Test-case Driven versus Checklist-based Inspections of Software Requirements – An Experimental Evaluation

Nina D. Fogelström and Tony Gorschek
Department of Systems and Software Engineering,
Blekinge Institute of Technology, PO Box 520, S-372 25 Ronneby, Sweden
Phone: +46 457 385000
nina.dzamashvili@bth.se, tony.gorschek@bth.se

Abstract

Software inspections have proved to be an effective means to find faults in different software artifacts, and the application of software inspections on requirements specifications is believed to give a high return on investment as problems are caught early. However, despite the existing evidence of positive effects requirements inspections are not a common practice in industry. The reason is believed to be the cost associated with inspections as a technology. This paper presents an evaluation of test-case driven inspections (TCD) - an emerging inspection technique that aims to cut costs associated with traditional requirements inspections. To formally test the efficiency and effectiveness of TCD inspections an experiment was conducted, in a controlled environment, where checklist based inspections was used as a point of reference. The experiment results indicate that TCD inspections perform better when it comes to effectiveness in finding major faults in a requirements specification.

Keywords: *Software inspections, software requirements, market-driven requirements engineering, experimental evaluation.*

1. Introduction

Software inspections are a well known technique for finding faults in different software artifacts by means of visual examination [1]. Obvious benefits of inspection as a technology are the possibilities of uncovering defects at an early stage of the development process, as early as during initial specification of requirements. This is beneficial as faults in requirements specifications are considered to be very costly since they many times impact downstream efforts as defects are transferred to the design and implementation phases [2-4]. This is probably why companies with limited resources are recommended to prioritize the inspection of requirements over other artifacts like for example code inspections [8, 9].

Despite the recommendations, and experience reports showing positive effects of inspection as technology, inspections are not as common in industry as one might expect. According to Ciolkowski *et al.* only about 40% of companies perform reviews on any development artifact [5]. The number of companies practicing requirements inspections was reported to be even lower. An explanation to this can be costs associated with inspections in terms of project lead time and recourses [10-12], in combination with the difficulty to quantify the return on investment when using inspections.

The increased complexity of market-driven product development demands further development of inspection technology as the handling of a continuous flow of requirements in an engineering effort generates large amounts of requirements from multiple sources threatening to overload companies [22-28]. In this situation, the use of inspection technology at an early stage can help guarantee requirements quality good enough for the purposes of requirements triage and selection, if costs and time aspects can be controlled [29].

Since the introduction of the traditional Fagan inspection in 1972, subsequent expansions and refinements, through the introduction of e.g. streamlined variants (see e.g. two-person inspection), and reading techniques like checklist based reading (CBR) and perspective based reading (PBR), were aimed at decreasing cost and time as well as increasing effectiveness and efficiency of the technology [1]. This paper continues on this theme, in attempt to further add to the understanding of the inspection technology by introducing and testing the efficiency and effectiveness of an inspection technology "variant" called Test-case Driven requirements inspection (TCD). The motivation for TCD inspections was identified in industry where companies faced large amounts of requirements and limited resources as time-to-market was a predominant factor, this was formerly reported by Gorschek and

Dzamashvili-Fogelström [29]. In an attempt to further investigate potential benefits, but also drawbacks with TCD inspections in a controlled setting minimizing confounding factors, this paper presents a formal experiment designed to benchmark TCD inspections against the well-established and known reading technique of CBR.

The paper is structured as follows. Section 2 presents the origin of the TDC inspection technology and related techniques. Section 3 provides the research questions, the hypothesis and the study design. Experiment results and analysis is found in Section 4 and finally Section 5 presents discussion and conclusions.

2. Background and Related Work

TCD inspections were initially piloted to support product management activities at Danaher Motion Särö AB (DHR). DHR operates in a market-driven development environment where large amounts of requirements need to be handled and inspected to be good-enough as decision support material for triage and selection (including initial risk analysis and estimation activities). Requirements arrive from the different sources resulting in not only large quantities of requirements, but also in requirements of vastly varying quality and refinement, so called pre-project requirements. These requirements are the main decision support material for product management and the basis for the identification of relevant batches of requirements allocated to development projects. Thus the need for having high quality pre- project requirements is obvious as requirements of low-quality may impede product management in taking crucial decisions.

From this perspective any introduced inspection technology needs to be low cost, in addition to effective. Low cost in relation to both sheer requirements volume, but also in relation to the early nature of the artifacts being inspected. Pre-project requirements have not yet been chosen for inclusion in the product being developed, thus effort going in to an inspection may potentially be wasted if the requirements in question are later rejected. On the other hand, the term “waste” is used with reservation as an improvement in a requirements quality through inspection may give a manager better decision support to dismiss an unwanted requirements early in the process.

The following sections will provide a brief summary of the TCD inspection process and an overview of the related inspection techniques. A more detailed description of TCD inspections can be found in [29].

2.1. TCD Inspections

The goal of TCD inspections is to enable software companies perform effective requirements inspections but at the same time minimize the cost. In order to achieve this TCD inspections rest on the foundation of the thoroughly tested and successful reading technique of perspective based reading [14, 15], but at the same time TCD inspections introduce a new way of applying the technique, focusing on minimizing involved recourses and maximizing the reuse of the involved personnel and produced artifacts.

Team Size and Involved Roles: The minimal size of the TCD inspection team is two persons, and company resources with key knowledge are involved, namely the product manager (writer/owner in this case) and the test engineer. The small size of the team offers obvious savings, but on top of this, by involving experienced personal no training is required (the tester is already well versed in writing test-cases). Both product manager and test engineer are considered to be system experts. It has been reported that two-person inspections are as effective as traditional inspections consisting of the larger team, especially if the persons involved in the inspection consist of an expert-pair [10]. This indicates that smaller team size used by TCD inspections should not influence the defect detection ability negatively; however it should decrease the cost.

Active Reading Technique: The test engineer inspects requirements by means of producing high level test-cases, which can be compared to perspective based reading (PBR). At the same time as being effective the traditional PBR inspections are rather costly since it requires that e.g. the requirements should be inspected from different perspectives. TCD inspections offer lower cost by utilizing only a tester’s perspective. It can be claimed that loosing other perspectives (i.e. not utilizing several perspectives) will adversely effect the defect detection efficiency; however studies at NASA [14] report that the defect detection rate of applying a single perspective was superior to other reading techniques. This can indicate that applying only one perspective when inspecting e.g. a requirement can still give better results than traditional reading techniques such as ad-hoc or check-list based reading. It is worth to mention that some researchers report positive results from industrial evaluation of using only tester’s perspective of PBR on requirements documents [16].

Reuse of the Staff and Artifacts: One of the main ideas behind TCD inspections is reuse. The inspection is performed by a test engineer who will have to read the requirements document and produce test-cases independent of the inspection. So why not reuse this recourse? The artifacts produced during inspection, namely test-cases are attached to the requirements and should be used in the design and implementation phases as additional requirements documentation. The test-cases are intended to create a basis for the testing activities as well.

TCD inspection process closely resembles the original Fagan's inspections and consists of the following steps:

Planning and Initiation: The inspection process is planned and initiated by the Product Manager (PM) who is also the owner of the inspected requirements. This step involves PM selecting a collection of pre-project requirements that should be inspected, and allocating recourses to the inspection process. The test engineer(s) reviewing requirements is a main recourse required by the process.

Defect Detection: The test engineer(s) examine the requirements. The reading technique used is similar to active reading as high level test-cases are created as a result of the inspection. The goal is to check the following attributes of software requirements: Testability, Completeness, and Conflicts between the requirements. During the course of the inspection both test-cases are created and identified defects are documented.

Inspection Meeting: The PM and test engineer review the inspection protocol and test-cases in order to reach agreement as to what needs to be corrected.

Defect Correction: PM corrects the defects agreed on.

Test-Case Completion: The corrected requirements specification is delivered to the test engineer, who confirms the correction and ensures that created test-cases are up to date. If no new faults are discovered during this process the inspection is considered complete. The test-cases are attached to the corresponding requirements for the future use in design and implementation phases (gives additional detail and information to e.g. developers).

2.2. Related Inspection Methods and Reading Techniques

TCD inspections make use of the ideas behind Perspective Based Reading, Two-Person Inspections and test driven development. This section also gives a brief introduction to Checklist Based Reading, which is a commonly used technique for requirements inspections in industry [13] and the technique which TCD inspection is tested against.

Perspective Based Reading (PBR): When inspecting a document using PBR the reviewers adopt a certain perspective, for example designers, testers or a system user's perspective. Inspections are performed by actively examining artifacts, for example when inspecting requirements specifications persons working from the designer perspective create high level design, user perspective produces users manual and test perspective creates test cases [14]. The idea is that by adopting a certain perspective the reviewers focus on issues relevant for this perspective which can increase the likelihood of finding defects [15]. Reviewers using different perspective find faults that are relevant for the specific view that they adopt. For example tester perspective may find faults related to testability of the requirement, user perspective may find missing requirements and designer perspective incomplete or conflicting requirements. It is believed that combining of the different views should provide a better coverage. The studies performed on PBR have shown that PBR may require more effort compared to less structured approaches, but at the same time it increases the defect detection ability [15]. In a typical case developers perform PBR adopting the different views.

Two-Person Inspections: This inspection method was designed for small to medium sized companies with limited resources in mind. The inspection team in this particular inspection method consists of only two persons, an author and a reviewer. The inspection process steps closely follow the original Fagan's inspections, the obvious difference from the original technique being the team size [1].

Checklist Based Reading (CBR): In CBR the artifact is inspected by means of using a pre-defined list of questions – a checklist. This technique is more structured and is believed to offer more support to the reviewer compared to for example ad-hoc reading [1]. Because of its simplicity this method is believed to be one of the most accepted inspection methods in the industry [13].

3. Experiment Planning and Operation

This section presents major steps in the experiment planning such as defining experiment context, formulating hypothesis, finding the suitable experiment design and evaluation of the possible threats to the validity. The details of experiment operation are found in the end of the section.

3.1 Experiment Purpose and Hypothesis

The goal of the experiment is to evaluate the effectiveness and the efficiency of Test-case Driven Inspections (TCD) when it comes to finding the major faults in a requirements specification. The evaluation is performed by means of comparing TCD inspections to the well tested and most commonly used technique Checklist Based Reading (CBR). The null and alternative hypotheses focus on measures of inspection effectiveness and efficiency and are formulated as follows:

- **$H_{0\ Efficiency}$:** There is no difference in *Efficiency of finding the major faults in a requirements specification* between the subjects applying TCD inspections and the subjects applying CBR inspections.
- **$H_{0\ Effectiveness}$:** There is no difference in *Effectiveness of finding the major faults in a requirements specification* between the subjects applying TCD inspections and the subjects applying CBR inspections.
- **$H_{a\ Efficiency}$:** There is a difference in *Efficiency of finding the major faults in a requirements specification* between the subjects applying TCD inspections and the subjects applying CBR inspections.
- **$H_{a\ Effectiveness}$:** There is a difference in *Effectiveness of finding the major faults in a requirements specification* between the subjects applying TCD inspections and the subjects applying CBR inspections.

In particular the experiment investigates if TCD inspections are cost effective by measuring the time it takes to conduct the inspection, and the number of the major faults that the reviewers detect within that time. Effectiveness of the inspection technique is defined as fault finding rate and is calculated by dividing the number of found faults with the total number of existing faults in the inspected requirement documents. The efficiency is defined as the number of found faults per hour. It is worth mentioning that effectiveness and efficiency of the inspection technique is measured in a similar way in number of other studies on software inspections [20, 21].

Major faults are defined as faults that will have potential effect on the system. These are faults like conflicting requirements, missing requirements, missing or wrong information in the requirements and so on. Defects like spelling errors, wording and incorrect usage of terminology are not considered as major faults.

3.2 Experiment Subjects

The subjects of the experiment were 21 master level students attending the Software Engineering education program at Blekinge Institute of Technology, Sweden. This group mostly consisted of Swedish students with a similar background since all of them had completed their bachelor level studies in the area of Software Engineering/Computer Science at Blekinge Institute of Technology. In the group there were 7 students with international background, who had completed their bachelor studies in the areas of Software Engineering/Computer Science at other universities (India, Pakistan, Germany and Belgium).

All students with international background had prior experience of working in the software industry as software engineers. The Swedish students had experience from software engineering projects in terms of project courses which are run in tight cooperation with industry and very much resemble real development in industry. During these project courses the students are trained to solve typical problems that occur in real-life software projects. To solve these problems students are required to have both advanced technical and managerial skills. Since the project courses simulate the projects run in industry the students who complete these courses are considered to be similar to fresh software engineers working in industry.

At the time of the experiment most of the students had completed the courses in Software Requirements Engineering and Software Verification and Validation, thus the students were assumed to be rather familiar with requirements engineering and software inspections concepts. The results of the post-test showed that the subjects considered themselves to have novice to moderate skills in software inspections and moderate skills in software testing. In addition the subjects had assessed their knowledge in requirements engineering to be from moderate to skilled. The scale used to assess subjects' knowledge and skills in a specific area was of ordinal character. The following categories were used: None, Novice, Moderate, Skilled and Expert.

3.3 Experiment Artifacts

The experiment instrumentation consisted of two requirements specifications, one for an Automatic Teller Machine (ATM) and the other for an Online Web Shop (OWS).

The requirements specification for the ATM contained 13 requirements detailing functionality like card support and validation, pin-check, money

withdrawal, account information handling and so on. The document size was four A4 pages. The ATM specification contained in total 17 faults of which 10 were seeded by the researchers and 7 uncovered by the reviewers when performing the inspection (and categorized as faults by the researchers post-experiment). The requirements specification for the OWS consisted of 16 requirements describing functionality for user log-in, product search, product viewing and purchase, and so on. The document size was four A4 pages. The OWS requirements specification contained in total 19 faults of which 8 were seeded by the researchers and 11 were uncovered by the reviewers.

Both specifications contained functional and non-functional requirements. In addition both documents contained a similar amount of spelling and grammatical errors (nine errors each). The faults in the requirements specifications that were not spelling errors were classified as follows:

- **Missing Requirement (MR):** Applies when a requirement that is needed is missing from requirements specification.
- **Missing Data (MD):** Describes incomplete requirements that lack certain critical information.
- **Unverifiable (UV):** Describes requirement that is not possible to verify or test.
- **Unclear (U):** Applies to requirements that are specified in an ambiguous manner and thus can be interpreted in multiple ways.
- **Requirement Containing Wrong Data (W):** When provided data or information in the requirement is wrong.
- **Requirement Conflict (C):** When information specified in one requirement conflicts with the other requirement(s) in the specification.

All of the listed fault types are considered to have medium to high system impact and therefore considered important in terms of being uncovered by the requirements inspections. The distribution of different types of faults for each document is shown in Table 3.1.

Table 3.1 Fault distribution in requirements specifications.

Document/Fault Type	ATM	OWS
MR	2	3
MD	6	8
UV	2	2
W	2	2
C	2	2
U	3	2
Total:	***17***	***19***

As can be observed in Table 3.1 the distribution of the different fault types is similar for the two systems. Both of the systems contain relatively large amount of missing data faults whereas other types of faults are evenly distributed between the systems.

Other than the requirements specifications the experiment instrumentation also consisted of forms that assisted the reviewers to document discovered faults, forms for documenting produced test-cases, and a requirements inspection checklist (used during the CBR inspections). The checklist consisted of nine questions, of which eight specifically targeted fault types presented in Table 3.1, and one question targeted spelling and grammar errors.

3.4 Experiment Design

The experiment was run at Blekinge Institute of Technology. The reviewers were divided into two groups according to alphabetically ordered list of reviewers' sir names. Reviewers with names starting with the letters A through J formed Group 1 (resulting in 10 persons, including 3 international students), the rest formed Group 2 (11 persons, including 4 international students). In general no correlation can be seen between academic or professional performance and first letter in sir name, therefore the division is considered to be random.

The experiment is of type blocked subject-object study [17] where each group performs both TCD and CBR inspections, but on different requirements specification documents, as shown in Table 3.2.

Table 3.2 Experiment design – sessions and groups.

	Session1	Session2
Group1	CBR_{OWS}	TCD_{ATM}
Group2	TCD_{ATM}	CBR_{OWS}

As can be seen in Table 3.2 the reviewers in Group 1 use CBR on the requirements specification for OWS first, and then apply TCD on the requirements specification for the ATM system. The subjects in Group 2 apply the treatments in the opposite order, i.e. first TCD and then CBR.

Independent of chosen experimental design there exist advantages and shortcomings. Potential advantages of the selected design are:

- The treatments are applied in different order which makes it possible to control the order effects between the treatments. Order effect implies the situation when the performance of

the subject is effected by the order in which the treatments are applied. For example if both of the groups would first apply the TCD inspection and then CBR there would be no way to check if the results of CBR inspections were influenced by what the subjects learned from the TCD inspections.

- The subjects apply treatments on different systems removing the effect of subjects getting familiar with the system and thus performing better when applying the second technique, this is also called learning effect.

Potential drawbacks of the chosen design are:

- The introduction of the requirements specification documents for two different systems may influence the results of the experiment unless some measures are not taken to ensure that the two systems are similar, and that none of the specifications is particularly suited for one of the tested inspection methods, or a specific subject group. This issue is further discussed in the validity evaluation (see Section 3.5).
- The design of the experiment limits researcher in investigating possible interaction effects between the applied inspection techniques and the inspected documents, as each of the teams perform TCD inspections on the ATM, and CBR inspections on the OWS. Interaction effects imply the situation when the measured effect of the treatment is influenced by some other variable. In this case the interaction effect between the inspections technique and the inspected documents would occur if the performance of the inspection technique could to some extent be explained by the document that the technique was applied to. However, this is not considered as a serious problem since the requirements specifications were initially designed to avoid this kind of interaction (see Internal Validity in Section 3.5).

3.5 Validity Evaluation

The validity of the experiment is evaluated by using four common perspectives, presented by Wohlin et al. in [17], namely Conclusion Validity, Internal Validity, Construct Validity and External Validity.

Conclusion Validity: Conclusion validity checks if the findings of the study are correct by evaluating how the data analysis was performed and how the appropriate statistical tests were selected. The threats to conclusion validity are considered to be small for this study. The applied measures and data collection process are well tested in other studies comparing different inspection techniques [20, 21], and robust statistical tests are used. In addition, all subjects received the same training level and instructions on how to proceed with the inspection process.

Internal Validity: Internal validity investigates if the observed relation between the treatment and outcome is really caused by the treatment or if there are other factors that the researcher may not be aware of. The common threats associated with internal validity are: History, Maturation, Instrumentation, Mortality and Social threats.

History and Maturation threats are reduced by scheduling the experiment on the same day and by using different requirements document for each treatment. However it can still happen that participants mature during the study. It is possible that the group applying the TCD inspections first gets inspired and applies the TCD thinking to CBR documents and the vice-versa. This risk is reduced by making sure that the subjects in Group 1 and Group 2 apply the treatments in different order.

Experiment instrumentation may pose a threat since the experiment is conducted on requirements specifications of two different systems. This provides certain advantages but can also mean a threat, i.e. interaction between a certain requirement document and the applied treatment or group of subjects. In order to minimize these risks the requirements specifications were designed to have similar size, level of complexity, and structure. In order to avoid interaction with the tested inspection techniques the same type of faults were seeded in both documents. The distribution of different types of faults is similar in both of the documents as shown in the Section 3.3. The subjects are expected to be familiar with both of the systems since most of the students have experiences of shopping online (OWS), and in using ATM machines. Additionally number of independent experts have evaluated size, level of complexity and the structure of the requirements specifications for the ATM and OWS systems and did not find major differences between them.

Mortality and social threats like compensatory rivalry and resentful demoralization do not apply since each subject applies both methods and none of the subjects have dropped out of the experiment.

Construct Validity: The experiment goal is well defined, the treatments relate well to the software inspections and the collected measures of efficiency and effectiveness should be able to correctly represent the effect construct. The material used in the experiment such as data collection forms as well as measured variables are the same as the ones used in the series of the inspection experiments conducted by other

researchers, thus they should be reliable [20,21]. The identified threats are mostly social in nature. Hypothesis guessing may occur, since there is a risk that the students may try to guess the expected outcome of the experiment and act on it. The second identified threat is experimenter expectancies. This threat is targeted by involving an independent researcher in the data analysis process.

External Validity: External validity is mainly concerned if the study findings can be generalized. Concerning the external validity, two threats are predominant in the case of the experiment presented in this paper. First, the interaction of the selection and treatment, i.e. there is a threat of using the students as the representatives of the professionals working in the industry. Second, interaction of the setting and the treatment, i.e. there is a threat that the size and complexity of the requirements document used in the experiment will not be representative of the requirements specifications used in industry. The first threat is addressed by allowing only master level students to participate in the study. Most of the master level students can be compared to the fresh software engineers in industry [20, 21]. Still it can be argued that the experience and the motivation of the professionals in industry are superior to that of students participating in the experiment. However, more experience and stronger motivation should not necessarily create difficulties in generalizing the results of the experiment. On the contrary it could even strengthen any positive effects detected in the study. The second threat is addressed by finding a reasonable size for the requirements specifications used in the experiment considering restrictions put forward by the time budget of the experiment. To our best knowledge the effects identified in this study should be applicable to larger documents as well. The increased size and complexity of the requirements document may increase the time it takes to perform the inspections but then this should be true for both of the studied inspection techniques.

3.6 Experiment Operation (Execution)

The experiment was conducted at Blekinge Institute of Technology. It consisted of two parallel sessions with one hour lunch brake in-between. The duration of each session was maximum three hours. The experiment execution closely followed the design presented in Table 3.2, during the first session between 9.00 and 12.00 Group 1 applied CBR inspections on the OWS system, and Group 2 applied TCD the ATM system. In the second session, between 13.00 and 16.00 the groups applied the techniques in the opposite order and on the requirements specification that they were not familiar with. For each session the groups were seated in separate rooms.

In the beginning of each session both of the groups received a brief introduction to the relevant inspections technique, and instructions on how use the supporting documentation, i.e. documenting faults.

Once the introduction was over the subjects applied the assigned technique without cooperating with fellow reviewers. During the course of the experiment it was allowed to take brakes as necessary, however it was made clear that the inspection technique, faults or the inspected document was not to be discussed until the experiment was completed. The inspection was concluded as a subject considered to be finished with the review or when the allocate time for the session was up. It is worth mentioning that most of the subjects used less than the allocated time to complete the inspection process.

4. Results and Analysis

This section shows the experiment results as well as analysis following the results. The initial step in the data analysis effort was to study the information collected in the inspection records. The inspection records provided several data items, i.e. data on the time when the inspection session started and finished as well as the time when a certain fault was found, description of each identified fault, and fault location in the requirements specification. The examination of the inspection records showed that there were no missing data points and all of the experiment material was fully usable.

In order to find values for inspection effectiveness and efficiency for each subject the number of identified faults and total time of the inspection was collected. Each reported fault was evaluated by the researchers in order to make sure that it was not a false positive. As a result of this process the reported faults were divided into the following classes:

- **False positives:** Reported faults that did not qualify as a fault in relation to the inspected requirements specification were assigned to this class.
- **Minor faults:** Faults like spelling and wording errors, plus incorrect usage of terminology make up this class.
- **Major faults:** Faults that are identified as being one of the fault types described in Table 3.1 are included in this class. In other words, this class describes faults like missing requirements, missing data in the

requirements, conflicting and unverifiable requirements and so on.

Effectiveness of the inspection technique is measured by finding a ratio of total number of identified major faults and total number of existing faults in the inspected requirements specification.

Efficiency describes the number of found faults per hour spent inspecting and is measured by finding ratio of total number of identified major faults and total time spent on the inspection process. Please note that the false positives and minor faults are excluded from the calculations of effectiveness and efficiency.

The power of the investigated inspection techniques are analyzed by comparing the mean values of effectiveness and efficiency for each student group. The significance level for rejecting the null hypothesis is set to 0.05. The applied Shapiro-Wilk normality test could not prove that the studied variables were not normally distributed (the test statistic for the variables of effectiveness and efficiency have showed to be larger than 0.05, whereas values less than 0.05 would have indicated not normally distributed data set). Therefore we could use the parametric T-test [17-19] for matched pairs of data to test the significance of the difference between the calculated mean values of effectiveness and efficiency of the studies inspection techniques.

The reminder of the section is organized as follows: Section 4.1 presents average inspection time for each group and inspection technique as well as the count of all reported faults including false positives and minor faults. Section 4.2 gives details on how false positives, minor faults and major faults are distributed in the reported faults. Finally, Section 4.3 presents the mean values for effectiveness and efficiency for each inspection technique and student group.

4.1 Inspection Time and Reported Faults

The values for Inspection Time and the number of Reported Faults are important as they affect inspection effectiveness and efficiency. In order to understand how these values differ for TCD and CBR inspections we chose to calculate and compare mean values for Inspection Time and Reported Faults for each subject group and applied inspection technique. Mean Inspection Time is calculated by taking an average value of individual inspection times in each group. Mean of Reported Faults is found by taking a mean of total faults reported by each individual in a group. Table 4.1 shows the result of the calculations. Inspection time is measured in minutes and the Reported Faults represents a count of faults identified by the reviewers. Each cell of the table presents the calculated mean value and corresponding measure for the standard deviation.

Table 4.1 Inspection time and reported faults.

	Group 1		**Group 2**	
	CBR	***TCD***	***CBR***	***TCD***
Inspection Time (Minutes)	Mean: 92.5 St.D: 26.2	Mean: 93.3 St.D: 26.5	Mean: 53.7 St.D: 14.6	Mean: 97.0 St.D:19.5
Reported Faults (Count)	Mean: 18.3 St.D: 4.9	Mean: 13.7 St.D: 4.1	Mean: 13.1 St.D: 4.5	Mean: 11.1 St.D: 4.4

At a first glance CBR seems to yield more faults found in less time than TCD. For example Group 1 uses on average almost the same amount of time to perform inspections, but finds approximately 28% more faults. The results of Group 2 show that reviewers using CBR spend on average 47 min less time and find approximately 18% more faults.

However it is important to note that calculations for the variable Reported Faults presented in Table 4.1 do not make difference between the classes of the reported faults. Thus all faults that are found by the reviewers are taken into account. This means that before making any claims of superiority of one or the other inspection technique it should be accounted how many of the reported faults are actually faults, how many are minor faults and how many are false positives.

In addition, comparing the mean values of reported faults can also be misleading unless these values are matched against the total number of the existing faults in the reviewed requirements documents. Thus it is more interesting to investigate the measure of inspection effectiveness which calculates the ratio between the identified and existing faults rather than concentrating on the number of identified defects only. Section 4.2 therefore presents the result of classifying the identified faults in classes such as Major Faults, Minor Faults and False Positives. Calculation of effectiveness and efficiency of each studied inspection technique based on the number of identified major faults is found is found in Section 4.3.

4.2 Fault Distribution

Table 4.2 presents the distribution of identified faults with respect to fault class (see Section 4). The values describe the percentage of identified False

Positives, Minor Faults and Major Faults per group and applied inspection technique.

Table 4.2 Distribution of reported faults (in %).

	Group 1		Group 2	
	CBR	***TCD***	***CBR***	***TCD***
False Positives	62.8	51.1	62.1	42.3
Minor Faults	8.7	0.7	9.0	9.8
Major Faults	28.4	48.2	29.0	48.0

According to Table 4.2 when applying TCD inspections both of the groups identify considerably less amount of false positives (e.g. Group 1 identifies 11.7 percent units less false positives using TCD than when using CBR and for Group 2 the number is 19.8 percent units). In addition, the identification of major faults is almost double when using TCD over CBR for both Group 1 and 2. The identification of minor faults is somewhat inconclusive. Looking at the results for Group 1 and CBR identified almost ten times as many minor faults as when using TCD. This result is somewhat contradicted by Group 2 where the fault identification for the minor defect class is rather homogenous. Analysing the results of Table 4.2 one can conclude that superior result of the CBR inspection presented in Table 4.1 can to the large extent be explained by the high amount of false positives found by the reviewers applying the technique. However once the false positives are removed from the calculations the results of CBR inspections are not as impressive. According to table 4.2 when it comes to major faults the reviewers using CBR inspections find fewer amounts of major faults compared to TCD inspections. Calculations of effectiveness and efficiency of finding the major faults in the requirements specifications is presented in the following section.

4.3 Effectiveness and Efficiency

The calculations of effectiveness and efficiency are based on the number of reported major faults. False positives and minor faults are excluded from the calculations. Table 4.3 presents the mean effectiveness and the mean efficiency for each group and each inspection technique together with the associated values for the standard deviation.

The mean effectiveness for a group that is applying a specific inspection technique is found by calculating the effectiveness of each reviewer in a group and then taking the mean of the individual effectiveness values. The effectiveness of an individual reviewer is calculated using the following formula: Effectiveness subject = ("Total number of major faults that the subject has reported" / "Total number of existing faults in the inspected requirements specification"). The mean efficiency for a group was calculated in the same manner, i.e. by finding the mean value of individual efficiencies of the subjects in a group. The efficiency of the individual reviewer shows how many major faults that are found by this reviewer per hour of inspection. The efficiency of an individual reviewer is calculated as follows: Efficiency subject = ("Total number of major faults that the subject has reported" / "Total time that the subject spent on the inspection").

Table 4.3 Effectiveness and Efficiency.

	Group 1		Group 2	
	CBR	***TCD***	***CBR***	***TCD***
Effectiveness (fault finding rate)	Mean: 0.27 St.D: 0.14	Mean: 0.39 St.D: 0.14	Mean: 0.20 St.D: 0.06	Mean: 0.31 St.D: 0.14
Efficiency (number of faults per hour)	Mean: 3.58 St.D: 1.81	Mean: 4.58 St.D: 2.40	Mean: 4.65 St.D: 1.93	Mean: 3.39 St.D: 1.51

Effectiveness: TCD inspections show better effectiveness for both Group 1 (TCD 0.39 vs. CBR 0.27 gives ratio equal to 1.44), and Group 2 (TCD 0.31 vs. CBR 0.20 vs. gives ratio equal to 1.55). In other words Group 1 is 1.44 times more effective when applying TCD and Group 2 is 1.55 times more effective. The statistical tests (T-test for matched pairs) indicate that the results (difference) for the variable effectiveness are significant for both of the groups, since the test statistics for the variable effectiveness are less than 0.05 (the target significance level for the experiment). The test statistic for Group 1 is 0.037 and for Group 2 is 0.046. This result implies that when it comes to effectiveness of finding major faults TCD inspections are superior to CBR.

Efficiency: The results for efficiency differ between the groups. Group 1 is more efficient when using TCD inspections (TCD 4.58 vs. CBR 3.58 gives ratio equal to 1.28). However, Group 2 is more efficient when using CBR inspections (CBR 4.65 vs. TCD 3.39 gives ratio equal to 1.38). This means that Group 1 is 1.28 times more efficient when applying the TCD inspections, whereas Group 2 is 1.38 times more efficient when applying CBR inspections.

The statistical tests (T-test for matched pairs) for efficiency show no significance since test statistics for both of the groups are larger than 0.05, test statistic for Group 1 being 0.160 and for Group 2 0.051. The

obtained result indicates that even though there are differences in inspection effectiveness between TCD and CBR inspections it can not be proven that one technique is finding a larger amount of major faults per hour than the other.

5. Discussion and Conclusions

The focus of the experiment presented in this paper was to evaluate TCD inspections, as a step to validate the technology prior to industry large-scale piloting. This was accomplished through the comparison of TCD inspections with CBR inspections.

At a first glance CBR inspections show a better result, i.e. it takes less time to use compared to TCD inspections, and the number of faults reported is also greater when using CBR. First addressing the time aspect, it is not surprising as TCD inspections use an active reading technique and the reviewers actually produce and document a test case design for the reviewed requirements (the potential benefit and reusability of the TCD artifacts is not the focus of this experiment, although important to remember). In contrast CBR inspections require only a fault report to be created. Second, in terms of faults found, the reviewers using CBR do find more faults, but mainly due to a large amount of false positives. Looking at major faults only the reviewers using TCD inspections was significantly superior to CBR inspections, at the same time fewer false positives were reported by the reviewers using TCD inspections. This result could indicate that when using TCD inspections reviewers apply an active reading technique which allows them to focus on important aspects, as opposed to the CBR inspections which provide general guidelines on what to look for when reading the documents. Several other reports evaluating active reading techniques report similar findings [14, 15].

The statistical evaluation of the experiment results (effectiveness and efficiency) show that the TCD inspections are significantly more effective compared to CBR inspections. When it comes to efficiency the experiment results are conflicting (Group 1 showing better result for TCD inspections and Group 2 for CBR inspections). The difference in the efficiency however can not be said to be significant for any of the groups. From this perspective we can dismiss null hypothesis **H0 Effectiveness** in favor of our hypothesis that TCD inspections are more effective than CBR inspections (**Ha Effectiveness**), however **H0 Efficiency** can not be dismissed. In other words, this indicates that TCD inspections are more effective at finding major faults in the requirements specifications while at the same time not necessarily taking more time compared to CBR inspection even if a test case design is created simultaneously. This clearly speaks in favor of TCD inspections as cost effective even if the value of reusing the produced test design can not be quantified based on the experiment.

A potential threat to the results obtained through the experiment is a possible object-treatment interaction, since CBR inspections were applied on one system and TCD inspections on the other. This may imply that the results observed are dependent on the requirements documents themselves and not on the tested technology (see Section 3.5). During the design of the experiment measures were taken to reduce this risk, however it is difficult to measure if the risk was completely removed.

With regards to the usage of students as reviewers in the experiment we do recognize the potential threat in terms of drawing conclusions about effectiveness and efficiency of TCD inspections in industry. However, we feel that professionals in industry should haven an even greater potential of utilizing TCD inspections as their level of motivation, knowledge in creating test designs, and domain knowledge should be at least if not superior to the students used in the experiment.

The next steps in evaluating TCD inspections have two main parts. First, is to conduct additional experiment that will replicate the one presented in this paper (although using more subjects and only one requirements specification) in order to seek confirmation of the results obtained. The second part revolves around testing the usability and usefulness of the artifacts produces (test designs) which was a spin-off effect of using TCD inspections. Pending these results industry piloting will commence.

6. References

[1] A. Aurum, H. Petersson, and C. Wohlin, "State-of-the-Art: Software Inspections after 25 Years," Software Testing Verification and Reliability, vol. 12, pp. 93-122, 2002.

[2] L. Karlsson, Å. Dahlstedt, J. Natt och Dag, B. Regnell, and A. Persson, "Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study," presented at Proceedings of the Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02), Essen, Germany, 2003.

[3] G. Kotonya and I. Sommerville, Requirements engineering: processes and techniques. New York: John Wiley, 1998.

[4] I. Sommerville, Software Engineering, 6 ed. Essex: Addison-Wesley, 2001.

[5] M. Ciolkowski, O. Laitenberger, and S. Biffl, "Software reviews: The state of the practice," IEEE Software, vol. 20, pp. 46-51, 2003.
[6] CMMI-PDT, "Capability Maturity Model Integration (CMMI), Version 1.1," in CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing Version 1.1 (CMMI-SE/SW/IPPD/SS, V1.1). Pittsburgh, 2002.
[7] ISO/IEC, "Software Process Assessment TR 15504:1998," vol. 2004. http://www.sei.cmu.edu/iso-15504/: ISO/IEC, 1998.
[8] O. Laitenberger, T. Beil, and T. Schwinn, "An industrial case study to examine a non-traditional inspection implementation for requirements specifications," presented at Proceedings of the Eighth IEEE Symposium on Software Metrics, Los Alamitos CA, 2002.
[9] S. R. Rakitin, Software Verification and Validation for Practitioners and Managers, 2. ed. Boston MA: Artech House, 2001.
[10] C. Sauer, D. R. Jeffery, L. Land, and P. Yetton, "The effectiveness of software development technical reviews: A behaviorally motivated program of research," IEEE Transactions on Software Engineering, vol. 26, pp. 1-14, 2000.
[11] L. G. J. Votta, "Does every inspection need a meeting?," presented at Proceedings of the 1st ACM SIGSOFT Symposium on Foundations of Software Engineering, New York, 1993.
[12] A. A. Porter, L. G. J. Votta, and V. R. Basili, "Comparing detection methods for software requirements inspections: A replicated experiment," IEEE Transactions on Software Engineering, vol. 21, pp. 563-576, 1995.
[13] Laitenberger O, DeBaud J-M, "An encompassing life-cycle centric survey of software inspection," Journal of Systems and Software, vol. 50, pp. 5-31, 2000.
[14] V. R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sorumgard, and M. V. Zelkowitz, "The empirical investigation of perspective-based reading," Empirical Software Engineering - An International Journal, vol. 1, 1996.
[15] F. Shull, I. Rus, and V. Basili, "How perspective-based reading can improve requirements inspections," Computer, vol. 33, pp. 73-79, 2000.
[16] T. Thelin and T. Berling, "A Case Study of Reading Techniques in a Software Company" presented at Proceedings of the ISESE'04, International Symposium on Empirical Software Engineering, IEEE 2004.
[17] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslen, Experimentation In Software Engineering, Kluwer Academic Publishers, 2000.
[18] Frederick J. Gravetter, Larry B. Wallnau, Essentials of Statistics for the Behavioral Sciences, Wadsworth Group 2002.
[19] Mario F. Triola, Elementary statistics – International edition, Pearson Education 2004.
[20] C.Andersson, "Exploring the Software Verification and Validation Process with Focus on Efficient Fault Detection," in Lund Institute of Technology - Department of Communication Systems. Lund: Lund University, 2003.
[21] T.Thelin, "Empirical Evaluations of Usage-Based Reading and Fault Content Estimation for Software Inspections," in Lund Institute of Technology - Department of Communication Systems. Lund: Lund University, 2002.
[22] Wieringa R, Ebert C (2004) Guest Editors' Introduction: RE'03: Practi-cal Requirements Engineering Solutions. IEEE Software 21(2):16-18.
[23] Karlsson L, Dahlstedt Å, Natt och Dag J, Regnell B, Persson A (2003) Challenges in Market-Driven Requirements Engineering - an Indus-trial Interview Study. In Proceedings of the Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02), Universität Duisburg-Essen, Essen, Germany, pp. 101-112.
[24] Kotler P, Armstrong G (2001) Principles of Marketing. Prentice Hall, Upper Saddle River NJ.
[25] Gorschek T, Davis A (2005) Assessing the Quality of Requirements Process Changes. In Proceedings of the Eleventh International Work-shop on Requirements Engineering: Foundation for Software Quality (REFSQ'05), Universität Duisburg-Essen, Porto, Portugal. Download at: http://www.bth.se/fou/Forskinfo.nsf/ . pp. 101-112.
[26] Lehmann DR, Winer RS (2002) Product Management. McGraw-Hill, Boston MA.
[27] Mintzberg H, Ahlstrand BW, Lampel J (1998) Strategy Safari : A Guided Tour through the Wilds of Strategic Management. Free Press, New York NY.
[28] T. Gorschek and C. Wohlin, "Requirements Abstraction Model," *Requirements Engineering journal*, vol. 11, pp. 79-101, 2006.
[29] T. Gorschek and N. Dzamashvili - Fogelström, "Test-case Driven Inspection of Pre-project Requirements - Process Proposal and Industry Experience Report," presented at the Requirements Engineering Decision Support Workshop held in Conjunction with the 13th IEEE International Conference on Requirements Engineering, Paris, 2005.