C. Rombach, O. Kude, A. Aurum, R. Jeffery and C. Wohlin, "An Empirical Study of an ER-Model Inspection Meeting", Proceedings Euromicro Conference, special track on "Software Process and Product Improvement", pp. 308-315, Antalya, Turkey, 2003.

# An Empirical Study of an ER-Model Inspection Meeting

Caroline D. Rombach[1], Oliver Kude [1], Aybüke Aurum[2], Ross Jeffery[3], Claes Wohlin[4]

[1]*Computer Science and Economics, University of Mannheim, Germany*
[2]*School of Information Systems, Technology and Management, Univ. of New South Wales, Australia*
[3]*School of Computer Science and Engineering, Univ. of New South Wales, Australia,*
*National ICT, Australia*
[4]*Department of Software Engineering and Computer Science, Blekinge Institute of Technology, Sweden*

## Abstract

*A great benefit of software inspections is that they can be applied at almost any stage of the software development life cycle. This paper documents a large-scale experiment conducted during an Entity Relationship (ER) Model inspection meeting. The experiment was aimed at finding empirically validated answers to the question "which reading technique has a more efficient detection rate when searching for defects in an ER Model". Secondly, the effect of the usage of Roles in a team meeting was also explored. Finally, this research investigated the reviewers' ability to find defects belonging to certain defect categories. The findings showed that the participants using a checklist had a significantly higher detection rate than the Ad Hoc groups. Overall, the groups using Roles had a lower performance than those without Roles. Furthermore, the findings showed that when comparing the groups using Roles to those without Roles, the proportion of syntactic and semantic defects found in the number of overall defects identified did not significantly differ.*

***Keywords:*** *Entity Relationship Model, Requirements Specification Inspection, Reading Techniques, Inspection Teams, Empirical Research*

## 1. Introduction

Entity Relationship (ER) Modeling provides designers with a means for data representation, known as the conceptual model, which is used during the early phases of database design [2]. The purpose of the conceptual model is to represent the informal requirements of an application in terms of a more formal description. The conceptual model gives a detailed overview of the enterprise, and provides a systematically fashioned framework of what the data requirements of the database users are, and how the database will be structured to fulfill these requirements [14].

One of the important tasks in database modeling is to specify how entities and relationships are distinguished in a conceptual model. It is also important to accurately map the business rules to ER-Models so that application software can be produced according to design standards. The designer reviews the ER-Model (a) to confirm that all data requirements are satisfied and are not in conflict with one another, (b) to remove any redundant features.

To ensure that the database design conforms to the design standards, it is necessary to review ER-Models by means of appropriate techniques. In this sense, we believe that software inspection techniques are effective to guarantee the quality of the ER-Models. Reading techniques, which are a series of procedures that a software developer follows when reviewing or inspecting a document, are one way of providing structure to the inspection process. Researchers generally agree that the choice of reading approach may have a potential impact on inspection performance and has become an important factor of a successful implementation when applied to software requirements [5, 7, 8, 11, 13]. Although there is a vast amount of empirical research on software inspection, comparing the different types of reading techniques, to the best of our knowledge, there has been no empirical study conducted in the software engineering literature on ER-Modeling inspection.

In order to investigate the effect of alternative defect detection techniques, namely the Ad Hoc and the Checklist-based reading for ER-Modeling, a large-scale experiment was conducted at the University of New South Wales. Note that, although referred to sometimes as 'Ad Hoc reading', the Ad Hoc approach is not actually a reading technique but an informal and unplanned approach to defect detection with no support. However, we believe that developers will have developed their own approach over the years for defect detection. Thus, even an informal approach, such as Ad Hoc, is subject to a developer's mental model for this task to a certain degree. In order to increase the readability in the paper, we will refer to this approach as Ad Hoc reading. For the Ad Hoc approach limited guidance was provided during the inspection. For the Checklist-based reading, reviewers were provided with a checklist that may be produced automatically from a data dictionary. During the experiment, the team members adopted predefined Roles: Moderator, Recorder or Reader. Note that, the roles mentioned here are 'meeting roles', not the 'individual preparation roles' which are used in Perspective Based Reading (PBR).

The motivation behind this experiment was to find out whether our results would be consistent with earlier experiments that concentrated on similar research questions but dealt with requirements documents rather than an ER-Modeling. Further, it was a main goal to provide statistically validated results to the question of the effectiveness of the Ad Hoc and Checklist-based reading. This was done by analyzing the defect detection rates of the inspection groups. Furthermore, this study concentrates on the change in performance after the participants were advised to use Roles in the team meeting. Can the results validate the arguments for the improvement of the meeting performance by advising the participants to use Roles? This paper also provides quantitative results on a more specific question regarding the ability of reviewers to find defects of a certain category after having classified all existing defects as either syntactic or semantic.

## 2. Defect Detection and Fagan Roles

Since there has been no research on the influence of the Ad Hoc and Checklist-based reading and the usage of Roles on the ER-Model inspection process so far, we have widened our literature review to the inspection of requirements documents overall. By doing so, we will be able to compare our results to those of earlier research.

It is important to have a complete and correct software requirements specification for the development of high quality systems. Although this is acknowledged by developers, the majority of defects occur during the early phases of the software development, and may have serious consequences if they are not identified and corrected in that particular stage of the development process [3].

It has been shown that defect detection techniques have a great influence on the defect detection rate of the requirements specification document. The inspector's ability to find defects varies with the technique used and his/her attention can be directed towards particular inspection targets [1]. Therefore, many researchers have focused on experiments that give statistically validated information on the effectiveness of different reading approaches applied to the requirements inspection. Porter *et al.*, [13] conducted an experiment with 48 graduate students that compared the Scenario-based reading to the approaches of Ad Hoc and Checklist. They concluded that the Scenario approach was superior to both the Ad Hoc and Checklist-based reading, referring to the defect detection rate. Additionally, it was found that the Checklist-based reading was no more effective than the Ad Hoc reading. In a replication of Porter *et al.*, experiment Miller *et al.*, [11] compared Scenario-based reading solely to the Checklist-based reading, finding that the defect detection rate of the Scenario approach was once again superior to that of the Checklist-based reading. On the other hand, Fusaro *et al.*, [6] obtained different results than those of the original experiment by Porter *et al*. They conducted an experiment with 30 undergraduate students and could not find any empirical evidence of better defect detection performance when using Scenarios. In accordance to Fusaro *et al*., and Sandahl *et al*., [15] could not support the superiority of the Scenario method either. In a later replication of the same experiment, Halling *et al.* [7] received results that were quite different from the other studies. Their large-scale experiment (150+ undergraduate students) led to the conclusion that the Checklist-based reading was overall more effective on the individual level, whereas the Scenario approach gained effectiveness when applied to a certain target focus, in their case, specific parts of the document. Finally, Wohlin *et al*., [17] who studied the impact of an individual reviewer on the effectiveness of an inspection team concluded that a good Checklist may indeed be cost effective. In this study, we explored the effects of Ad Hoc and Checklist-based reading on ER-Modeling. The students involved in the experiment had a strong understanding of data modeling techniques. They had also been exposed to both the Ad Hoc and Checklist-based reading before the experiment. Scenarios, on the other hand, were not studied because the students came from different disciplines – thus lacking in sound knowledge and a complete understanding of the software development process and the different roles that can be involved in it.

Although there are only few studies on the effects of Fagan Roles [4] in group meetings, Levine *et a*l., [10] provided evidence that Roles do in fact have a positive impact on the performance of reviewers. Parnas *et al*., [12] also recognized that Roles were a helpful means of improving the defect detection rate, even though the roles that they used defined a perspective from which to read the document and were therefore different from Fagan's 'meeting roles'. The overall idea of assigning Roles to the team members is to improve the results of software inspections. According to [4], the Roles are: Author, Moderator, Reader and Recorder(s). The Author is the person who writes the artifact. The Moderator has to make sure that the group agrees about the found defects. The Reader drives the review process by reading the document and looking for flaws within the ER-Model. The Recorder documents every found defect. Each role requires specific skills and knowledge. In an experiment on the use of procedural Roles in code inspection, a main finding was that procedural Roles made only a limited difference to group performance, even though Land *et al.,* [9] claims their benefits. This experiment also aims at providing further insight into the effectiveness of Roles in team meetings.

## 3. Research Questions

Our experiment investigates three main research questions: Firstly an answer is given to the question of whether the usage of Roles has an effect on the defect detection rate of an ER-Model inspection. Secondly, we look at what kind of effect the structuring of the inspection process by using a Checklist would have? Finally, we address whether the Ad Hoc or Checklist-based reading influences the ratio of semantic/syntactic defects found to the overall number of defects identified by the teams. These three research questions result in three hypotheses:

Q1. H0: Roles will not affect the percentage of defects identified
H1: Roles will affect the percentage of defects identified
Q2. H0: Structuring the inspection process by using a Checklist will not affect the percentage of defects identified
H1: Structuring the inspection process by using a Checklist will affect the percentage of defects identified
Q3. H0: The approach taken has no effect on the quota of semantic/syntactic defects of overall defects found
H1: The approach taken has an effect on the quota of semantic/syntactic defects of overall defects found

## 4. Experimental Design and Implementation

### 4.1. Background

The requirements reviewed in the experiment were provided in the form of an ER-Model. Introduced by Chen [2], the ER-Model has become a standard modeling technique in the database environment. The ER-Model, using varying notations and semantic approaches, has enjoyed a remarkable and increasing popularity in both the research community -the computer science curriculum- and within in industry. In step with the increasing diffusion of relational platforms, ER-Modeling has acquired growing appreciation. The primary representation elements of the ER-Model are its entities, attributes and their interacting relationships. It also includes connectivity and cardinality, which are derived from business rules.

### 4.2. Participants

The experiment was conducted with 303 students who had taken a Database Systems course at the University of New South Wales. The students consisted primarily of a combination of first year Information System and Software Engineering students, second year Computer Science, and Commerce students. The remainder came from different disciplines e.g. law, psychology and civil engineering. The Database Systems course is an introductory course in which students acquire knowledge and exercise skills in a number of data modeling and design techniques. In particular, students are taught relational database design and modeling. At the conceptual level, emphasis is put on the Entity Relationship model.

### 4.3. Design

In a training session subjects were exposed to both Ad Hoc and Checklist-based reading and applied them on several different ER-Models as part of their schooling for the database course. The experiment consisted of two parts. In the first part subjects examined the ER-Model individually using only the checklist approach [16]. The second part of the experiment took place in tutorial sessions. This paper presents and focuses on the data collected in the second part of the experiment i.e. 'inspection meeting', where the subjects studied the ER-Model in groups of three.

Table 1: 2x2 Experimental design

| | | Defect Detection | |
|---|---|---|---|
| | | Ad Hoc | Checklist |
| Roles | No | Ad Hoc without Roles | Checklist without Roles |
| | Yes | Ad Hoc with Roles | Checklist with Roles |

The tutorial sessions were randomly divided into four major groups. These were Ad Hoc without Roles, Ad Hoc with Roles, Checklist without Roles and Checklist with Roles. This led to the 2x2 experimental designs (Table 1). The students were then randomly split into teams of three within their tutorial sessions. In groups using roles, students decided which Roles to adopt themselves. A supervisor (either a Tutor or a Lecturer) was assigned to every group to provide assistance and to make sure that the participants understood their respective tasks. Students from groups with roles were also provided with a description of the task and both written and oral instructions explaining the responsibility of the Roles that they adopted. In the case of groups without roles, only a description of the task was provided to the students. Each group had an approximate time of 50 minutes to complete the inspection meeting. In order to cover the ethical issues, students had been informed about the nature of the exercise as well as the experiment, including what was expected from them, how their anonymity would be ensured, how the information that they provided would be treated in confidence, and so on. They had also been told that the exercise would not form part of their assessment for the course.

## 4.4. Experimental Material

The ER diagram used for the experiment had 7 entities and 6 relationships. The relevant problem was supported by business rules. Each entity had its own attributes, including primary and foreign keys, derived and multi-valued attributes. In addition to this, the connectivity and cardinalities were also displayed. Connectivity is used to describe the relationship classification. The ER diagram indicates connectivity by placing a number near the related entities e.g. 1:1, 1:M. The cardinality expresses the specific number of entity occurrences associated with one occurrence of related entity. The relationships in the ER-Model included mostly many-to-many relationships.

A total amount of 25 defects was embedded into the diagram. Defects were classified as either (a) Syntactic defects, or (b) Semantic defects. Syntactic defects were related to the notation in the ER-Model. Semantic defects were mainly concerned with the business rules. The complete description of the experiment, including experimental material, and the checklist can be obtained from the third author of this paper.

## 4.5. Experimental Implementation

During the inspection meeting, the participants were provided with a sheet with the business rules and the actual ER-Model. By examining the ER-Model and the business rules, they were required to (a) study the ER-Model in terms of its notation, cardinality, connectivity, referential integrity and entity integrity, (b) identify all the defects. Furthermore, a checklist was provided to subjects who were assigned the Checklist-based reading. This checklist was created systematically from the data dictionary of the database, which included all the attribute names and entity names from the ER-Model. For each component on the checklist, subjects were required to indicate in a checkbox the correctness of the component, with a tick, or incorrectness, with a cross.

## 4.6. Threats to Validity

The following four types of validities may be applicable in our experiments [18].

**Conclusion Validity:** This is concerned with the statistical relationship between the treatment and the outcome. The conclusion validity of this experiment is high. Since our participants were undergraduate students, it was possible to have a large number of participants in comparison to most other studies. Therefore the statistical power is not a threat to the validity. Since there were 25 given defects in the ER-Model, and it was clear to assess whether a defect was found or not, our measures were reliable too. The biggest threat to conclusion validity was the eventuality that, although the students were put together randomly, they had differing abilities and knowledge concerning ER-Models and finding defects. Therefore, we assessed the students' ability to deal with ER-Model inspections. The results of our measures indicated that there were no significant differences in the average marks (in class assessment) of the four groups in this assessment. Therefore this threat to the validity can be excluded.

**Internal Validity:** Internal validity is described as the degree to which the reality under study is accurately represented by the results and the conclusion. The experimental design was such that, subjects were required to use Checklist-based reading during the individual preparation stage of the experiment. The experiment also functioned as their midterm exam. During the inspection meeting they were divided into treatment groups in which they were required to apply either the Ad Hoc or Checklist approach. It is possible that the students who were assigned to Ad Hoc groups could intuitively build up the Checklist in their minds during the exam and refer back to their previous experience during the inspection meeting. It is also possible that students in the Ad Hoc groups felt disoriented since they were not provided with a Checklist as a starting point for identifying errors in the inspection. One can argue that the measurements taken from the inspection meeting for Ad Hoc groups were contaminated by these aforementioned issues. However, it is our belief that the considerable length of elapsed time (about 10 days) between each run of the experiment would have significantly alleviated this problem. Unfortunately, it is possible that students who participated in these inspection meetings were not highly motivated to do their best, since there were no incentives, such as marks, or other personal valuations to complete the task. Another threat to the internal validity is the varying guidance abilities of the tutor or lecturer overseeing each group during the inspection meetings.

**Construct Validity**: This validity concerns generalizing the result of the experiment to the concept behind the experiment. There is no threat to the construct validity, since the experiment was clearly structured and the object, an ER-Model, is easy to evaluate. The given defects could be found or not be found, there is no chance of interpretation differences.

**External Validity:** The question of whether the results can be generalized to industrial practice cannot be answered. Since the participants were undergraduate students, their selection may not be representative enough to directly transfer the results to the software industry. However, the results are of interest and, in particular, it is important to find out whether using both Checklists and Roles is a good approach. Thus, there is a great need for replication to enable the generalization of the results. The ER-Model may not be representative of industrial problems. The data model used in this study is smaller and less complex than industrial data models. However, the diagram uses a majority of the concepts normally found in an ER-model.

## 5. Data Analysis and Results

In order to analyze the effect of Ad Hoc and Checklist-based reading, and Roles in an ER-Model inspection meeting a two-tailed t-test was applied for the first two research questions. To examine the proportion of syntactic/semantic defects identified to the overall number of defects found, a two-tailed proportion test was used. For each of the tests, we used an alpha level of ten percent.

### 5.1 Groups with Roles versus Groups without Roles

The first research question stated in Section 3 led to the following hypothesis.

H0: Roles will not affect the percentage of defects identified
H1: Roles will affect the percentage of defects identified

To allow comparisons to be drawn between the Roles and No-Roles groups, the four groups were combined in the following way: The two groups using Roles were united to form a single group, while the two groups not using Roles were united to form a second single group. The formula used to represent the ability of finding defects was the number of correctly located defects found by the group, divided by the number of defects which existed overall in the ER-Model. The students in the group No-Roles were not given any guidelines concerning their behavior as a team, whereas the students in the Roles group were told to adopt Roles as either a Moderator, a Reader or a Recorder.

The teams in the No-Roles group found an average of 72% (std dev. 19.4%) of the 25 given defects in the ER-Model, whereas the teams in the Roles group only found an average of 64% (std dev. 22.4%) of these defects. The outcome of the t-test was that the null-hypothesis could be rejected (p=0.05) and therefore the teams using Roles performed significantly worse than the ones not using Roles.

To further examine the above findings for both Ad Hoc and Checklist readers, the effectiveness of Roles within each reading group was studied in detail. Again, a two-tailed t-test and an alpha level of ten percent to search for significant differences were used. The teams using Ad Hoc without Roles found 57% (std dev. 16.1%) of the given defects, the teams using Ad Hoc with Roles found 70% (std dev. 15.3%). This was a significant increase (p=0.01). In other words, Roles played a positive effect when  tailored to Ad Hoc reading in defect detection. Compared to the Checklist teams, there was a decrease in the percentage of correctly found defects. The teams using Checklist without Roles found 80% (std dev. 16.1%) of the given defects, while the teams in the Checklist with Roles group only found 57% (std dev. 27.0%). This decrease was significant (p=0.01). In other words, Roles lost its positive effect when it was combined to Checklist-based reading which was already tailored to defect detection.

It is quite interesting that the results of this further research go into different directions. Applying Roles to the Ad Hoc reading increases the performance, while applying them to the Checklist-based reading decreases it. However the Checklist with roles seems to dominate, since the application of Roles generally decreases the performance significantly.

### 5.2. Ad Hoc Readers versus Checklist Readers

The following hypothesis resulted from research question two.

H0: Structuring the inspection process by using a Checklist will not affect the number of defects identified
H1: Structuring the inspection process by using a Checklist will affect the number of defects identified

When comparing the groups that used the Ad Hoc approach to those that used a Checklist, two new groups were formed by combining both the Ad Hoc groups and the two groups using a Checklist.

The average percentage of defects found was 64% (std dev. 16.8%) for the teams without a Checklist, while teams using the Checklist found 71% (std dev. 23.5%) of all defects in average. Obviously there was an increase in the performance of the Checklist group. The t-test came up with a p-value of 0.08 and therefore the null-hypothesis could be rejected. The teams in the Checklist group performed significantly better than the ones applying the Ad Hoc approach. Thus there is reason to believe that structuring the inspection process by using a Checklist increases the number of defects found in an ER-Model.

To further examine the above findings, the effectiveness of reading approaches for Roles versus No-Roles for both Ad Hoc and Checklist readers was also studied. Ad Hoc readers without Roles found 57% (std dev. 16.1%) of defects whereas Checklist readers without Roles found 80% (std dev. 16.1%) of the defects. Checklist readers without Roles performed significantly better than Ad Hoc readers without Roles (p=0.01). On the other hand, Ad Hoc readers with Roles outperformed Checklist with Roles (p=0.05). Ad Hoc with Roles found 70% (std dev. 15.3%) of the given defects, Checklist with Roles only found 57% (std dev. 27.0%).

Like in 5.1 the results of this further examination go into different directions. The combination of both aids, Checklist and Roles, seems to raise some problems.

### 5.3. Quota of Semantic/Syntactic Defects

The idea of classifying the overall number of defects in the inspection material is not completely new in software inspection literature. [1] classified the defects of his requirements document according to their location in the document and the severity level of their impact on development and product quality. He found that Scenario-based readers were more successful at identifying defects with a high severity level than the reviewers using a checklist. In our research defects are categorized as either semantic or syntactic defects.

Similar to questions one and two, four groups (No-Roles, Roles, Ad hoc, Checklist) were formed, but this time the key figure was the ratio of syntactic/semantic defects found to the number of overall defects identified by the groups. In other words, the focus was set on the question of whether the groups' overall found defects had a different proportion of syntactic/semantic defects. This time the number of correctly found defects by each group was the basis of our figures. This led to the following hypothesis:

H0: The approach taken has no affect on the quota of semantic/syntactic defects of overall defects found

H1: The approach taken has an affect on the quota of semantic/syntactic defects of overall defects found

Table 2: Detection rate for defect categories and the proportion test p-value

| Defect classification | Semantic (%) | Syntactic (%) | p-value |
|---|---|---|---|
| No-Roles | 41.73 | 58.27 | 0.87 |
| Roles | 43.36 | 56.64 | |
| Ad Hoc | 44.03 | 55.97 | 0.78 |
| Checklist | 41.38 | 58.62 | |

Table 2 shows the results of the above hypothesis. Similar to research questions one and two we compared the four illustrated groups in order to find out, whether there were significant differences in the results. To calculate the numbers in Table 2, the number of syntactic and semantic defects found by the group was divided by the overall number of defects identified by the group.

Since the question had not been discussed before, there was no guideline to which approach increases or decreases the quota of semantic/syntactic defects found. Therefore we had to formulate the hypothesis above, which made a two-tail test necessary. Since we used proportions in these studies, we had to apply a proportion test. Again an alpha-level of ten percent was used. The p-values for the compared approaches are also included in Table 2. The null-hypothesis could not be rejected and therefore the differences are not significant.

### 5.4. Combined Effect

Referring back to our 2x2 experimental design, we took a look at the original groups and compared the combination of reading approaches and Roles. The values in Table 3 were calculated by dividing the number of correctly found defects by 25, which is the number of overall defects in the ER-Model.

Table 3: Performance of combined groups

| Defect Detection Rate | Mean (%) | Std dev. (%) |
|---|---|---|
| Ad Hoc without Roles | 57.00 | 16.10 |
| Checklist without Roles | 80.00 | 16.08 |
| Ad Hoc with Roles | 70.15 | 15.27 |
| Checklist with Roles | 57.92 | 26.91 |

It is remarkable that the group Checklist without Roles performed so much better than the other three groups. It was expected that the application of Roles to the teams using a Checklist would increase the performance, or at least not affect it, but in this case a significant decrease was found. Looking at the application of Roles to the Ad Hoc groups, we found the expected increase.

Comparing the two best performing groups, Checklist without Roles and Ad Hoc with Roles, one could conclude that the Checklist is a more helpful aid than Roles, since both of these groups are working with only one aid (on the one hand Checklist, on the other hand Roles) and there was a difference of 10 % in their performance.

We could not determine a difference between Ad Hoc without Roles and Checklist with Roles. The results are discussed in section 6, where we try to explain the interesting combined effects of two given aids.

# 6. Discussion

## 6.1 Roles versus No-Roles

The findings showed that not only did the usage of Roles not improve the defect detection rate, the group performance was even significantly lower than that of the teams working without this guideline. When taking a closer look at Table 3, it becomes clear that the usage of Roles actually does have a positive effect on the groups using the Ad Hoc reading, whereas the Checklist with Roles group had an unexpectedly low defect detection rate, and in this way had a negative effect on the whole Roles group.

The reasons for this result are only speculative. It is plausible that one or more of these reasons might provide the explanation. Since the Ad Hoc approach does not use any systematic approach to defect detection, Roles may provide a structured approach to the defect detection process. Looking at only the Checklist groups, the usage of Roles results in more than 22% lower performance than the Checklist groups applying the original meeting structure. Another reason might be the distribution of Roles between the students. In groups with Roles, students decided amongst themselves which roles they were allocated. In some groups we observed that the Recorder was inclined to act as a leader and a decision maker, and tended to make the final decision about the defect. The same dominating leadership behavior was also observed in some groups without Roles. Although there was no Role assigned to students in these groups, one of the students tended to dominate the group throughout the inspection process. This approach may not the best form of making decisions, especially if the leader is not competent on the topic.

## 6.2. Performance of the Checklist Readers

As seen in Table 3, the performance of the Checklist with Roles group was very low when compared to the participants using the Checklist method without Roles. The question arises as to whether two given aids cause quite the opposite of a more structured meeting, and thus result in less targeted procedure. More provocatively, they are simply not suitable to combine unless they are adapted to each other. This assumption was made after looking at the two groups using the Ad Hoc reading (Table 3). In their case, the usage of Roles had quite the opposite effect, namely, a positive effect on the ability to find defects. One can again speculate that when applying the Ad Hoc reading the distribution of Roles in the team meeting adds the missing structure and support needed to assure a successful meeting. Furthermore, every group had received instructions prior to the inspection and during the inspection process from a tutor. The individual group execution, therefore, also depended on the particular guidance, which may not necessarily be similar to each other. In other words, the tutorial guidance of the Checklist with Roles group could have been poorer than that of the other groups although we tried to instruct them to ensure as equal information as possible. Nonetheless, a statistically proven answer to the question concerning the cause for their low defect detection rate cannot be given.

## 6.3 Checklist versus Ad Hoc

With respect to the second hypothesis, the conclusion is quite obvious. The groups using the Checklist-based reading had a significantly higher defect detection rate than the participants using Ad Hoc. This result complements findings from other studies e.g. [7]. In our experiment, the reason for the success of the Checklist reviewers may have been the quality of the Checklist employed. Since the Checklist was created systematically for our specific ER-Model, each checkpoint of the ER diagram was included. This fact assured much better coverage of the inspection material. Another factor related to students' preference. In couple of tutorials students who are assigned to Ad Hoc groups complained about the Ad Hoc reading, declaring that they would prefer a Checklist to the Ad Hoc reading and that they could do a better job with a Checklist. Another reason for the relatively better performance of groups using the Checklist may again be the lack of experience that the students had with the inspection process. The guidance provided in the Checklist-based reading therefore had a significant impact on the group's ability of finding defects.

## 6.4. Quota of semantic/syntactic defects

In this experiment, defects were categorized as syntactic defects, which were related to ER-Model notation, and semantic defects, which were related to business rules and implementation of the data model. The results showed that participants were able to find more syntactic defects than semantic defects. In other words, they were comfortable with handling notation but not with implications of business rules and database implementation issues.

With respect to the third main research question, it was quite interesting to find that the quota of semantic/syntactic defects found by the four groups displayed in Table 2 was not statistically different. A two-tailed proportion test validated this result. The fact that the proportion of the classified defects found remains the same with each of the methods, leads to the conclusion that in our experiment only the total amount of defects found could be influenced by the choice of a different reading approaches. However, *e.g.* Checklist participants were not able to find a greater proportion of semantic or syntactic defects. In summary, we can say that the chosen defect detection technique may have a significant impact on the total number of defects found, but does not have an effect on the proportion of semantic/syntactic defects within this number.

## 7. Conclusion and Further Research

The experimental data raises many interesting questions. Firstly, and perhaps most interestingly, is the question of whether the usage of Roles does in fact not improve the inspection process or, as in this case, have a negative effect? Since there has not been a sufficient amount of research in this area, we recommend further research on the effects of Roles in team meetings.

Secondly, the data validates that the Checklist reading is a superior inspection method when compared to the Ad Hoc reading when it comes to inexperienced inspectors. The question arises as to whether the Checklist support can always improve the defect detection rate. Our results also imply that the quality of the used Checklist may have a great impact on the defect detection rate of this technique.

Also, further research is needed in the area of defect classification and the effects of different reading approaches on the quota of semantic/syntactic defects. As mentioned earlier, a significant difference in the proportion of semantic and syntactic defects could not be found when comparing Roles to No-Roles and Checklist to Ad Hoc.

Another topic of our research was to conduct a statistical investigation into the combination of aids. As we arranged the four groups consisting of Ad Hoc with Roles, Ad Hoc without Roles, Checklist with Roles, Checklist without Roles, we received data which allowed us to conclude that the combination of Checklist and Roles had a negative effect on the defect detection rate. Can the use of two given aids actually result in a lower ability to find defects? Does this apply to inspectors on a whole, or only to those with the least experience?

## Acknowledgement

The authors would like Dr. Lesley Land for her valuable input on the Roles of members of inspection teams. Two of the authors, Caroline Rombach and Oliver Kude, were intern students at the School of Information Systems, Technology and Management (SISTM), University of New South Wales, Australia and they supported by a scholarship from the Centre for Advanced Software Engineering Research (CAESER).

## References

[1] S. Biffl, ‘Analysis of the Impact of Reading Technique and Inspector Capability on Individual Inspection Performance’. *IEEE, 7th Asia-Pacific Software Engineering Conference*, 2000, 136-145

[2] P.P. Chen, ‘The Entity-Relationship Model – Toward a Unified View of Data’. *TODS*, 1976, 1(1), pp. 9-36

[3] A.M. Davis, *Software Requirements: Objects, Functions, and States*, Prentice Hall, USA, 1993

[4] M.E. Fagan, ‘Design and Code Inspections to Reduce Errors in Program Development’. *IBM Systems Journal*, 1976, 15(3), pp. 182-211

[5] B. Freimut, O. Laitenberger, and S. Biffl, ‘Investigating the Impact of Reading Techniques on the Accuracy of Different Defect Content Estimation Techniques’. *IEEE, 7th Software Metrics Symposium, METRICS 01*, 2001 pp. 51-62

[6] P. Fusaro, F. Lanubile and G. Visaggio, ‘A Replicated Experiment to Assess Requirements Inspection Techniques’. *Empirical Software Eng.*, 1997, 2, pp. 39-57

[7] M. Halling, S. Biffl, T. Grechenig and M. Koehle, ‘Using Reading Techniques to Focus Inspection Performance’. *IEEE Proceedings of Euromicro Conf.*, 2001, pp. 248-257

[8] O. Laitenberger, ‘Cost Effective Detection of Software Defects through Perspective-based Inspections’. *Empirical. Software Engineering*, 2001, 6, pp. 81-84

[9] L. Land, C. Sauer, and R. Jeffery, ‘The Use of Procedural Roles in Code Inspections: An Experimental Study’. *Empirical Software Enineering.*, 2000, 5, pp. 11-34

[10] J.M. Levine and R.L. Moreland, ‘Progress in small group research’. *Annual Rev. of Psychology*, 1990, 41, pp. 585-634

[11] J. Miller, M. Wood and M. Roper, ‘Further Experiences with Scenarios and Checklists’. *Empirical Software Engineering*, 1998, 3, pp. 37-64

[12] D.L. Parnas, and D.M. Weiss, ‘Active Design Reviews: Principles and Practices’. *Journal of Systems & Software.* 1987, 7, pp. 259-265

[13] A.A. Porter, L.G. Votta and V.R. Basili, ‘Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment’. *IEEE Transactions on Software Engineering*, 1995, 21(6), pp. 563-575

[14] P. Rob and C. Coronel, *Database Systems: Design, Implementation, and Management*. Course Technology, Thomson Learning, Inc, Boston, Mass, USA, 2002

[15] K. Sandahl, O. Blomkvist, J. Karlsson, C. Krysander, M. Lindvall and N. Ohlsson, ‘An Extended Replication of an Experiment for Assessing Methods for Software Requirements Inspections’. *Empirical Software Engineering* 1998, 3, pp. 327-354

[16] C. Wohlin and A. Aurum, ‘Evaluating Cost-Effectiveness of Checklist-Based Reading of Entity-Relationship Diagrams”. To be published in the proceeding of *9th Int. Software Metrics Symp.*, Sydney, Australia, Sept. 3-5, 2003

[17] C. Wohlin, H. Petersson, and A. Aurum, ‘Combining Data from Reading Experiments in Software Inspections: A Feasibility Study’. *Lecture Notes in Empirical Software Engineering.* N. Juristo and A. Moreno (Eds.), World Scientific Publishing. Springer, Berlin, Germany, 2003

[18] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, The Netherlands, 2000