than a general computing background, as in the original study. Second, we could study the effect of English proficiency because the participants were not native English speakers. Third, we could better investigate the effects of industrial experience. Because the participants came from the general population of the companies rather than from a training course, they were more experienced than the participants in the original study (average of 5.3 years of industrial experience vs. 1.9 years for the participants in the original study).

The move from one company to three companies also introduces some additional complicating factors. Because each company was likely to have its own personality and characteristics, it is important to understand these differences and their potential effects on the results of the replication. While the companies were of similar size, i.e. 50–100 developers, they did differ on some important attributes. First, the presence of international clients is important because when a company works for an international client or is involved in a multinational project, artifacts are generated in English and internationally accepted quality standards are utilized. The software quality assurance processes, including the fault detection process and inspections, may be impacted by the presence of international clients. All three companies conduct inspections. Companies A and B are both involved in or develop software for international projects (between 10 % and 99 % of projects for company A, less than 16 % for company B, and unknown for company C). Company A is the only company where the majority of its clients are international. Second, development of safety critical systems is important because developers typically place high value on inspections in this type of system. Companies A and C both develop safety critical software in the aviation domain. Most of company B and C's clients are military clients. Beyond these general characteristics, we do not have any specific details about how often inspections are used in each company.

2.4 Experimental Materials (Artifacts)

We used the lab package from the original experiment to guide the replication. The participants inspected the requirements document for the Loan Arranger system, which banks can use to bundle loans for resale to investors. The document was written plain (non-formal) English. It was 10 pages long, including 49 detailed requirements and seeded with 30 faults. In addition to being used in the original study, this general, i.e. non-company specific, document has also been used in other requirements inspection studies (Carver et al. 2003; Shull et al. 2001).

For the inspection checklist, we used the fault report form in the lab package, which also included the list of fault classes. The fault classes in the checklist focused mostly on the quality of the information in the requirements specification itself (not just the quality of the documentation). The fault classes included in the fault report form were: omission, ambiguity, inconsistency, extraneous information, incorrect fact, and miscellaneous (faults which do not fit in any other classes).

To study the effect of **Artifact Format**, some participants inspected a softcopy of the requirements while others inspected a hardcopy of the requirements. In the original experiment, all participants inspected a hardcopy of the requirements. We assigned the participants to the requirements format as follows: company A participants all inspected the hardcopy requirements, company B participants all inspected the softcopy requirements on laptop computers, company C participants were randomly assigned to either the hardcopy requirements or the softcopy requirements on desktop computers.

2.5 Experiment Execution

We followed the same process as in the original experiment (see Fig. 2 in the original paper (Carver et al. 2008)). We conducted the study onsite at each company. In companies A and B we used meeting rooms that are also used for training purposes. In company C we used training laboratories.

At each company, we performed the following experimental activities on the same day. First, Albayrak (the experimenter) presented an introduction to the experiment and a training session on inspections (60 minutes total). She presented the slides in English but spoke in Turkish. Following the training, the participants took a 10 minutes break. After the break, the experimenter gave the participants the experimental materials (requirements document, fault report lists, and background questionnaire). The participants spent 75 minutes finding faults and recording them on the fault list. The experimenter then collected the fault lists and background questionnaires (5 minutes).

Next, the experimenter distributed the actual list of faults to the participants. For each fault on that list, we asked each participant to state whether they agreed that it is a true fault, if they saw it and whether they think they reported it (30 minutes). The experimenter then collected these lists (5 minutes). Conducting experiments with professionals is expensive. As in the original experiment, we collected this post-study data with the expectation of conducting additional analyses in the future.

Later, after data analysis, Albayrak delivered a summary of the company-specific results to each company. She also discussed the faults with the participants and gave them a chance to provide feedback on the experiment (10 minutes).

3 Data Analysis

Table 3 provides descriptive statistics of the study data. Section 3.1 describes the preliminary analysis we performed prior to the main data analysis. Section 3.2 describes the results relative to the study hypotheses. Section 3.3 discusses the impact of False Positives on the results.

3.1 Preliminary Analysis

This section discusses the results of two preliminary analyses that guide the remaining analyses. First, due to the potential differences among the companies, as described in Section 2.3, we compared the performance of the participants across the three companies to determine whether that factor influenced the results. Second, due to the large number of variables and the likelihood that the variables were not independent, we conducted an exploratory factor analysis with the secondary variables, similar to the original study.

3.1.1 Company Analysis

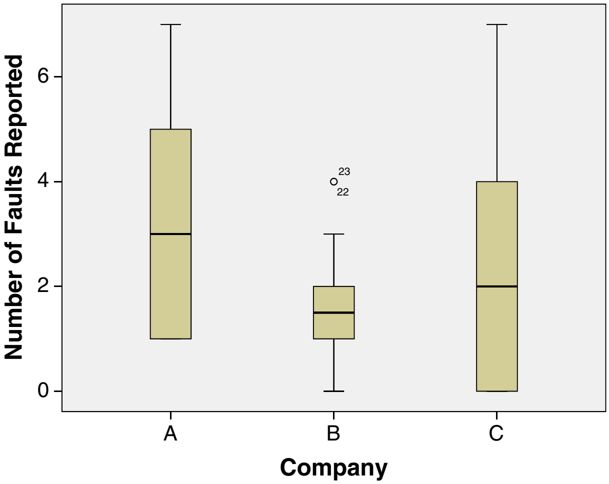
Based on the differences between companies described in Section 2.3, we expected that the participants from different companies may perform differently. Specifically, we expected that, on average, the participants from company A would perform the best (international clients and safety critical software), followed by those from company C (safety critical software) followed by those from company B. The boxplot in Fig. 1 shows that the **Company** variable does in fact impact the effectiveness of the participants. This data did not meet the ANOVA requirement for equality of population variance (Levine's test: $F=5.136$, $p=0.008$), so we performed a non-parametric Median test (which tests whether the medians of the samples are different). This test indicated that there was a significant

Table 3 Descriptive statistics

	Company	*N*	*Mean Faults*	*Std. Dev.*
Software Engineering Related	A	10	2.70	1.829
	B	7	1.86	1.345
	C	21	1.86	1.824
Non-Software Engineering Related	A	3	6.00	1.732
	B	13	1.54	1.127
	C	15	2.73	2.120

N Sample size (number of participants); *Mean Faults* Average number of correct faults found by participants; *Std. Dev.* Standard Deviation of the mean faults

Fig. 1 Faults vs. Company



difference between the companies (X_{69}^2=6.413, p=0.041). As expected, the participants from company A were the most effective followed by those from company C then company B. At this point, we cannot definitively conclude that the relationship between company and performance is causal, that is, we are not sure whether working for company A necessarily makes an inspector better. This observation should be tested further in later studies.

This result and the discussion in this section and in Section 2.3 indicate that there are differences in the performance of the participants from the different companies. Therefore, in the analysis of the replication results, to investigate the impact of the **Company** variable on the other independent variables, we conduct a series of 2-way ANOVAs with **Company** and each other variable.

3.1.2 Initial Analysis of Variables

Because we measured multiple independent variables regarding the participants' background, we needed to ensure that all variables should remain in the detailed analysis. The goal of this analysis was to determine whether the secondary variables (i.e. the variables other than **EDB** and **Company**) were independent from each other or whether they were statistically related.

The first step was to perform an Exploratory Factor Analysis (EFA) to determine whether there were any underlying latent factors. For this analysis we considered the **REQ**, **INS**, **IND**, **DEG**, **ART** and **ENG** variables. We excluded the main variable of interest, **EDB**, and the **Company** variable from the EFA. The Principle Components Analysis with Varimax Rotation extracted two factors which explain 64 % of the variance. Factor 1 included the **REQ**, **INS**, **IND** and **DEG** variables. Factor 2 included the **ART** and **ENG** variables. The variables that loaded on Factor 1 are all clearly related to a latent construct of *Experience*. The Chronbach's Alpha test of the reliability of this factor resulted in high reliability (0.719). The alpha value did not increase when any of the variables were removed. Therefore, we create a new variable **Experience** (**EXP**) by summing the values of these four variables. The **REQ**, **INS** and **IND** variables had values of *0—Low* or *1—High*. Because the **DEG** variable had three values and because we wanted it to have the same weight as the other three variables, we converted the values to *0—Bachelor's Degree*, *0.5—Master's Degree* and *1—PhD*. Therefore the new **EXP** variable ranges from 0 to 4. Finally, we converted the new

EXP variable into a two-level variable like the other variables, as follows. Participants who had a high value on at least 2 of the underlying variables (i.e. a score of 2) were considered *High* experience. The rest were considered *Low* experience. This division resulted in a split of 72 % in the Low group and 28 % in the High group. As a result of this new variable, we also added a new hypothesis:

> H_8: Experience affects the effectiveness of an inspector.

The second step was to ensure that distributions of the remaining variables were independent from each other to determine if any of the remaining variables should be removed from the analysis. To make this determination, we conducted a Chi-Square analysis. The results in Table 4 shows that the distributions of the ENG and ART variables were not independent. One had to be eliminated. Because the ART variable was not evaluated in all companies, we chose to eliminate it from the analysis and keep the ENG variable.

3.1.3 Overview of Main Analysis

Based on the preliminary analysis in the previous two sections, three variables remained for the main analysis: **EDB**, **EXP** and **ENG**. we conducted a 2-way anova for each variable combined with the **Company** variable.

The following section presents the detailed results for each variable, including a table with the *F*-value, the *p*-value, Cohen's effect size (f^2) and the post-hoc power computed by SPSS. Cohen's effect size is an indication of the magnitude of the effect measured in the study. Effect sizes are classified into *small* (>0.02), *medium* (>0.15) and *large* (>0.35). To guard against family-wise error, we used the Bonferroni correction for the three ANOVA tests. Therefore, instead of the alpha value being 0.05, it is reduced to 0.017 (0.05/3).

3.2 Results

> H_1: The effectiveness of an inspector is affected by their educational background (software engineering related or non-software engineering related).

Figure 2 shows that participants who had a software engineering background were less effective than those who had a non-software engineering background. Table 5 provides additional descriptive details including the mean, standard deviation and count for each condition. The detailed results of the 2-way ANOVA in Table 6 shows a significant main effect for **EDB** (with a small effect size) and for **Company** (with a medium effect size) and a non-significant interaction between the two. Overall, we can conclude that participants with a degree in a software related field were significantly less effective inspectors. Even though the interaction between **EDB** and **Company** was not significant (after the Bonferroni correction), Fig. 2

Table 4 Chi-square results (X^2 value/*p*-value)

	EDB	EXP	ENG	ART
EDB	–	0.084/0.771	0.689/0.406	1.116/0.291
EXP	0.084/0.771	–	1.948/0.163	3.598/0.058
ENG	0.689/0.406	1.948/0.163	–	5.073/0.024
ART	1.116/0.291	3.598/0.058	5.073/0.024	–

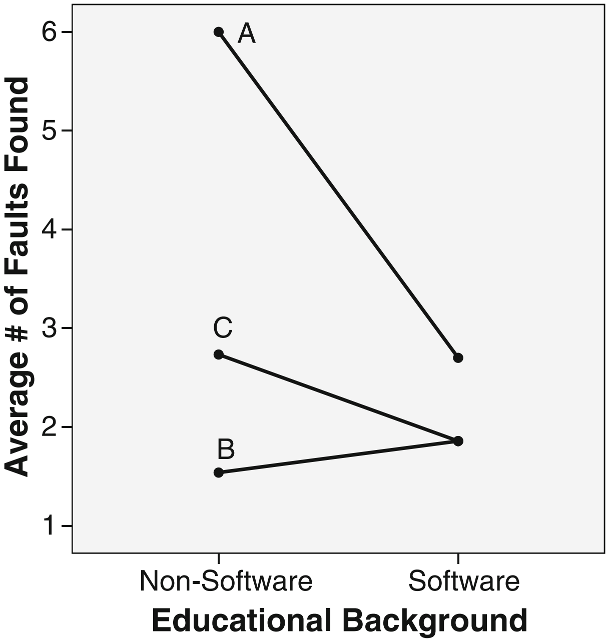


Fig. 2 Educational Background vs. Company

shows that a large portion of the variation appears to come from the participants from companies A and C. The participants from company B show little variation.

H$_8$: Experience affects the effectiveness of an inspector.

Figure 3 shows that participants who had *high experience* were more effective than those who had *low experience*. Table 7 provides addition descriptive statistics, including the mean, standard deviation and count for each condition. The results of the 2-way ANOVA in Table 8 showed significant main effects for **EXP** (with a medium effect size) and **Company** (with a medium effect size) but no significant interaction between the two variables. These results indicate that overall the more experience someone has, the more effective he will be during an inspection.

H$_6$: English proficiency level affects the effectiveness of an inspector.

Note that for the **ENG** variable, not all of the participants provided information, so the N is lower than for the other variables. Figure 4 shows that for the participants from companies A and C, those with higher **English Proficiency** were more effective. For the participants from company B there was little effect related to **English Proficiency**. Table 9 provides additional descriptive details including the mean, standard deviation and counts for each condition. The results of the 2-way ANOVA in Table 10 showed a significant main effect of **English Proficiency** (with a medium effect size) and a non-significant main effect of **Company,** with

Table 5 Descriptive statistics for Educational Background. Mean/ Standard Deviation (Count)

	Company A	Company B	Company C
Non-Software	6/1.732 (3)	1.54/1.127 (13)	2.73/2.120 (15)
Software	2.7/1.829 (10)	1.86/1.345 (7)	1.86/1.824 (21)

Table 6 ANOVA results for Educational Background

	Effect	*N*	*n*	*d*	*F*	*p*	f^2	*Power*
H_1	Educational Background (EDB)	69	1	63	6.373	0.014	0.101	0.701
	Company	69	2	63	7.299	0.001	0.232	0.926
	EDB*Company	69	2	63	3.295	0.044	0.105	0.605

N is the number of data points analyzed; *n*, *d* are the numerator and denominator degrees of freedom, *F* is the value of the ANOVA *F*-test, *p* is the probability of the *F*-test, f^2 is Cohen's effect size, *power* is the post hoc power

no significant interaction between the variables. These results suggest that inspectors with a higher proficiency in English tend to find more faults when inspecting an English-language requirements document.

3.3 Discussion About False Positives

In addition to the primary results relative to **Inspection Effectiveness**, we also collected data on false positives (i.e. faults reported by participants that did not match faults seeded into the requirements document). Due to space limitations, we only provide a high-level analysis of the **False Positive** variable.

The analysis of false positives provides additional insights into the results discussed in the previous subsection. Specifically, we classified every item reported by a participant to be either a true fault or a false positive. By analyzing the false positives, we wanted to determine whether the participants who were more effective in finding true faults achieved this status simply by reporting more potential faults (and therefore having more false positives).

In general, there is no significant difference between the average number of false positives reported by participants in the *software related* group and the average number of

Fig. 3 General Experience vs. Company

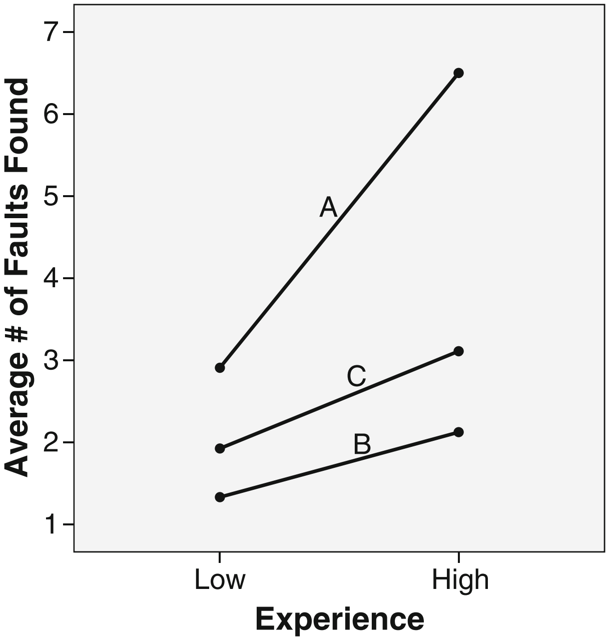


Table 7 Descriptive statistics for Experience. Mean/Standard Deviation (Count)

	Company A	Company B	Company C
Low	2.91/1.973 (11)	1.33/1.155 (12)	1.93/1.979 (27)
High	6.50/0.707 (2)	2.13/1.126 (8)	3.11/1.764 (9)

false positives reported by the participants in the *non-software related* group (9.47 vs. 9.16). In addition, there is only a very weak correlation between faults and false positives (r^2= 0.092). Therefore, we can conclude that the results reported in Section 3.1 do not seem to have been caused by over reporting of potential faults.

4 Comparison of Results Across Both Studies

This section compares the results of the replication with those from the original study. Based on the definitions provided by Shull et al. (Shull et al. 2008), this replication is an exact, dependent replication because it uses the same artifacts, analysis and procedures as the original study. In addition, we expanded on the original study by adding some new variables. Section 4.1 discusses results that were consistent between the studies. Section 4.2 discusses results that were different between the two studies. Table 11 provides an overview of the results from both studies.

4.1 Consistent Results

Due to our redefinition of the variables, which was necessitated by the sample (see Section 2.2.1), we cannot make a direct comparison between all of the results of the two studies. However, for the **Educational Background** variable, the replication produced similar results as the original study, which provide additional insight into the phenomena under study.

The results of the original study showed that the inspectors with degrees in *non-computer related* majors were significantly more effective than inspectors with degrees in *computer related* majors. It also showed that of the computer related participants, those with computer science or software engineering degrees were less effective than participants with other computer related degrees.

In the replication, we investigated the effect of software engineering background in more detail. The results showed that participants who had degrees that contained a large amount of software engineering were significantly less effective than participants who had degrees which contained less software engineering. This result is especially evident in participants from companies A and C.

Table 8 ANOVA results for Experience

	Effect	*N*	*n*	*d*	*F*	*p*	f^2	*Power*
H_8	Experience (EXP)	69	1	63	10.867	0.002	0.172	0.901
	Company	69	2	63	7.358	0.001	0.233	0.928
	EXP*Company	69	2	63	1.690	0.193	0.054	0.343

N is the number of data points analyzed; *n*, *d* are the numerator and denominator degrees of freedom, *F* is the value of the ANOVA *F*-test, *p* is the probability of the *F*-test, f^2 is Cohen's effect size, *power* is the post hoc power

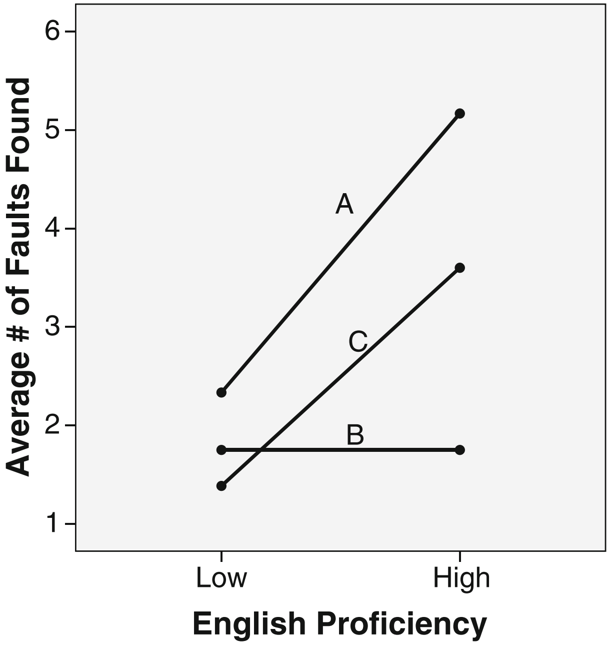


Fig. 4 English Proficiency vs. Company

4.2 Differences in Results

The replication did not produce any results that were inconsistent with the original study. Due to the introduction of new variables, the replication did produce some new results. Because we defined three new variables in the replication, **English Proficiency**, **Artifact Format**, and **Company**, their results could not be compared with the original study. The replication showed that the **Company, English Proficiency** and our new **Experience** variables all had a significant impact on effectiveness.

5 Discussion of Results Across Both Studies

The results of the both the original study and the replication indicate that **Educational Background** is an important factor to consider when choosing members for a requirements inspection team. The results of the replication also indicate that **General Experience** and **English Proficiency** are also important factors.

As described in Section 1.2, in the replication we made some deliberate changes to the design of the original study to broaden the applicability of the results. First, by moving the study from one US-based company to three Turkish-based companies we could investigate the effects of **English Proficiency**. Second, by having some participants use hardcopy artifacts (as in the

Table 9 Descriptive statistics for English. Mean/Standard Deviation (Count)

	Company A	Company B	Company C
Low	2.33/2.309 (3)	1.75/1.357 (12)	1.38/1.502 (13)
High	5.17/1.722 (6)	1.75/0.957 (4)	3.6/1.776 (10)

Table 10 ANOVA results for English

	Effect	*N*	*n*	*d*	*F*	*p*	f^2	*Power*
H_6	English (ENG)	48	1	42	10.192	0.003	0.242	0.877
	Company	48	2	42	3.885	0.028	0.185	0.671
	ENG*Company	48	2	42	2.573	0.088	0.122	0.486

N is the number of data points analyzed; *n*, *d* are the numerator and denominator degrees of freedom, *F* is the value of the ANOVA *F*-test, *p* is the probability of the *F*-test, f^2 is Cohen's effect size, *power* is the post hoc power

original study) and others use softcopy artifacts, we investigated the effect of **Artifact Format**. Finally, the redefined **Educational Background** variable focuses on software engineering knowledge rather than more general computing knowledge as in the original study. The discussion of results in Section 3 indicated that these variables had an effect on the results.

We argue that these changes explain why participants in the replication found fewer faults on average than participants in previous studies using the Loan Arranger (including the study upon which we based the replication). To support this argument, we analyzed the performance of the subset of replication participants who were most similar to the participants in the original study (i.e. have high English proficiency and used the hardcopy artifact). This subset contains 13 participants. These participants performed quite similarly to the participants in the original study. The *software related* participants found 3.5 faults compared with 4 for the *computer related* participants in the original study. The *non-software related* participants found 5.6 faults compared with the *non-computer related* participants in the original study who found 5.5 faults. While we could not statistically test the differences between the original study and the replication, qualitatively, these numbers are similar. Furthermore, in this reduced subset, the *non-software related* participants outperformed the *software related* participants (5.6 faults vs. 3.5 faults). The reduced dataset lowered the power of the statistical test, therefore the result of the statistical test (t_{11}= 2.003, p=0.07) exceeds the selected threshold for significance used for the other results (0.017).

Relative to **Educational Background**, the original study showed that inspectors with a computer related background were significantly less effective than those with a non-computer background. It also showed that within the computer related group, those who were closer to software engineering (i.e. computer science and software engineering majors) were the least effective of all (though the result was not significant). The results of the replication strengthened this result and provided additional insights about this factor. We found that inspectors with a software engineering background were significantly less effective than those without a software engineering background. The replication increased the external validity of this result by showing that the impact and the direction of **Educational Background** variable are not specific to Microsoft.

The fact that inspectors with a non-software engineering background were more effective than those with a software engineering background may be a result of the skill set required to

Table 11 Results overview

Variable	Original Study	Replication
Educational Background	√	√
Experience	√ (Requirements Experience)	√
English Proficiency	N/A	√
Company	N/A	√

√ = statistically significant difference inspect requirements. It is quite possible that this skill set is obtained more frequently by people who earn degrees in non-software engineering related fields. For example, the ability to quickly switch between multiple documents may have an impact on inspection performance as inspectors have to alternate between the document being inspected and the fault report list (Hungerford et al. 2004). Another potential explanation is the difference in training that the different types of participants receive. Software engineers are trained to develop software as a product, while non-software engineers are trained to use software as a tool to solve other problems. As suggested in the original study, it is possible that when a software engineer reads a requirement they begin thinking in terms of design and coding while a non-software engineer may think in terms of how they would use the software. This difference in thought process, if it could be established empirically, would suggest a reason why participants with different backgrounds find different faults. Now that this effect has been observed in multiple studies, we need to conduct further studies to better understand why this effect occurred.

To further investigate the **Educational Background** variable, we plan to conduct a qualitative analysis of the participants' feedback about the master fault list. We believe that such an analysis will help to reveal any systematic differences in the interpretation of a requirements fault by inspectors with different backgrounds.

The original study also suggested that **Requirements Experience** should be explored in future studies. While in the end we did not analyze this variable directly (it became part of the new **Experience** variable), we did conduct some qualitative analysis related to it. After the replication, we interviewed some of the participants, two of the software development managers and an independent consultant familiar with all of the companies. They indicated that the study participants who wrote requirements as part of their normal work (i.e. had a high value for the **Requirements Experience** variable) were trained on how to write high-quality requirements. Thus, those experienced in writing requirements were familiar with fault classes and knew how to avoid making faults, which may have impacted their performance on the inspection task.

During the post-study interview, almost 30 % of the participants suggested that the type of the requirement document they wrote and inspected could be an important factor in inspection effectiveness. Most of the participants from company B indicated that the Loan Arranger requirements were not ready for inspection. This fact could help to explain why the participants from company B were the least effective overall. The abstraction level of the requirements and the presentation of the requirements may also be important factors. Thus, the **Requirements Experience** variable and the type of requirements document inspected should be explored further in future studies.

6 Threats to Validity

As with any empirical study, this replication has a number of threats to validity. This section indicates the threats to validity from the original study addressed by the replication and discusses the threats that remained unaddressed.

6.1 Internal Validity

Internal validity refers to the accuracy of the conclusions drawn from the data collected. Threats to internal validity concern issues other than the identified independent variables that may cause the observed difference in the dependent variable (Fraenkel and Wallen 2006). The presence of internal validity threats reduces the confidence the researcher and the reader can have in the accuracy of the results.

Selection Bias means that the results could be caused by the specific participants chosen for the study or by the way those participants were assigned to the experimental groups. As described earlier, the managers of each company chose the participants without knowledge of the study hypotheses. Therefore, the likelihood of a selection threat in this replication is minimized. The companies were chosen as a convenience sample because we had access to conduct the experiments and follow-up.

Instrumentation Bias means that the results could be affected by the means of participant instruction or data collection. In the original study, there was a threat of instrumentation bias because the researcher who conducted the study was a manager within the company from which the participants were drawn. This situation could cause the participants to behave differently than normal and result in a biased result. In the replication, the first author collected the data in all three companies. She had no direct connection with the companies. Furthermore, we used the same data collection instruments as in the original study. Overall, the replication reduced the potential impact of instrumentation bias.

6.2 Construct Validity

Construct validity refers to the accuracy of the operationalization of the study constructs (variables and artifacts). The presence of construct validity threats suggests that the study may have been incorrectly measuring the intended phenomena and therefore reduces the confidence one can have in the results. The main threat to construct validity we addressed in the replication was related to the operational definitions of the variables.

If the operational definitions of the independent and dependent variables do not properly match the intended theoretical construct, the results of the study have less reliability. The construct validity for the dependent variable is identical to that of the original study. *Number of correctly identified faults* is a common measure for inspection effectiveness. Therefore, there is not a construct validity threat present. One unaddressed threat related to the dependent variable is that we did not take fault severity into account when analyzing the data.

In the replication we sought to increase the construct validity of the main independent variable, **Educational Background**. Instead of categorizing participants based on a subjective opinion of which majors were computer related, we used a more objective approach. By comparing the content of each degree program to the SEEK, we reduced the threat of mischaracterizing a participant. Conversely, by weighting the value of the Bachelor's, Master's and PhD degrees based on a survey of the companies, it is possible that we introduced a new bias.

Finally, the participants self-reported their level of **English Proficiency**. It is possible that participants overestimated or underestimated their proficiency. In addition, as all participants were non-native English speakers, we could not analyze the effects of an inspectors' primary language being the same as or different from their working language, only the effects of the level of self-reported knowledge of the working language. In the future we could use a more objective test to measure this variable.

6.3 External Validity

External validity refers to the extent to which the results of a study can be generalized. In both the original study and the replication, the participants were professional developers. This choice of participants increases the external validity and makes it more likely that the results will hold for other professionals than if we had chosen students as the participants, which is the case with many empirical studies in software engineering.

One of the main goals of the replication was to increase external validity. Therefore, we replicated the study in three companies in a different country than the original study. This change increased the external validity of the results, especially those discussed in Sections 4 and 5, which were drawn from both studies together.

While we generally increased the external validity, one external validity threat from the original study remained unaddressed. By reusing the requirements document from the original study, we retain the threat that the document may not be representative of industrial requirements documents. The fact that the requirements document was in a textual format (rather than a formal notation) could limit the results to similar kinds of documents. However, by using the same artifact as in the original study, we did not introduce any new variation. Therefore, we cannot be confident that the same results would hold when using a real, industrial requirements document. Additional replications are necessary to address this question.

A second threat related to the requirements document used in the study is the potential that the domain was unfamiliar to the participants. We did not specifically measure this variable, but it is likely that the participants from Turkey would not have been as familiar with the type of financial system described in the requirements. Unfamiliarity with the domain could lead to a lower number of faults identified.

A final threat to external validity is the potential for sampling bias. Overall, this replication reduces the sampling bias by including participants from three companies rather than only one. Conversely, all study participants were software developers. In practice, requirements inspections may be conducted by people other than developers. Therefore, we cannot be confident that these results will apply to all types of inspectors. In fact, if the results from our study hold, it would suggest that people who are not software developers, or at least who do not have a software engineering background, may in fact, be better inspectors. Further study will investigate these issues.

6.4 Replication Validity

We introduce a new type of validity specifically related to replications. Replication Validity describes the ability to compare the results of the replication with those from the original study. In our case, we redefined the **Educational Background** variable and we introduced a new **Experience** variable. For both variables, we made these changes to increase external validity and construct validity. By doing this, we potentially reduced our ability to compare the results of the replication directly with the results of the original study.

6.5 Conclusion Validity

Conclusion validity refers to the degree to which the conclusions are reasonable given the data. Threats to conclusion validity are: 1) concluding that there is no relationship when one is actually present and 2) concluding that a relationship is present when in reality none exists. Based on the results of the replication, the second threat is the most concerning one. To reduce the chance of drawing an incorrect conclusion, we used the Bonferroni correction to correct for family-wise error.

7 Contributions and Limitations

We used much of the same procedures, artifacts, inspection techniques, hypotheses, measurements and analysis techniques as used in the original study. The greater the degree of similarity between the studies, the more confidence one can have in the results of a comparative analysis. We made deliberate variations in using new measurements for some variables and adding new variables to increase the external validity and allow for a higher degree of generalization. The original study was conducted only in one company, while the replication was conducted in three companies, thus we extended the results to new environments and investigated the range of conditions under which the findings hold.

Another contribution of this study was to minimize the limitations of the original study by addressing instrumentation bias, increasing external validity and redefining the previously subjective measure of the **Educational Background** variable. Measurement and quantification of a person's educational background is not discrete and easy, especially when the participants have multiple degrees (i.e. a Bachelor's and a PhD) in different fields. Instead of relying on the subjective judgment of the researchers, as in the original study, we defined a more objective formula.

A final contribution relates to the successful completion of a (semi-)independent replication. As mentioned in Section 1.2, Juristo and Vegas note that a replication tends to be successful only when the original researchers conduct the replication at the same site. We claim that we have performed a successful replication (i.e. obtaining similar results), without fulfilling this requirement. Only one of the researchers was common to both studies. The replication was conducted in a different context than the original study. These results indicate that with adequate communication and support, a successful replication can be conducted in a different setting with different researchers than the original study.

There are also several limitations in the replication. First, we used professional developers, but we did not use a repeated measure design that included a control group. Second, the artifact inspected could be a limitation of the study. It is not clear what results we would obtain if we used a different and more complex requirements document. Third, we only investigated the impact of individual characteristics on inspection effectiveness. All participants used a checklist-based approach. We did not study the effect of different inspection techniques. This topic will be investigated in future studies. Fourth, some of the limitations in the study stem from participants' characteristics. Most participants were developers rather than formally trained inspectors. Finally, we measured effectiveness as the number of faults, treating all faults equally. We did not study how the severity of faults or the type of faults could impact the results. This topic will also be studied in more detail in the future.

8 Summary and Future Plans

This replication contributes to the body of knowledge about the impact of individual factors on the effectiveness of requirements inspections. The variables studied include: **Educational Background**, **Educational Degree**, **Industrial Experience**, **Requirements Experience**, **Inspection Experience**, **English Proficiency**, **Artifact Format** and **Company**. The replication had 69 participants drawn from three companies. The main goals of the replication were to: 1) provide further insight into the results from the original study and 2) introduce new variables and new methods for measuring the variables from the original study.

For the first goal, across both studies we can draw conclusions about two variables. First, relative to the **Educational Background** of an inspector, we can conclude that, at best, having a computer science or a software engineering background is not helpful during an inspection. At worst, that experience is actually harmful. In the original study and in two out of three companies in the replication inspectors who had a strong computer science or software engineering background are significantly less effective than the other inspectors.

Relative to **Experience**, the studies have slightly different, but not necessarily contradictory results. In the original study participants with more **Requirements Experience** were significantly more effective inspectors. In the replication, the **Requirements Experience** variable was combined with the other experience variables into a new **Experience** variable. Participants who had high **Experience** were significantly more effective inspectors. The replication result is a more general instance of the original study's results. The **Experience** variable needs to be studied further to determine how best to measure it in the context of an inspection.

For the second goal, of the three new variables introduced, two had a significant impact on inspector effectiveness. The first variable was **English Proficiency**. In the original study, conducted in the United States, most of the participants were native English speakers and working in English, so this variable was not present. Conversely, the participants in the replication were not native English speakers. An inspector's level of **English Proficiency** had a significant positive effect on his inspection effectiveness. This result makes sense because the experimental materials were presented in English. The second variable was **Company**. The company which a participant worked for had a significant impact on their effectiveness. Further study is needed to understand which characteristics of the company may have caused this observation. Our initial ideas are: 1) whether the company has multinational clients and 2) whether the company writes safety-critical software.

Based on the results of this replication, we can propose some immediate next steps. First, using the data collected after the study, we plan to analyze which faults are found more often and to explain whether certain types of faults are found by different type of participants.

Next, we will further investigate how to measure of **Educational Background**. It is important to determine what type of educational background is particularly relevant to the requirement inspection task. In addition, to explore the impact of the company variable in more detail, we will investigate other variables including: different artifacts and different inspection techniques. We will also investigate the interaction among these variables to determine whether the same effect is present when inspectors are using other inspection techniques, like Perspective Based Reading (PBR) (Basili et al. 1996). We plan to conduct a qualitative study to determine the types of expertise that inspectors believe will improve their effectiveness. This information could be used to augment the **Educational Background** variable. Regarding the dependent variable, in future studies we suggest the use of different measurements for inspection effectiveness, including fault severity.

Finally, additional replications are needed to investigate the interaction of the **Educational Background** variable with other important variables including: inspection reading technique and type of requirements.

Ultimately the goal is to improve the practice of software requirements inspections. The outcomes of this line of studies will inform universities about which skills need to be taught to their students. It will also inform industry about the most appropriate types of people to select for inspection teams and the most appropriate types of training to provide to their requirements inspectors.

We encourage other researchers to replicate this study in different contexts to provide more data, which will enable us to draw broader conclusions. To facilitate replications, we have posted a lab package: (http://steel.cs.ua.edu/~carver/BackgroundReplication). Interested researchers are invited to contact us for more information.

Acknowledgment We thank the employees and management of the companies for their participation. We thank Jorge L. Diaz-Herrera, Mustafa Akgül and Duygu Albayrak, Erhan Yüceer, David Davenport for helping the education survey. We thank Natalia Juristo for comments on the paper.

References

Aceituna D, Do H, Walia GS, Lee S-W (2011) Evaluating the use of model-based requirements verification method: A feasibility study. In: Proc. 2011 First International Workshop on Empirical Requirements Engineering (EmpiRE), pp. 13–20

Andersson C (2007) A replicated empirical study of a selection method for software reliability growth models. Empir Softw Eng 12(2):161–182

Aurum A, Petersson H, Wohlin C (2002) State-of-the-art: software inspections after 25 years. Soft Test Verif Rel 12(3):133–154

Aurum A, Wohlin C, Petersson H (2005) Increasing the understanding of effectiveness in software inspections using published data sets. J Res Pract Inf Technol 37(3):253–266

Basili VR, Green S, Laitenberger O, Lanubile F, Shull F, Sørumgård S, Zelkowitz MV (1996) The empirical investigation of perspective-based reading. Empir Softw Eng 1(2):133–164

Biffl S (2000) Analysis of the impact of reading technique and inspector capability on individual inspection performance. In: Proc.7th Asia-Pacific Softw. Eng. Conf, pp. 136–145

Carver J (2003) The Impact of Background and Experience on Software Inspections. PhD Thesis. Dept. of Comp. Sci., Univ. of MD.

Carver J (2004) The impact of background and experience on software inspections. Empir Softw Eng 9 (3):259–262

Carver J, Lemon K (2005) Architecture reading techniques: A feasibility study. In: Proc.4th Int'l Symp. on Emp. Softw. Eng. (Late Breaking Research Track). pp. 17–20

Carver J, Shull F, Basili V (2003) Observational studies to accelerate process experience in classroom studies: An evaluation. In: Proc.2nd Int'l Symp. on Emp. Softw. Eng., pp. 72–79

Carver J, Shull F, Basili VR (2006) Can observational techniques help novices overcome the software inspection learning curve? an empirical investigation. Empir Softw Eng 11(4):523–539

Carver JC, Nagappan N, Page A (2008) The impact of educational background on the effectiveness of requirements inspections: an empirical study. IEEE Trans Softw Eng 34(6):800–812

Ciolkowski M (2009) What do we know about perspective-based reading? an approach for quantitative aggregation in software engineering. In: Proc.3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009), pp. 133–144.

Dillon A, McKnight C, Richardson J (1988) Reading from paper versus reading from screen. Comput J 31 (5):457–464

Fagan ME (1976) Design and code inspections to reduce errors in program development. IBM Syst J 15 (3):182–211

Fagan ME (1986) Advances in software inspections. IEEE Trans Softw Eng SE-12(7):744–751

Fraenkel JR, Wallen NE (2006). How to design and evaluate research in education, 6th edn. McGraw-Hill Publishing Company, New York

Fusaro P, Lanubile F, Visaggio G (1997) A replicated experiment to assess requirements inspection techniques. Empir Softw Eng 2(1):39–57

Garousi V (2010) Applying peer reviews in software engineering education: an experiment and lessons learned. IEEE Trans Educ 53(2):182–193

Hungerford BC, Hevner AR, Collins RW (2004) Reviewing software diagrams: a cognitive study. IEEE Trans Softw Eng 30(2):82–96

Johnson PM, Tjahjono D (1997) Assessing software review meetings: A controlled experimental study using CSRS. In: Proc.9th Int'l Conf. on Softw. Eng, pp. 118–127

Juristo N, Vegas S (2009) Using differences among replications of software engineering experiments to gain knowledge. In: Proc.3rd Int'l Symp. on Emp. Softw. Eng. and Measurement, pp. 356-366

Kitchenham BA (2008) The role of replications in empirical software engineering—a word of warning. Empir Softw Eng 13(2):219–221

Kollanus S (2009) Experiences from using ICMM in inspection process assessment. Softw Qual J 17(2):177–187

Kollanus S (2011) ICMM—a maturity model for software inspections. J Softw Maint Evol Res Pract 23(5):327–341

Kollanus S, Kosnimen J (2009) Survey of software inspection research. The Open Software Engineering Journal 3:15–34

Laitenberger O, DeBaud J (1997) Perspective-based reading of code documents at Robert Bosch GmbH. Inf Softw Technol 39(11):781–791

Laitenberger O, Atkinson C, Schlich M, El Emam K (2000) An experimental comparison of reading techniques for defect detection in UML design documents. J Syst Softw 53(2):183–204

Laitenberger O, Emam KE, Harbich TG (2001) An internally replicated quasi-experimental comparison of checklist and perspective-based reading of code documents. IEEE Trans Softw Eng 27(5):387–421

Land LPW, Wong B, Jeffery R (2003) An extension of the behavioral theory of group performance in software development technical reviews. In: Proc.10th Asia-Pacific Softw. Eng. Conf., pp. 520–530

Lung J, Aranda J, Easterbrook SM, Wilson GV (2008) On the difficulty of replicating human subjects studies in software engineering. In: Proc.30th International Conference on Software Engineering (ICSE), pp. 191–200.

Martin J, Tsai W (1990) N-fold inspection: a requirements analysis technique. Commun ACM 33(2):223–232

McCarthy P, Porter A, Siy H, LG Votta J (1996) An experiment to assess cost-benefits of inspection meetings and their alternatives: A pilot study. In: Proc. Metrics, pp. 100

McMeekin DA, von Konsky BR, Robey M, Cooper DJA (2009) The significance of participant experience when evaluating software inspection techniques. In: Proc. Australian Software Engineering Conference (ASWEC'09), pp. 200–209

Noyes JM, Garland KJ (2008) Computer- vs. Paper-based tasks: are they equivalent? Ergonomics 51 (9):1352–1375

O'Hara K, Sellen A (1997) A comparison of reading paper and on-line documents. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems, pp. 335–342

Olalekan AS, Adenike OO (2008) Empirical study of factors affecting the effectiveness of software inspection: a preliminary report. Eur J Sci Res 19(4):614–627

Parnas DL, Weiss D (1985) Active design reviews: principles and practice. In: Proc.8th Int'l Conf. on Softw. Eng., pp. 132–136.

Porter A, Votta L, Basili VR (1998) Comparing detection methods for software requirements inspections: a replication using professional subjects. Empir Softw Eng 3(4):355–379

Regnell B, Runeson P, Thelin T (2000) Are the perspectives really different? Further experimentation on scenario-based reading of requirements. Empir Softw Eng 5(4):331–356

Robbins B, Carver J (2009) Cognitive factors in perspective-based reading (PBR): A protocol analysis study. In: Proc.3rd International Symposium on Empirical Software Engineering and Metrics. Oct. 15–16, pp. 145–155.

Sandahl K, Blomkvist O, Karlsson J, Krysander C, Lindvall M, Ohlsson N (1998) An extended replication of an experiment for assessing methods for software requirements inspections. Empir Softw Eng 3(4):327–354

Sauer C, Jeffery DR, Land L, Yetton P (2000) The effectiveness of software development technical reviews: a behaviorally motivated program of research. IEEE Trans Softw Eng 26(1):1–14

Schneider GM, Martin J, Tsai WT (1992) An experimental study of fault detection in user requirements documents. ACM Trans Softw Eng Methodol 1(2):188–204

Shull F, Carver J, Travassos G (2001) An empirical methodology for introducing software processes. In: Proc. Joint 8th Eur. Softw. Eng. Conf. and 9th ACM SIGSOFT Foundations of Softw. Eng. Sept. 10–14, 2001, pp. 288–296

Shull F, Basili V, Carver J, Maldonado J, Travassos G, Mendonca M, Fabbri S (2002) Replicating software engineering experiments: Addressing the tacit knowledge problem. In: Proc.1st Int'l Symp. on Emp. Softw. Eng. Oct. 3–4, 2002, pp. 7–16

Shull F, Mendonca M, Basili V, Carver J, Maldonado J, Fabbri S, Travassos G, Ferreira M (2004) Knowledge-sharing issues in experimental software engineering. Empir Softw Eng 9(1):111–137

Shull F, Carver J, Vegas S, Juristo N (2008) The role of replications in empirical software engineering. Empir Softw Eng 13(2):211–218

The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, Software Engineering (2004) Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, Retrieved July 19, 2012 from http://sites.computer.org/ccse/SE2004Volume.pdf

Travassos G, Shull F, Fredericks M, Basili V (1999a) Detecting defects in object oriented designs: Using reading techniques to increase software quality. In: Proc. OOPSLA '99

Travassos G, Shull F, Carver J (1999b) Reading techniques for OO design inspections. In: Proc. 24th NASA Softw. Eng. Wksp.

Votta L (1993) Does every inspection need a meeting? In: Proc. ACM SIGSOFT Symp. on the Foundations of Softw. Eng, pp. 107–114

Winkler D, Thurnher B, Biffl S (2007) Early software product improvement with sequential inspection sessions: An empirical investigation of inspector capability and learning effects. In: Proc.33rd EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 245–254

Winkler D, Biffl S, Faderl K (2010) Investigating the temporal behavior of defect detection in software inspection and inspection-based testing. In: Proc. Product-Focused Software Process Improvement., pp. 17–31

Wong YK (2011) Do developers matter in system review? Behav Inform Technol 30(3):353–378

Zhang Z, Basili V, Shneiderman B (1999) Perspective-based usability inspection: an empirical validation of efficacy. Empir Softw Eng 4(1):43–70

Özlem Albayrak is an instructor at Computer Technology and Information Systems department of Bilkent University, Turkey. She received the PhD degree from Ankara University (2002), MBA(1994) and BSc. in Computer Engineering and Information Sciences(1992) from Bilkent University. She enjoyed working as a software engineer in the industry, being a graduate student at University of Maryland at College Park (Management Information Systems). Her main research interests are empirical studies on software engineering education, software inspections and assurance for high quality software.

Jeffrey C. Carver earned his PhD degree in Computer Science from the University of Maryland. He is an associate professor in the Department of Computer Science at the University of Alabama. His main research interests include empirical software engineering, software quality, software engineering for computational science and engineering, code clones, software architecture, human factors in software engineering and software process improvement. He is a Senior Member of the IEEE Computer Society and a Senior Member of the ACM. Contact him at carver@cs.ua.edu.