Empir Software Eng
DOI 10.1007/s10664-011-9174-8

Replication of an experiment on linguistic tool support for consolidation of requirements from multiple sources

Krzysztof Wnuk · Martin Höst · Björn Regnell

Editor: Daniel M Berry

Abstract Large market-driven software companies continuously receive large numbers of requirements and change requests from multiple sources. The task of analyzing those requests against each other and against already analyzed or implemented functionality then recording similarities between them, also called the requirements consolidation task, may be challenging and time consuming. This paper presents a replicated experiment designed to further investigate the linguistic tool support for the requirements consolidation task. In this replication study, 45 subjects, working in pairs on the same set of requirements as in the original study, were assigned to use two methods for the requirements consolidation: (1) lexical similarity and (2) searching and filtering. The results show that the linguistic method used in this experiment is not more efficient in consolidating requirements than the searching and filtering method, which contradicts the findings of the original study. However, we confirm the previous results that the assisted method (lexical similarity) can deliver more correct links and miss fewer links than the manual method (searching and filtering).

Keywords Requirements engineering · Experiment · Linguistic method · Replication

1 Introduction

Requirements engineering in a market-driven context can be characterized by continuous elicitation, time-to-market constraints, and strong market competition

K. Wnuk (✉) · M. Höst · B. Regnell
Department of Computer Science, Lund University, Ole Römers väg 3, 223 63 Lund, Sweden
e-mail: Krzysztof.Wnuk@cs.lth.se

M. Höst
e-mail: Martin.Host@cs.lth.se

B. Regnell
e-mail: Bjorn.Regnell@cs.lth.se

(Natt och Dag 2006a; Regnell and Brinkkemper 2005). In this context, requirements are continuously arriving from multiple sources, throughout the development process (Regnell et al. 1998). When the company is growing and expanding, more products are created which result in a more complex variability structure, and more effort is needed to handle product customizations, for example by utilizing the Software Product Line (SPL) concept (Pohl et al. 2005). This constant flow of requirements needs to be analyzed from the perspective of new market opportunities and technical compliance. In a case when a company is large and develops complex software solutions, the quantity of information to constantly analyze and assess may severely impede the analytical capacity of requirements engineers and managers (Gorschek et al. 2007; Leuser 2009). Providing a method that can assist in analyzing large numbers of natural language requirements for the purpose of finding and recording similarities between them can significantly reduce time needed to perform the task (Cleland-Huang et al. 2007), help to miss fewer requirements links (Natt och Dag et al. 2006) and increase the accuracy of the task.

The process of analyzing incoming requirements from customers or customer representatives (also called proxy-customers) against requirements already present in the requirements repository can be called *requirements consolidation*. This process includes gathering incoming documents, finding similarities, and merging or linking similar descriptions into a consolidated single description that covers all analyzed aspects. This process can also be a part of the broader impact analysis task. The core of the requirements consolidation process is finding the similarities between requirements and recording them by making links between them (Natt och Dag et al. 2006). However, the number of possible links grows exponentially with the increase of the number of requirements to analyze, which may result in overwhelming the company's management and analytical skills (Leuser 2009). As a remedy to this problem, Natt och Dag et al. (2006) developed and evaluated a method for requirements consolidation that utilizes linguistic techniques and provides a list of requirements that are the most similar to the currently analyzed requirement. The evaluation of this method showed that using the method can significantly improve the performance of the consolidation process as well as the number of correctly linked requirements, and that it can help to miss fewer requirements links (Natt och Dag et al. 2006). However, the *unsupported* method used in the original experiment was limited to a simple search functionality, while most currently available requirements management tools offer more advanced filtering and searching techniques.

This replication study has been designed to assess whether the tool with a linguistic analysis of the similarity between requirements can still perform better than currently available commercial requirements management tools in the task of requirements consolidation. A replicated experiment has been chosen due to its falsifiable nature. Replications provide a possibility to evaluate whether the output parameters of a system remain stable if one or more input parameters are systematically changed.

In this experiment, two subject groups, working in pairs, were asked to consolidate two requirements sets by finding and linking requirements that address the same underlying functionality. This replication reuses the original procedures in terms of the study design, experimental steps and the two requirement sets. The changes to the original experiment are: (1) using another set of subjects which were asked to work in pairs due to unexpected housing issues and (2) changing one of the treatments. Due to a limited number of available computers in the laboratory

room, the subjects were asked to work in pairs on the assignment. Given this context, this replication study can be classified according to Shull et al. (2008) as a dependent replication. However the classification provided by Shull does not define if a replication where both the population and one of the methods used to test the hypotheses is changed can also be categorized as an exact replication (Shull et al. 2008, only mention changing either the population or the artifact) on which the technique is applied. According to the classification by Basili et al. (1999), this replication type is the one that varies the research hypotheses. The unchanged object in this replication study, also called the *assisted* method, is a research prototype tool, called ReqSimile (Natt och Dag 2006b), that utilizes linguistic analysis to assist in the task of finding similar requirements. The second object, which was changed compared with the original experiment, is called the *manual* method, and it utilizes searching and filtering functionalities implemented in a tool called Telelogic Doors (IBM 2010a).[1]

The objectives of the study are twofold: firstly to assess if a significant differences between the two methods tested in the original experiment can be confirmed in a replicated experiment setup and secondly to compare the results for the same methods between the two experimental sessions. The objectives are refined to two main research questions in Section 4.

The paper is structured as follows: Section 2 provides industrial problem description. Section 3 provides related work. Section 4 describes the experimental design. Section 5 explains experiment execution procedures. Section 6 describes the experiment results analysis. Section 7 provides an interpretation of results. Section 8 concludes the paper.

2 Industrial Problem Description

New requirements and changes to existing requirements are inevitable situations at all stages of the system development process (Kotonya and Sommerville 1998). The two principal requirements management activities that address this phenomena are: (1) change control and (2) change impact assessment. The change control ensures that, if a change is accepted, its impact on design and implementation artifacts will be addressed. The change impact assessment warrants that proposed changes have a known impact on the requirements and software system (Kotonya and Sommerville 1998). A company that is operating in a market-driven mode should continuously monitor the market situation by checking competitors latest achievements, researching market needs and collecting all possible feedback from the market in a chase for achieving or maintaining the competitive advantage within its operational business. This pursuit after an optimal market window, together with other reasons, creates a constant flow of new requirements and ideas throughout the entire software product lifetime (Karlsson et al. 2002). As a result, the requirements

[1]The Telelogic DOORS tool has recently changed its vendor and its name to Rational DOORS. However, since the Telelogic Doors version 8.3 was used in this experiment, we will refer to this tool throughout this paper as Telelogic Doors. Both methods were compared for the task of requirements consolidation meaning that comparing the two tools in general is outside of the scope of this paper.

process for market-driven contexts needs to be enriched with procedures to capture and analyze this constant flow of requirements (Higgins et al. 2003).

As pointed out by practitioners from large companies (Berenbach et al. 2009), when development projects grow in complexity and new products are released to the market with many features, the importance of good practices in requirements management grows. In the case when a company is large and operates globally, the diversity of customers and the complexity of software products can make the list of sources of new requirements and change requests extensively long, including: customers and proxy-customers (marketing, customer representatives and key account managers), suppliers, portfolio planners and product managers. The company's requirements analysts should, in this, case analyze all incoming requirements and change requests in order to find an optimal set of requirements that will address the needs of as many customers as possible. In this context, the concept of Software Product Lines (SPL) (Pohl et al. 2005) is often used to increase the reuse of common components while providing necessary diversity of similar products, requested by various customers.

Change management in a Software Product Lines context can be particularly challenging, for example, because of the extensive and often exhaustive variability analysis that has to be performed while analyzing the impact of a change. Moreover, the requirements analyst has to consider if a certain new requirement or request has already been analyzed and what was the result of this analysis. One of the methods to assist with the analysis of incoming requirements versus those already present in the requirements database is to find and record similarities, making traceability links. In the industrial case example, provided by the original experiment, the experts became frustrated during the analysis because they had to identify compliance to the same or very similar requirements multiple times. Large parts of the new versions of requirements request documents, arriving from the same customer are typically the same as previous versions. Furthermore, the same and very similar requirements can appear in the request from different customers (Natt och Dag et al. 2006). Providing an automatic or semi-automatic method of analyzing similarity between incoming requirements could significantly decrease the amount of time needed to perform this task.

The process of finding and recording similarities between software development artifacts is a part of the requirements traceability activity, which has been widely recognized as a useful method for recording relations and dependencies between software project artifacts for the purposes of change and impact analysis tasks (Ramesh et al. 1995; Wiegers 2003; Antoniol et al. 2002; Jarke 1998; Gotel and Finkelstein 1994). The core of the requirements traceability task is to find and record the dependencies between the traced elements, which are assumed to be exhibited by their lexical similarity (Natt och Dag 2006a). The importance of requirement traceability is significant; the U.S. Department of Defense invested in 2001 about 4 percent of its total IT budget on traceability issues (Ramesh and Jarke 2001). Other large companies, have also stressed the importance of implementing traceability in their industry projects (Samarasinghe et al. 2009; Berenbach et al. 2009; Konrad and Gall 2008; Panis 2010; Leuser 2009).

However, despite recognition of its importance, implementing a successful traceability in practice is challenging (Cleland-Huang et al. 2002). The task of finding relationships between the elements and establishing traces between them is a

"mind numbing" (Hayes et al. 2003), error prone and time consuming activity. Moreover, maintaining a traceability scheme is difficult because the artifacts being traced continue to change and evolve as the system is developed and extended (Zowghi and Offen 1997; Strens and Sugden 1996). Furthermore, as pointed out by Leuser (2009), current traceability approaches used in practice are cumbersome and very time consuming, mainly because they are almost completely manual. The size of requirements specifications in large industrial projects may reach thousands of requirements (Leuser 2009; Konrad and Gall 2008). To tackle these issues, several researchers proposed using Information Retrieval (IR) methods such as the Vector Space Model (VSM), also used in this experiment, (Antoniol et al. 2002; Cleland-Huang et al. 2007; Natt och Dag et al. 2004), the Probabilistic Network Model (Cleland-Huang et al. 2005, 2010), and Latent Semantic Indexing (LSI) (Lucia et al. 2007; Hayes et al. 2006; Lormans and Van Deursen 2006; Marcus and Maletic 2003) for semi-automatic recovery of traceability links.

3 Related Work

Replications play an important role in software engineering by allowing us to build knowledge about which results or observations hold under which conditions (Shull et al. 2008). Unfortunately, replications in software engineering are still rarely reported (Ivarsson and Gorschek 2009). A recent survey of controlled experiments in software engineering revealed that replications are still neglected by empirical researchers, only 18% of the surveyed experiments are reported as replications (Sjøberg et al. 2005). Moreover only 3.9% of analyzed controlled experiments can be categorized according to the IEEE taxonomy as requirements/specification related (Sjøberg et al. 2005; IEEE 2010).

The awareness of new possibilities that Natural Language Processing (NLP) can bring to requirements engineering has been present from the beginning of the requirements engineering discipline, when Rolland and Proix (1992) discussed the natural language approach for requirements engineering. Shortly after, Ryan (1993) warned that although natural language processing provides a variety of sophisticated techniques in the requirements engineering field, they can only support sub-activities of requirements engineering and that the process of using natural language processing techniques has to be guided by practitioners. The possibilities mentioned by Ryan (1993) and Rolland and Proix (1992) have later been explored by a number of research studies and publications, where applications of various NLP techniques in supporting requirements management activities were evaluated and discussed. Among those that include some kind of empirical evaluations, the vast majority of natural language process tools are used to examine the quality of requirements specifications in terms of, for example, the number of ambiguities (Fantechi et al. 2003) by assigning ambiguity scores to sentences depending on the degree of syntactic and semantic uncertainty (Macias and Pulman 1995), or detecting ambiguities by applying an inspection technique (Kamsties et al. 2001). Furthermore, Rupp (2000) produced logical forms associated with parsed sentences to detect ambiguities. Among other quality attributes of requirements artifacts that natural language processing attempts to analyze and improve, Fabbrini et al. (2001) proposed a tool that assesses understandability, consistency, testability, and

correctness of requirements documents. Providing measurements that can be used to assess the quality of a requirements specification document is the aim of the ARM tool proposed by Wilson et al. (1997). Mich et al. (2002) reported on an experiment designed to assess the extent to which an NLP tool improves the quality of conceptual models. Finally, Gervasi and Nuseibeh (2000) used natural language processing techniques to perform a lightweight validation (low computational and human costs) of natural language requirements.

Apart from the quality evaluation and assurance tasks, NLP techniques have also been applied to the task of extracting abstractions from textual documents (Aguilera and Berry 1991; Goldin and Berry 1997) and helping combining crucial requirements from a range of documents that include standards, interview transcripts, and legal documents (Sawyer et al. 2002). Sawyer et al. (2005) have also reported how corpus-based statistical language engineering techniques are capable of providing support for early phase requirements engineering. Rayson et al. (2001) reported experiences from a project where probabilistic NLP techniques were used to identify and analyze domain abstractions. Their results were further supported by a later study by Sawyer et al. (2004), where ontology charts of key entities were produced using collocation analysis. The continued interest in this issue has been reported by Gacitua et al. (2010) who proposed a new technique for the identification of single- and multi-word abstractions named Relevance driven Abstraction Identification (RAI). Finally, Gervasi et al. (1999) used lexical features of the requirements to cluster them according to specific criteria, thus obtaining several versions of the requirements document.

The ReqSimile tool evaluated in this paper uses a correlation to measure lexical similarity and thus rank candidate requirements for linking, presenting to the user the "top" subset of those requirements. The linguistic method, also called the *cosine* measure, uses a vector-space representation of requirements where each requirement is represented using a vector of terms with the respective number of occurrences (Natt och Dag et al. 2004; Manning and Schütze 2002). Each term can be seen as a dimension in an N-dimensional space while a whole requirement can be represented as a point in the N-dimensional space. Similar requirements will be represented in this space as m points closely clustered. From the matrix, which shows how many times a term appears in each requirement, the information may be derived about how many terms the two requirements have in common. The very similar requirements will result in closely clustered points in this vector space (Manning and Schütze 2002). In the evaluated method (Natt och Dag 2006a) a frequency of terms has been used, instead of counting the occurrences. The cosine correlation measure is often chosen in text retrieval applications for the purpose of finding similar requirements, as it does not depend on the relative size of the input (Manning and Schütze 2002).

$$\sigma(f,g) = \frac{\sum_t w_f(t) * w_g(t)}{\sqrt{\sum_t w_f(t)^2 * \sum_t w_g(t)^2}} \tag{1}$$

The measure in (1) is used for calculating the degree of similarity, where f and g are two requirements, t ranges over terms, and $w(t)$ denotes the weight of term

t. The term weight is typically a function of the term frequency, since while the number of times a word occurs is relevant, its relevance decreases as the number gets larger (Manning and Schütze 2002). However, there is a challenge in the way stemming rules are used in this method. For example, the stemming rules do not reduce the verb *containerization* and the noun *container* to the same stem. From a semantic point of view this is perfectly correct, but as the two terms concern the same domain concept their association should be utilized to increase the similarity measure. The realization of the Vector Space Model used in this paper does not support this association. Another potential problem has to do with synonyms as they are not considered in the model. Finally, as mentioned in (Natt och Dag 2006a), there is no guarantee that two requirements that are similar according to the $\sigma(.)$ measure are indeed related. The method evaluated does not consider hypernyms and hyponyms (Jackson and Moulinier 2002).

The ReqSimile tool evaluated in this paper is not the only research tool that provides support for requirements traceability. Hayes et al. (2007) proposed a REquirements TRacing On-target (RETRO) tool that uses the LSI technique to find similarities between analyzed elements and help the analyst with making traceability links. Lin et al. (2006) proposed a Web-based tool called POIROT that supports traceability across distributed heterogeneous software artifacts. A probabilistic network model is used by POIROT to generate traces between requirements, design elements, code and other artifacts stored in distributed third party case tools.

The second method evaluated in this study uses searching and filtering functionalities provided by Telelogic DOORS tool (IBM 2010b). According to the product documentation, the "finding text in a module" function can search for all the objects that contain a specific search string. The tool displays the options of the search that have already been set in the search window. The possible options are: (1) highlight matches, (2) match case and (3) use regular expressions. To change the search options, the *Advanced* tab in the same window has to be used. Additionally, it is possible to select the attributes included in the search, for example object heading or object text. The tool provides UNIX-style regular expressions support when searching for text. For example, using **c.t** will search for all three letter words that start with **c** and end with **t**. Using **200[123]** will search for either 2001, 2002, or 2003. Subjects during the experiment can also use filters to control what data is displayed on the screen. The tool provides two types of filters: simple and advanced. Simple filters can be used to filter: (1) the contents of every attribute of type text or string, (2) object heading number, (3) the content of any column or the value of a single attribute of any type. Additionally, it is possible to filter on the basis of whether the object has links or is either the current object or a leaf object. Using advanced filters gives the possibility to combine simple filters to create complex filters, specify filter options that control what is displayed.

Using filters requires more steps than using the searching functionality. First, the filter has to be defined and its attributes have to be set. After this step, the user has to define if the filter should match case option or regular expressions. Finally, it is possible to filter only objects that have certain types of links, such as in-links, objects that are leafs and filter the content of columns. While using advanced filters, it is possible to combine previously defined simple filters by using **And, Or** and **Not** to combine them into logical expressions. It is also possible to control the output of applying a filter. The possible options are: (1) show ancestors or (2) show

descendants of the object that match the filter criteria and (3) display all table cells and their contents regardless of whether they match the filter criteria. According to the product documentation, the Telelogic DOORS tool does not provide any auto-completion, stemming or proximity search options. Exclusion searches are possible by combining two simple filters where one is negated by using the "NOT" logical expression.[2]

The requirements consolidation task can, in a broad perspective, be considered as the more general task of negotiating the scope of the future project with multiple stakeholders. Assuming this broad definition of the task, we discuss alternative approaches of supporting this negotiation process. Fricker et al. (2010) propose a negotiation process, called handshaking with implementation proposals. The process has been used to communicate requirements effectively, even in situations where almost no written requirements exist and where distance separates the customer from the developers. The architectural options are used in this case to understand requirements and make implementation decisions that will create value. Sommerville et al. (1997) proposed a viewpoint-based approach to requirements engineering which may be used to structure the requirements description and expose conflicts between different requirements. As systems usage is heterogeneous, viewpoints can organize different types of information needed to specify the system and by that help to structure the process of requirements elicitation. Breaux (2009) uses grounded theory to analyze regulations and legal documents for the purpose of ensuring that the software system to be built is demonstrably compliant with relevant laws and policies.

4 Experimental Design

The two main objectives of this study, presented in Section 1, can be further refined to the following research questions:

Q1: Can significant differences between the *assisted* and the *manual* methods that were achieved in the original experiment be confirmed in a replicated experiment where the original *manual* method is replaced with a keyword searching and filtering tool?

- **Q1a:** Is the *assisted* method more efficient in consolidating two requirements sets than the *manual* method (where efficiency is calculated as the number of analyzed requirements)?
- **Q1b:** Is the *assisted* method more correct in consolidating two requirements sets by assigning more correct links than the *manual* method (correctness is calculated as a number of correctly linked requirements)?
- **Q1c:** Does the *assisted* method help to miss fewer requirements links than the *manual* method?

[2]The information about searching and filtering functionalities has been based on the manual for DOORS version 8.3. The instruction of how to use Telelogic Doors for linking used in this experiment is available at http://fileadmin.cs.lth.se/serg/ExperimentPackages/ReplicationReqSimile/HelpSheetDoors.pdf.

Q2: Is there any difference between the original and replicated experiment sessions for the same method?

- **Q2a:** Is there any difference in the results for the *assisted* method between the original and the replicated experiments?
- **Q2b:** Is there any difference in the results for the *manual* methods between the original and the replicated experiments?

The aim of Q1 is to assess if the results obtained in the original experiment holds even if one of the tools is changed. The research question Q1 is divided into three sub-questions, where each of them is explicitly addressing various quality aspects of the consolidation process. Question Q2 aims to assess the difference between the two experiments. The possible differences provide valuable input regarding the nature of the consolidation task and the subjects used in both experiments.

The central part of the requirements consolidation task is finding similarities between the two sets of requirements as described in detail in Section 3. The methods evaluated in the experiment were implemented in two tools: (1) ReqSimile (Natt och Dag et al. 2006) and (2) Telelogic Doors (IBM 2010a). As mentioned in Section 1, the goal of the study is not to evaluate the tools in general, but to compare the methods that they provide. The planning phase was based on the original experiment (Natt och Dag et al. 2006) and, when possible, the original material is reused and extended according to the guidelines of designing experiments presented by Wohlin et al. (2000). In order to draw more general conclusions, the authors put additional effort into minimizing the difference between this experiment design and the original experiment design.

Since most of the design has been reused from the original experiment, the evaluation of the experiment design for the replication sake was limited to checking additional changes. The changes concern questionnaire improvements and new instructions regarding the use of the Telelogic Doors tool (IBM 2010a). The experiment design was evaluated by an additional researcher, experienced in conducting empirical research in software engineering, before executing the experiment. The same researcher participated in the pilot study where both tools were used to find similar requirements and create links between them. Comments and suggestions regarding readability and understandability of the laboratory instructions were given and later implemented. Finally, since the requirements sets used in this experiment were the same as in the original experiment, the correct answer to the consolidation task remained unchanged. The experiment pack can be accessed at http://fileadmin.cs.lth.se/serg/ExperimentPackages/ReplicationReqSimile/Experiment.rar.

Similarly to the original experiment, this replication study was also conducted in the form of a laboratory experiment, since it captures the consolidation problem in an untainted way. Figure 1 depicts the conceptual solution of the consolidation activity. To the left in Fig. 1, two requirement sets A and B are shown. They represent two consecutive submissions of requirements specifications from the same key customer. We can also assume that the earlier specification is set A in this case, and that it would have already been analyzed and the result from the analysis is available in the central requirements database. The subjects use one of the tools, either Telelogic Doors for the *manual* method (IBM 2010a) or ReqSimile for the *assisted* method (Natt och Dag et al. 2006), to find requirements in the set B that were already analyzed in the set A and to mark them by assigning a link between them. The

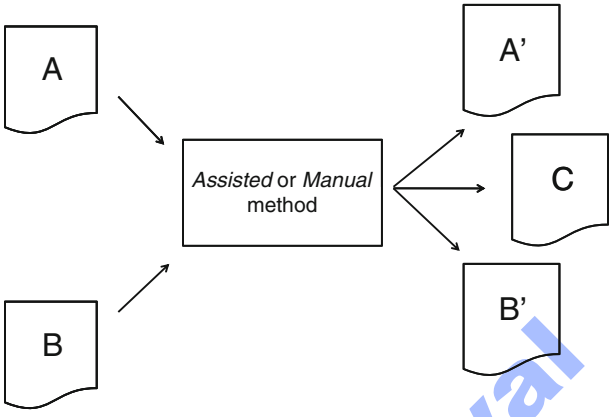

Fig. 1 The process of using the support tool for requirements consolidation

output of the process is shown to the right of Fig. 1. The subset A' comprises all requirements that are not linked to any requirement in the set A. The subset B' represents all new requirements that have not previously been analyzed. Finally, there is the subset C, which comprises all requirements in the new specification that previously have been analyzed. The analyst would then send the requirements in set B' to the experts for analysis. The experts are thus relieved from the burden of re-analyzing the requirements in subsets A' and C.

4.1 Goals, Hypothesis, Parameters and Variables

The variables in this experiment were kept unchanged from the original study (Natt och Dag et al. 2006). They can be grouped into independent, controlled and dependent:

- The independent variable is the method used in the experiment. The two methods compared are *manual* and *assisted*.
- The controlled variable is the experience of the participants. In order to analyze the individual experience of the subjects, a questionnaire was used.

The dependent variables are:

T	time used for the consolidation
N	the number of analyzed requirements
N_{cl}	number of correct links
N_{il}	number of incorrect links
N_{cu}	number of correctly not linked
N_{iu}	number of missed links (incorrectly not linked)

These dependent variables are used to analyze the hypotheses. The number of analyzed requirements is used in case the subjects are not able to analyze all requirements, which will affect N_{iu} and N_{cu}. The hypotheses for comparing the *manual* and the *assisted* method remain unchanged from the original experiment design. The rationale of the proposed hypotheses is based on the following theory regarding using the *assisted* method. The *assisted* method provides a list of candidate requirements ranked by their similarity degree to a currently analyzed requirement.

As a result the requirements analysts has to review only a subset of all possible combinations, for example the top ten candidate links. Thus we state a hypothesis that the *assisted* method can help to analyze requirements faster. Moreover, since the most similar requirements are placed next to each other on the list of candidates it is easier to read all potential candidates that exhibit high degree of lexical similarity. The result is expected to be an increased number of correct links, better precision and accuracy. Finally, the sorting according to lexical similarity should, in our opinion, help to miss fewer correct requirement links, since there is a high probability that all possible links to analyze will be show in the top 10 or 20 candidate requirements. Presented below are six null hypotheses:

- (H_0^1) The *assisted* method results in the same number of requirements analyzed per minute, N/T, as the *manual* method.
- (H_0^2) The *assisted* method results in the same share of correctly linked requirements, $N_{cl}/(N_{cl} + N_{iu})$, as the *manual* method.
- (H_0^3) The *assisted* method results in the same share of missed requirements links, $N_{iu}/(N_{cl} + N_{iu})$, as the *manual* method.
- (H_0^4) The *assisted* method results in the same share of incorrectly linked requirements, N_{il}/N, as the *manual* method.
- (H_0^5) The *assisted* method is as precise, $N_{cl}/(N_{cl} + N_{il})$, as the *manual* method.
- (H_0^6) The *assisted* method is as accurate, $(N_{cl} + N_{cu})/(N_{cl} + N_{il} + N_{cu} + N_{iu})$, as the *manual* method.

Since the subjects may not use exactly the same time for the task, the performance is normalized as the number of analyzed requirements divided by the total time spent on the consolidation task (in minutes).

4.2 Subjects

In this study, a different set of subjects compared to the original experiment, although from the same kind of population was used. The sample includes participants of the course in Requirements Engineering at Lund University (2011a). The course is an optional master-level course offered for students at several engineering programs including computer science and electrical engineering. It gives 7.5 ETCS points (ECTS 2010) which corresponds to five weeks full time study. Although the experiment was a mandatory part of the course, the results achieved by the subjects had no influence on their final grade from the course. The students were between 24 and 41 years old with an average of 27 years. There were four female and 41 male students. Before conducting the experiment, the subjects had been taught requirements engineering terminology and had gained practical experiences through their course project. The result from the pre-test questionnaire revealed that the difference in English reading and writing were small, varying from "very good knowledge" for the majority of subjects to "fluent knowledge" for some of them. When it comes to the industrial experience in software development of the subjects, most of them reported no experience at all (28 out of 45 students). Among the subjects that reported any degree of industrial experience, the length of the experience varied between three months and two years with an average value of 11 months. The analysis of industrial experience in pairs of subjects revealed that ten pairs had varying degrees of industrial experience which was not always equal.

Table 1 The number of years of industrial experience in software development for pairs of the subjects that participated in this replication

Pair of subjects	Experience of the first subject in the pair (in years)	Experience of the second subject in the pair (in years)
A1	0.5	1.5
A3	1	1
A7	1	2
A10	0	1
A11	0.5	1
M4	1	2
M5	0.5	0
M6	0.25	0
M7	0.5	1
M8	0.25	1.5

The remaining pairs of subjects exhibited no industrial experience for both pair members. The letter *A* indicates that a pair of subjects used the *assisted* method while the letter *M* indicates that a pair of subjects used the *manual* method. The IDs (Mx and Ay) are consistent with Table 5. The *rows highlighted gray* indicate data points that were removed from the analysis (outliers)

The analysis of the experience of both pair members is presented in Table 1. The difference in experience varied between three months and 15 months with an average value of nine months. The impact of the industrial experience on the results achieved by these subjects is outlined and discussed in detail in Section 7.1 and Table 7.

We have also performed an analysis of the experience of subjects from the project course that the requirements used in this experiment were developed here

Table 2 The experience and the roles of the subjects that participated in the replication from the course that the requirements originate from

Pair of subjects	Role of the first subject in the pair	Role of the second subject in the pair
M2	Have not taken the course	Tester
M3	Have not taken the course	System group member
M4	Development manager and developer	Test manager and tester
M5	Tester	Have not taken the course
M6	System group member	Project manager
M7	Have not taken the course	Developer
M9	System group member	Developer
M10	Development manager	Project manager
M11	Have not taken the course	System group member
A1	Project manager	Have not taken the course
A2	Project manager	Tester
A3	Test manager and tester	Have not taken the course
A4	Developer	Developer
A5	Have not taken the course	Tester
A6	Project manager	Have not taken the course
A7	Developer	Have not taken the course
A12	Developer	

The letter *A* indicates that a pair of subjects used the *assisted* method while the letter *M* indicates that a pair of subjects used the *manual* method (the numbers are consistent with Table 5). The *rows highlighted gray* indicate data points that were removed from the analysis (outliers)

(Lund University 2011b). The results of the analysis are presented in Table 2. The results revealed that 22 out of the 45 subjects have not taken the course that the requirements originate from, while the rest had taken the course and acted in various roles during the project phase of the course.

Next, the roles taken in the course that the requirements originate from in the pairs formed by subjects were analyzed. For nine pairs, outlined in Table 2, the pairs are formed by an inexperienced person and an experienced person from the course. This may have a positive impact on the task, since the more experienced person can help the inexperienced person to understand the nature and origin of the requirements set. However, the more experienced person can bias the consolidation task by bringing knowledge about the requirements sets and possible similar requirements from the course. The remaining seven pairs represented experience from various roles including: developer, development manager system group manager, project manager and tester. Only one pair had the same experience from being a developer, in other cases the roles taken in the course project did not overlap.

When it comes to the experience in analyzing and reviewing requirements, 80% of the subjects declared to have experience only from courses. Among the remaining 20% of the subjects, one pair (M5) had experience from both courses and industry (less than one year). In this case, the second pair member had only experience from courses. Furthermore, in cases (A1, A3 and A11), one pair member reported both experience from courses and less than a year of industrial experience in analyzing and reviewing requirements. In all three cases, these participants were paired with subject reporting only experience from the courses. Finally, in the case of pair M8, one member reported more than one year of industrial experience, while the other pair member reported no experience at all. The analysis of the results achieved by these subjects in relation to their experience is discussed in Section 7.1.

A further question concerned the subject's experience with the tool that implements the *manual* method, that is Telelogic Doors. The analysis indicated that 91% of subjects reported no experience with Telelogic Doors and that they had never heard about the tool. Although four persons have heard about the tool, they have never used it. We can conclude that the subjects are homogenous in this matter and that we can exclude this threat from aspects influencing the results.

4.3 Treatments

The treatments of the experiment are the methods used in supporting the requirements consolidation task. The assisted method is implemented in the ReqSimile tool using linguistic engineering to calculate the degree of similarity between requirements by lexical similarity as a way of approximating semantic similarity (Natt och Dag et al. 2004) (for more details see Section 3).

The other treatment is the *manual* method which comprises searching and filtering functionalities provided by the Telelogic Doors tool (IBM 2010a). The goal of the

Table 3 The treatments and tools used in the original and the replicated experiments

	Original experiment		Replicated experiment	
Treatment	Assisted method	Manual method	Assisted method	Manual method
Tool	ReqSimile	ReqSimileM	ReqSimile	Telelogic doors

experiment is not to compare the two tools in general, but the functionality that they provide to support the requirements consolidation task. The objects used in the original and the replicated experiment are listed in Table 3. Compared to the original experiment, one of the tools was kept unchanged while the second one was changed. The change comprises substituting ReqSimileM from the original design (Natt och Dag et al. 2006) by Telelogic Doors (IBM 2010a) for the manual method. More information regarding tools used in the replication can be found in Section 4.5.

4.4 Requirements

Two requirements sets were reused from the original experiment. The requirements specifications were produced as a part of a course "Software Development of Large Systems" (Lund University 2011b). The course comprises a full development project, including: requirements specification, test specification, high-level design, implementation, test, informal and formal reviews and acceptance testing. At the end of the course, the students deliver a first release of the controller software for a commercial telecommunication switch board. Two requirements specifications were randomly selected from the course given in years 2002 and 2003. The requirements have been specified in use case style or features style (Lauesen 2002), and all are written using natural language. Two requirements sets containing 30 and 160 requirements respectively, were imported to ReqSimlieA and Telelogic Doors. An example of requirements from the specification comprising 30 requirements is depicted in Table 4. However, the requirements were neither written by a native English language

Table 4 Example requirements from the specification comprising 139 requirements

Key	Id	Type	Selection	Description
3	Scenario13	Functional	Service: regular call	Regular call-busy actors: A: calling subscriber, B: called subscriber, S: system prerequisites: Both A and B are connected to the system and are not unhooked. Step 13.1. A unhooks. Step 13.2. S starts giving dial tone to A Step 13.3. A dials the first digit in B_s subscriber number Step 13.4. S stops giving dial tone to A. Step 13.5. A dials the remaining three digits in B_s subscriber number Step 13.8. S starts giving busy tone to A Step 13.9. A hangs up Step 13.10. S stops giving busy tone to A
80	SRS41606	Functional	Service: call forwarding	Activation of call forwarding to a subscriber that has activated call forwarding shall be ignored by the system. This is regarded as an erroneous activation, and an error tone is given to the subscriber. (Motivation: together with SR41607, avoids call forwarding in closed loops)
111	SRS41804	Functional	Service interaction	The service call forwarding shall be deactivated if a customer removes either the subscriber from which calls are forwarded or the subscriber to which calls are forwarded.

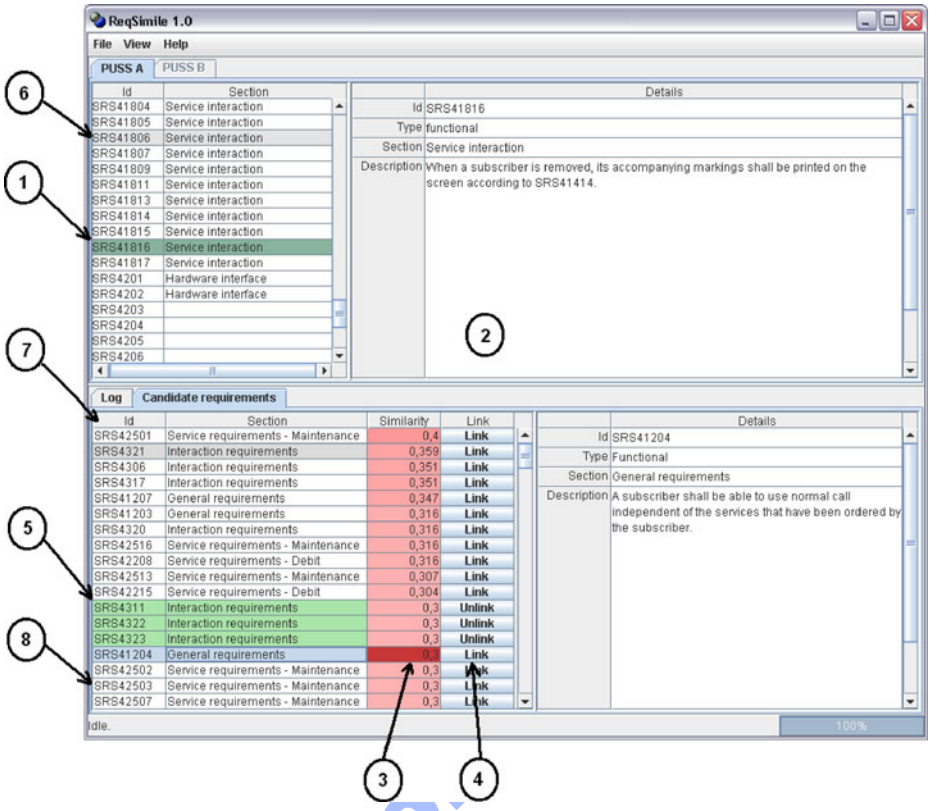

Fig. 2 The user interface of the ReqSmiliA tool used in the experiment. The full-size color figure can be found at http://fileadmin.cs.lth.se/serg/ExperimentPackages/ReplicationReqSimile/ReqSimiliA.bmp

writer, nor given to a native English language speaking, experienced requirements analyst for editing and rephrasing.

4.5 Tools

In this experiment, one tool remained unchanged from the original experiment while the other tool was changed. The tool that implements the *assisted* method, that is ReqSimile (Natt och Dag 2006b), was kept unchanged. The user interface of ReqSimile is presented in Fig. 2. The left side of the top pane of the window presents a list of requirements. The data has already been pre-processed by the tool so the user can start analyzing requirements for similarities. Selecting a requirement (1) makes the requirement's details display on the right (2) and a list of similar requirements in the other set appear in the bottom pane (7), sorted on the similarity value (3). Requirements that have already been linked in the set of analyzed requirements are highlighted using another (gray) color (6). Requirements that have been linked to the currently selected requirements (1) are highlighted using another (green) color (5). Unlinked requirements are not highlighted (8). Links can be made between the

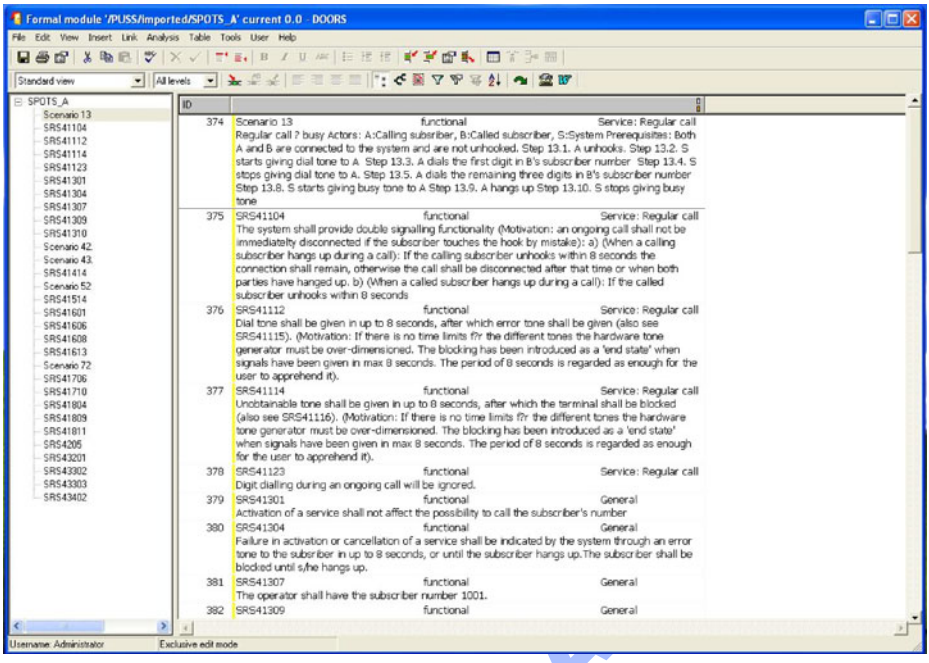

Fig. 3 The user interface of Telelogic Doors used in the experiment. The full-size color figure can be found at http://fileadmin.cs.lth.se/serg/ExperimentPackages/ReplicationReqSimile/DOORS.png

selected requirement (2) and the requirement with the associated link button (4). Once a requirement has been selected (1), the user has to perform two operations, click on the requirement that is supposed to be linked and click the button "link" to make the link.

The second tool, described by Natt och Dag et al. 2006 as ReqSimileM was changed in this experiment to Telelogic Doors (IBM 2010a). The user interface of Telelogic Doors is shown in Fig. 3. The two sets of requirements were opened in Doors from separated modules and placed next to each other on the screen. Figure 3 illustrates one of the requirements sets opened in a module. This orientation is similar to ReqSimile's view and enables easy visual comparing between the two sets of requirements. The finding and filtering capabilities were used in Telelogic Doors to perform the consolidation task. These capabilities can be accessed respectively from the Edit menu and the Find command or the Tools menu and the Filters command. The subjects were given detailed instructions with screen-shots of each step and each dialog window that was related to finding and filtering capabilities. After finding similar requirements, links were established using the tool's built in traceability solution. During the planning activities, it was discovered that making links in Telelogic Doors is not as straightforward as in ReqSimile, where only one mouse click is required. In this case, the user has to perform two operations in order to search for or filter out desired requirements. To make a link, the user must perform two operations: initiate the link on the source side and terminate the link on the destination side. The instructions for linking requirements in Telelogic Doors is available at http://www.—anonymized—.com. The subjects received no training in

using the tools other than the instruction given at the beginning of the experiment and some practice time, to get familiar with the tool.

4.6 Correct Consolidation

To enable measurement of the subjects' accuracy of linking requirements that are semantically similar, the original key for assigning correct links has been reused. This original key was created by the first author of the original experiment article (Natt och Dag et al. 2006), having many years of experience from this course in various roles. It is therefore justifiable to consider this key as one provided by an expert in the domain. The key was created a priori to any analysis of the subjects' assigned links in order to reduce any related validity threats. More information regarding the correct consolidation key, together with the distribution of the position at which the correctly similar requirements are placed by the tool in the ranked lists, is available in the original experiment article (Natt och Dag et al. 2006).

4.7 Instrumentation

In this experiment, most of the original experiment's guidelines were kept unchanged. In particular, the instructions for how to use the *assisted* method (ReqSimile tool) was reused. A new set of instructions describing how to use the *manual* method (Telelogic Doors) to find similar requirements and assign links between them, was developed and evaluated by an independent researcher. Since Telelogic Doors has a more complex user interface, the instructions were significantly longer than those for ReqSimile, consisting of eight pages of text and figures. Due to its length (eight pages), it was decided that subjects using Telelogic Doors should get more time to read through the instructions for that application. The pre- and post-test questionnaires were updated according to the changes made from the original study. In the pre-test questionnaire, the authors added one question about the experience using Telelogic Doors to be able to measure the impact of this phenomenon on the results. Furthermore, two questions related to English skills that were separated in the original design were merged into one. The rationale for this decision was that the subjects of the experiment will only read requirements so their skills in writing are not relevant. A pre-test questionnaire including five questions about the subjects' industrial experience in software development, experience with analyzing and revising requirements and possible knowledge and skills in Telelogic Doors was prepared. Before collecting the data, an additional experienced researcher evaluated the questionnaire to check the understandability of questions and their relevance for this study.

Due to a limited number of available computers in the laboratory room, the subjects were asked to work in pairs for the experiment. This deviates from the original experiment design, where subjects were performing the task individually and demands additional analysis to ensure that groups were formed equally. Some changes were also made to the post-test questionnaire. The original questions regarding (1) the time spent on the consolidation task, (2) the number of finished requirements, (3) the number of found duplicates, similar and new requirements and (4) the usefulness of used methods were kept unchanged. Moreover, two

questions about the scalability of used methods and possible improvements were kept unchanged comparing to the original experiment design.[3]

4.8 Data Collection Procedure

The data collection procedure was kept as similar as possible to the original experiment design. The subjects were given the introduction and problem description by the moderator. After the introduction, subjects were given some time to read through the assigned tool's instruction and make themselves familiar with its user interface. At this stage, the groups assigned to work with Telelogic Doors were given some extra time (approximately 5–10 min) since the tool interface was more complex and the instruction was longer. One of the important changes here is that the subjects answered pre-study test right before starting the task. The results of the pre-study survey were analyzed afterward and are presented in Section 4.2. Next, the subjects were asked to work for 45 min on the consolidation task. The results from the actual experiment were collected by analyzing the information recorded in the tools about the number of links made and requirements analyzed by the subjects. The experimental results were also checked against the results of the post-questionnaire to ensure consistency. The post-questionnaire was asked after performing the task.

4.9 Validity Evaluation

As for every experiment, questions about the validity of the results must be addressed. Threats to validity are presented and discussed using the classification of threats to conclusion, internal, construct and external validity as proposed by Wohlin et al. (2000).

Conclusion Validity In order to not have too low power of the statistical tests, parametric tests (t-test) were used after having investigated the normality of the data.

Subjects were selected from the same education program to limit their heterogeneity. Moreover, the questionnaire about the experience of subjects from industry, experience from the course where requirements originate from and experience in reviewing requirements was used to assess the heterogeneity of the subjects in these aspects. However, the threat related to the fact that subjects were asked to work in pairs may impact the conclusion validity. It affects in particular the random heterogeneity of subjects, since created pairs may manifest differences in industrial experience or experience from the previous courses. We discuss this threat in the analysis of the pre-study questionnaire results in Section 4.2 and the results achieved by the subject in relation to their experience in Section 7.1. The way how the subjects took seats in the laboratory room and thus the way how they were assigned to the methods can also be questioned here. As pointed out by Wilkinson (1999), random assignment is sometimes not feasible in terms of the control or measure of the confounding factors and other source of bias. Elements outside the experimental setting that may disturb the results were minimized. Students were not interrupted

[3]Both questionnaires are available at http://fileadmin.cs.lth.se/serg/ExperimentPackages/ReplicationReqSimile/PreTest.pdf and http://fileadmin.cs.lth.se/serg/ExperimentPackages/ReplicationReqSimile/PostTest.pdf.

during the experiment sessions and no significant noise was present. In order to minimize random irrelevance in experimental setting, the experiment moderators ensured that any discussions in pairs of subjects should be made as quietly as possible.

Although all subjects have taken the same education program for 2.5 years, the individual differences in industrial experience, experience in the course from which the requirements originate, and knowledge of English may affect the results. The searching for a specific result threat was addressed by not notifying the subjects which method is supposed to perform better than the other. The threat to the reliability of measurements is addressed by reusing the original measurements for the replication case. Moreover, all subjects received the same instruction for using the treatments which helps to standardize the application of treatments to subjects. Finally, the error rate of the significance level and the use of Bonferroni correction (Arcuri and Briand 2011) are among the threats to conclusion validity. The possibility of using the Bonferroni correction to adjust the level of significance is discussed in Section 7.

Internal Validity In this case, threats related to the history, maturation etc. have to be mentioned. The history threat to internal validity is minimized by applying one treatment to one object. Both days when the experiment sessions were held where normal working days not followed by any holidays (the first session took place on Tuesday and the second session on Friday). The maturation threat was minimized by dedicating only 45 min for the consolidation task (we assume that the subjects won't get bored by the task in 45 min). The instrumentation threat is addressed in two ways: (1) by reusing the original experimentation instrumentation, if no changes were needed, and (2) reviewing the instrumentation documentation by an independent researcher. However, since subjects were not divided into groups according to the results of the pre-study questionnaire (the questionnaire has been filled in right before the experiment's execution), the statistical regression threat can not be as easily addressed as in the original experiment. The analysis related to this threat is presented in Sections 4.2 and 7.

The incentives of participants are, next to their experience, an important factor that may influence the results of this study. According to the classification presented by Höst et al. (2005), both the original experiment and replication can be classified as I2 E1 where I2 means that the project is artificial (in terms of incentive). The subjects typically have no prior knowledge of the artifacts that they are working with and the requirements sets used in the experiment were developed by the researcher or borrowed from an industrial organization. The E1 level on the experience scale means that the subjects are undergraduate students with less than three months recent industrial experience, where recent means less than two years ago. Although the identical comparison of two I2E1 cases is not present in Höst et al. (2005), the example of two experiments classified as E1 I1 (where I1 means an isolated artifact) shows no significant difference in their outcomes. Moreover, three other pairs of experiments, classified in the same category, also shows the same outcomes (Höst et al. 2005).

The selection threat, as in the original design, may influence the results since the subjects are not volunteers and the laboratory session where the experiment was performed is a mandatory part of the course. The social threat to internal validity is addressed since the subject had nothing to gain from the actual outcome of the experiment; the grading in the course is not based on results of, or preparation

for, the experiment. Unlike the original experiment design, the experiment groups were not separated, however no information about which method is expected to perform better was revealed to the subjects. The possibility of looking at other subjects' results during the experiment execution was minimized by placing the subject in a way that separated each of the two treatments by the other treatment. Compensatory rivalry may be a problem in this case, since the group that will use the open-source solution (ReqSimile) or the commercial solution (Telelogic DOORS) may try to perform better to make their favor type of software win. This threat was addressed by explicitly stating in the beginning of the experiment that there is no favorite or assumingly better method. The potentially more problematic threat is that the subjects had to analyze and link requirements written in English when they had themselves used only Swedish to specify their own requirements in the domain. Further, the participants ability to objectively evaluate their skill in English language is a subject to question. In further work it would be interesting to execute the experiment on a set of native English subjects, also handling the original set of requirement to a native speaker experienced requirements analyst who will edit and rephrase them.

Construct Validity In this case, the theory is that the *assisted* method implemented in the ReqSimile tool provides better assistance for a particular task than the method implemented in Telelogic Doors. We base this theory on the fact that the *assisted* method provides a list of candidate requirements augmented with the degree of lexical similarity. As a result, the analyst can only look at a subset of possible candidate requirements (the most similar requirements, up to a certain threshold) thus saving time required to do the task. Moreover, we believe that the list of possible candidates helps the analyst to miss fewer requirements links and increase the precision of making the links. In contrast to the original experiment design, none of the authors have developed any of the tools. However, the originally mentioned threat related to the awareness of subjects about their own errors is still present in this experiment. This threat may have influenced the number of correct and faulty links. Also, as pointed out by Natt och Dag et al. (2006), when subjects know that the time is measured, it is possible that they become more aware of the time spent and the performance results may be affected.

The level of experience of the subjects, especially in reviewing requirements, may influence the outcome of the experiment. This threat to construct validity is addressed by the analysis and discussion of the results achieved by the subjects having experience in reviewing requirements in Section 7. Because the same 30 and 160 requirements were used by all subjects in both treatments and experimental sessions, this may result in a situation where the cause construct is under-represented. This threat is discussed in Section 7 where alternative designs for this experiment are outlined. Moreover, keeping the requirements sets unchanged opens up the possibility of discussing other factors as well as differences between the original and replicated experiment to assess their influence on the results. The interaction of different treatments threat to construct validity is minimized by involving the subjects in only one study. The differences in the user interfaces and their usability may have influenced the results achieved by the subject. In particular, this threat could influence the numbers of requirements analyzed and links assigned by the subjects. This threat has been addressed in two ways: (1) by providing detailed instructions

on how to make links in the tools and by giving the subjects as much time as they requested to get familiar and comfortable with the user interface, (2) by making the user interfaces look as similar as possible by placing the two requirements sets next to each other in Telelogic DOORS. Finally the evaluation apprehension threat is minimized by: (1) clearly stating that the performance of the subject has no effect on their grade in the course and (2) by using requirements that were not written by the subjects.

External Validity The largest threat in this category is the number of analyzed requirements. Since only a relatively small number of requirements was analyzed during the experiment, it is hard to generalize the results to large sets of requirements, which often is the case in industry projects (Berenbach et al. 2009; Leuser 2009). Using students as subjects is another large threat. Even though the subjects were on their last year of studies, they can be considered as rather similar to an ordinary employee. However, as mentioned by Kitchenham et al. (2002) students are the next generation of software professionals and they are relatively close to the population of interest. Since they participated in the requirements engineering course, they are familiar with the application domain. Finally, the diversity of experience of subject from industry and from analyzing and reviewing requirements, although hindering the conclusion validity, has a positive influence on the heterogeneity of the population sample used in this experiment.

The time spent on the task is also among potential threats to external validity. To analyze 30 requirements in 45 min subject should spend on average 90 s on each requirement. It remains an open question whether or not this is enough time for the subjects to perform both lexical and semantic analysis. However, this threat was partly addressed by stating at the beginning of the experiment that the subjects don't have to analyze all 30 requirements in 45 min (the subjects had to record how many requirements were analyzed and how they browsed the list of candidate requirements).

5 Experiment Execution

The replication was run in two two-hour laboratory sessions in January 2008. The first 15 min of each session were dedicated to the presentation of the problem. During this presentation, the importance of the industrial applicability of the results and the goal of the experiment were stressed. All students were given the same presentation. The general overview and differences between the included methods and tools were presented without favoring one method over the other. To avoid biasing, no hypotheses were revealed and it was made very clear that it is not known which approach will perform better. After the presentation of the problem, students were given time to get familiar with the instructions of how to use the tools and report that they are ready to start working on the task. The starting time was recorded in the questionnaire as students required varying times to get familiar with the instructions. After the 45 min time assigned for the task, subjects were asked to stop, record the number of analyzed requirements, the time, and to fill in the post-questionnaire. The remaining time was used for exchanging experiences and discussion the about tools used. This approach is similar to the original experiment execution described by

Natt och Dag et al. (2006). The difference from the original experiment is that subjects used both methods in both experimental sessions. Half of the subject pairs in each session (there were two sessions in total) were assigned to the assisted method while the other half to the manual method. The subjects were to use only one method during the experiment and participate in only one of the two sessions. The difference from the original experiment here is that the methods were not separated in different sessions. That is, in each session the pairs using the *assisted* and the *manual* methods were mixed.

After the presentation, the subjects were assigned to the methods. Because only one laboratory room could be used for each session and this room did not have enough computers for all subjects (which was not known while planning the experiment), the subjects were asked to work in pairs. Each pair was randomly assigned to the method later used for the consolidation task. There were no name tags or other indicators of the method on the laboratory desks when subjects took their seats in the laboratory room. Therefore, subjects could not take a preferable method seat or be attracted by the name on the desk. Subjects were asked to discuss the solutions only within their own pair. Since the nearest group was not using the same method, the possibility of comparing or discussing results was avoided. The subjects were allowed to ask questions of the moderator, if they experiences any problems. Only answers related to the difficulties of using tools were given in a straightforward manner. No answers related to assessing similarity between requirements were given. The material used in the experiments comprised:

- The ReqSimile application with automated support of similarity calculations and a database containing: (1) 30 randomly selected requirements from the first set, (2) all 160 requirements from the second set. These requirements should be browsed through by the subjects.
- The Telelogic Doors application with the same two sets of requirements imported into two separated modules. The application's graphical user interface was set as presented in Fig. 3 in order to make it as similar to the ReqSimile user interface as possible.
- The documentation comprising: (1) An industrial scenario describing the actual challenge (one page). (2) A general task description (one page). (3) Detailed tasks with space for noting down start and end times (one page). (4) A short FAQ with general questions and answers about the requirements (one page). (5) A screen shot of the tool user interface with the description of the different interface elements in the ReqSimile case (one page) or a eight pages instruction with screen shots from the steps needed to analyze requirements and make links using Telelogic Doors.
- The instruction to the students was as follows: (1) Review as many of the requirements as possible from the list of 30 requirements shown in the tool. For each reviewed requirement, decide if there are any requirements in the other set that can be considered identical or very similar (or only a little different) with respect to intention. (2) Assign links between requirements that you believe are identical or very similar. (3) Note down the start and finish time. (4) When finished, notify the moderator.

Given the experience from the original study, it was decided to dedicate 45 min to the consolidation task. The subjects were notified about the time left for the task both

15 and 5 min before the end of the lab session. After approximately 45 min, subjects were asked to stop working on the task unless they, for any reason, spent less than 40 min on the task. All students were asked to fill in a the post-test questionnaire described in Section 4.7. Apart from noting the finishing time and the number of analyzed requirements, subjects were also asked to assess the usefulness of used methods in terms of the given task and, if applicable, propose improvements. Right after executing the experiment, it was known which data points had to be removed due to tool problems or subjects' attitude. Three groups had problems with the tools used which resulted in loss of data and one group performed unacceptably analyzing only three requirements during 45 min and making only two links. These four groups were treated as outliers and removed from the analysis.

6 Experiment Results Analysis

In this section, methods of analyzing the results are described. In order to keep the procedures as similar to the original experiment design as similar as possible, the same statistical methods were used to test if any of the null hypotheses can be rejected. Additional analysis was also performed in order to assess if working in pairs influences the performance of subjects. Hypotheses were analyzed separately, while

Table 5 The results from measuring dependent variables

Pair of subjects	T (min)	N	Links assigned	Correctly linked (N_{cl})	Correctly not linked (N_{cu})	Incorrectly linked (N_{il})	Missed (N_{iu})
M1	38	12	6	1	4	5	6
M2	41	13	10	5	4	5	3
M3	46	30	15	11	10	4	9
M4	44	13	10	5	4	5	3
M5	45	18	16	8	11	8	12
M6	48	20	5	2	6	3	9
M7	45	22	21	6	6	15	8
M8	44	19	9	4	7	5	8
M9	46	19	15	7	5	8	5
M10	45	16	9	4	4	5	5
M11	45	18	?	?	?	?	?
A1	41	14	13	5	5	8	5
A2	45	20	25	9	4	16	3
A3	49	18	19	5	3	14	6
A4	45	13	25	5	0	20	3
A5	50	20	15	8	4	7	4
A6	50	21	19	6	3	13	6
A7	29	3	11	1	0	12	0
A8	44	30	?	?	?	?	?
A9	34	30	23	13	7	10	7
A10	41	30	16	12	8	4	8
A11	50	23	19	7	7	12	7
A12	35	30	20	13	8	7	7

The *rows highlighted gray* indicate data points that were removed from the analysis (outliers)

any relations and accumulated results are presented in Section 7. The standard t-test has been used, as the data was confirmed to have a normal distribution. Just as in the original experiment from which requirements were reused, in this experiment one analyzed requirement can be linked to several others. The results from measuring dependent variables can be found in Table 5. Subjects that used ReqSimile are marked with the letter A (as an abbreviation of the *assisted* method), and subjects that used Telelogic Doors with the letter M (as an abbreviation of the *manual* method). The dependent variables are described in Section 4.1. Table 5 presents the results from both experimental sessions (Tuesday and Friday sessions).

Rows M10, M11, A7 and A8 in Table 5 represent data points that were removed from the analysis. The pair M10 was removed from the results due to the inconsistency between the results stated in the post-task questionnaire and the results saved in the tool. The pair M11 was removed due to loss of data. Similar problems caused the authors to remove group A8 from the analysis since the links were not saved in the tool. Finally, group A7 was removed due to their lack of their commitment to the task.

The time spent on the task is presented in column 2 of Table 5. The results for the number of finished requirements, derived from the post questionnaire and confirmed with the results recorded in the tool used, are listed in column 3. Next, other dependent variables values are presented in the remaining columns. The values were calculated based on the results saved in the tools and from the answers to the questionnaires questions.

The results for the number of analyzed requirements per minute are depicted as a box plot in Fig. 4. It can be seen that there is no statistically significant difference in the number of analyzed requirements between the *manual* and the *assisted* method. The group that used the *manual* method analyzed on average 0.41 requirements per minute while the group that used the *assisted* method analyzed on average 0.51 requirements per minute. In this case, we observe that the medians are most likely equal, while the lower and upper quartiles values differ significantly. The t-test gave a p-value of 0.20 which gives no basis to reject the null hypothesis H_0^1. The notches of

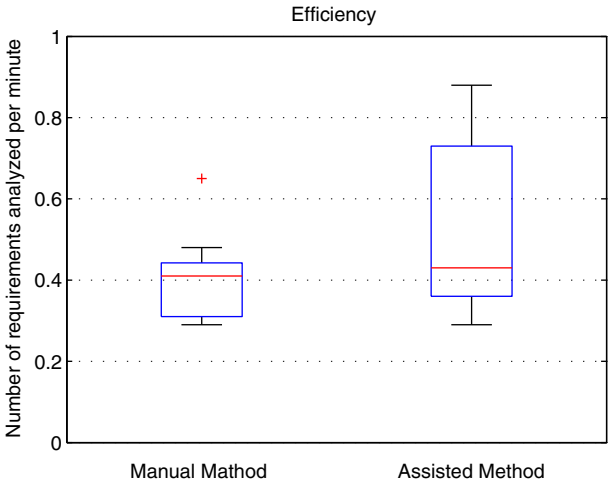

Fig. 4 The results for the number of analyzed requirements

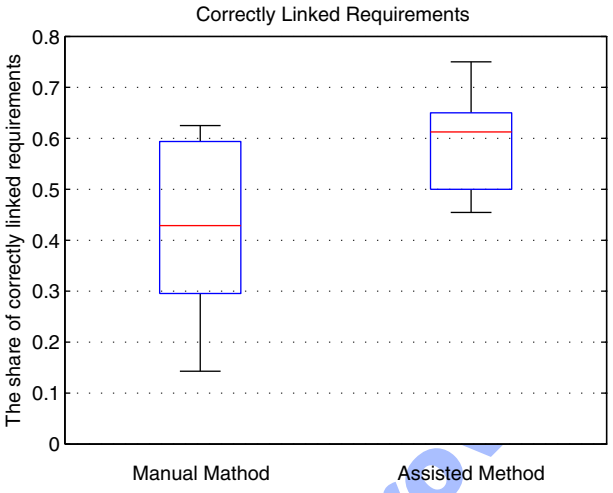

Fig. 5 The results for the share of correctly assigned links

the box plot overlap. To summarize, the *assisted* method turned out not to be more efficient in consolidating two requirements sets than the *manual* method (research question Q1a).

The results for the number of correct links assigned by subjects are depicted in Fig. 5. The group that used the *assisted* method correctly assigned on average 58% of the links that the expert assigned, while the group that used the *manual* method correctly assigned on average 43% of the correct links. The medians differ significantly from 61% for the *assisted* method to around 42% for the *manual* method. The t-test gave in this case the p-value 0.013 which makes it possible to reject hypothesis H_0^2. Thus, we can state a positive answer to research question Q1b, the *assisted* method is superior to the manual method when consolidating two requirements sets when measured by which method delivers more correct links (Fig. 6)

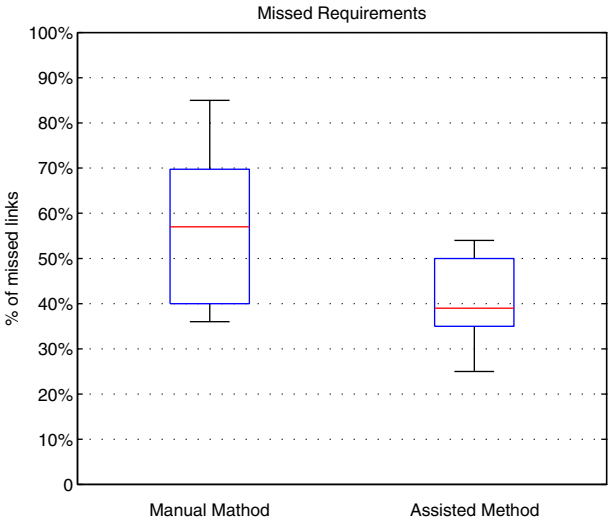

Fig. 6 The results for the percentage of missed links

Table 6 The results of the t-tests for original and replicated experiments

Hypotheses	The p-values in the original study (Natt och Dag et al. 2006)	The p-values in this replication
H_0^1 efficiency	0.0034	0.20
H_0^2 correct links	0.0047	0.013
H_0^3 missed links	0.0047	0.02
H_0^4 incorrect links	0.39	0.14
H_0^5 precision	0.39	0.62
H_0^6 accuracy	0.15	0.72

To address hypothesis H_0^3, requirements analyzed by each pair of subjects were reviewed, and the number of links that should have been assigned but were not, was calculated. In the case when subjects did not analyze all requirements, only requirements that had been analyzed were taken into consideration. Each pair of subjects stated in their post-test questionnaire how many requirements were analyzed and how they had worked through the list of requirements. This information was used to correctly count the number of missed links. The results are depicted in Fig. 5. The group that used the *assisted* method missed on average 41% of the links, while the group that used the *manual* method missed on average 57% of the links. The medians in this case are 38% for the *assisted* method and 57% for the *manual* method. The t-test gives a p-value of 0.0207 which means that we can reject H_0^3 and confirm the original experiment's conclusions by making the conjecture that the *assisted* method helps the subjects to miss significantly fewer requirements links than the *manual* method (research question Q1c).

For the number of incorrectly assigned links (N_{il}) (related to research question Q1), the t-test resulted in a p-value of 0.14, so the hypothesis H_0^4 cannot be rejected. Furthermore, for the hypothesis H_0^5 (related to research question Q1) the t-test gave the p-value 0.62 and for the hypothesis H_0^6 (related to research question Q1) the t-test resulted in the p-value 0.72. To summarize, research question Q1 can be answered as "yes", for some aspects. Our results confirm the results from the original experiment for correctness and number of missed links but we can't confirm the result for the efficiency. As in the original experiment the hypotheses H_0^4, H_0^5, H_0^6 could not be rejected here. Compared to the original experiment, this experiment confirms no statistical difference in the number of incorrect links, precision and accuracy between the two analyzed treatments. The question regarding different results for H_0^1 is discussed in Section 7. The summary of the original and the replicated experiments is depicted in Table 6.

7 Experiment Results Interpretation and Discussion

This section presents an interpretation of the results presented in Section 6. Since this experiment was conducted on a set of students, it is important to emphasize here that the results from this study are interpreted in the light of the population where the experiment was held (Kitchenham et al. 2002). The section discusses the results of this experiment in isolation as well as in relation to the results of the original experiment.

7.1 Interpretation of this Experiment

The results achieved in this replicated experiment allow for the rejection of two out of six stated null hypotheses (see Table 6). As already mentioned, four data points were removed from the analysis for various reasons, as described in Section 6. The results achieved by group A7 show that the subjects were not motivated to do the task or misunderstood the task (they analyze only 3 requirements in 29 min). It is surprising since both subjects in this pair had one or more years of industrial experience (pairs with similar experience (A3 and M4) performed significantly better on this task), but no experience in reviewing requirements. It is an open question how they could influence the results if properly motivated. Similarly, due to unexpected tool issues (the results were not saved in the tool) we can only assume that the results achieved by groups A8 and M11 could positively influence the results for both assisted and manual method (group A8 achieved efficiency of 0.68 requirement per minute, group M11 0.4 requirement per minute and group M10 0.35 requirements per minute). Adding these three incomplete data points (A8, M10 and M11) to the analysis of the performance will not change the result of the hypothesis H_0^1 testing (although it can minimize the p-value to 0.0890).

As for the H_0^1 (performance) hypothesis (research question Q1a), the lack of a statistically significant difference can be interpreted in the following way: we have not found this property (namely lower efficiency of the manual method comparing to the assisted method) on a different requirements management tool, namely Telelogic DOORS. The lack of statistical significance can be explained by a rather large variation in the *assisted* method (the minimum value for the performance is 0.29 requirement per minute while the maximum value is 0.89 requirement per minute). Furthermore, albeit the medians are almost identical for both the *assisted* and the *manual* method with respect to the performance, the range of the third quartile is much larger in the *assisted* method. This fact can be interpreted in favor of practical significance (Kitchenham et al. 2002) in the following way: if we assume that both groups assigned to the methods are rather homogeneous, we can also assume that in both groups there are similar numbers of more and less motivated subjects. In the light of the fact that motivation has been reported to be an important determinant of productivity and quality of work in many industries (Baddoo et al. 2006), the practical significance of the results is that the *assisted* method gives the possibility to achieve higher values of the performance than the *manual* method. As more motivated subjects usually achieve better results with a given task, we can assume that the top scores for both methods correspond to the most motivated pairs of subjects. The evidence reported by Baddoo et al. (2006), albeit studied on developers rather than requirements engineers, confirms that the traditional motivators of software developers, e.g. intrinsic factors, but also opportunity for achievement, technically challenging work and recognition have a strong influence on the developer's performance. Thus, when comparing the top score of both methods, we could conclude that the *assisted* method may boost the performance of the motivated subjects more than less motivated subjects.

The analysis of the results for efficiency versus the experience of the subjects revealed that the subjects with experience reviewing requirements (pairs A1, A3 and A11) were not the fastest (the values were lower or around the median value). We can postulate here that the industrial experience led these pairs to be more cautious

Table 7 The analysis of the industrial experience of the subjects in relation to their results

	All data		Assisted method		Manual method	
	Exp.	Unexp.	Exp.	Unexp.	Exp.	Unexp.
Efficiency H_0^1 [N/T]	0.44	0.48	0.51	0.52	0.40	0.42
Correct H_0^2 [%]	44	55	53	61	39	47
Missed H_0^3 [%]	55	44	46	38	60	52
Incorrect H_0^4 [%]	39	54	40	66	38	33
Precision H_0^5 [%]	45	43	50	41	42	46
Accuracy H_0^6 [%]	46	43	49	40	44	48

The *cells colored gray* indicate cases where industrial experience had positive effect on the analyzed aspect

when analyzing requirements. On the other hand, the top score in this group (A9) was achieved by a pair of subjects that reported no industrial experience and no experience from the course from which the requirements originated. Surprisingly, the two lowest values of performance were achieved by the pairs having either one year of industrial experience, including experience with reviewing requirements (pair A3), or experience from the course from which the requirements originated (pair A4). In the light of these facts, we can not draw any strong conclusions about the effect of both industrial experience and experience with reviewing requirements on the performance of subjects. However, our results show indications of negative impact of experience on the performance of the subjects. The full analysis of results of experienced versus inexperienced subjects is presented later in this section and in Tables 7 and 8.

The results concerning the number of correct links (research question Q1b) can be interpreted as follows. The group that used the *assisted* method assigned on average 58% of the correct links, while the group that used the *manual* method assigned on average 43% of the correct links. The results of the t-test allows us to reject H_0^2. This fact may be interpreted in the following way in favor of the *assisted* method: even if the *assisted* method is put next to a rather sophisticated requirements management tool, it can still provide better support for assessing more correct links between requirements. The fact that both in the original and the replicated studies the *assisted* method provided a better support in linking similar requirements may lead to the

Table 8 The analysis of the experience from the course where requirements originate from

	All data			Assisted method			Manual method		
	E	E and U	U	E	E and U	U	E	E and U	U
Efficiency H_0^1 [N/T]	0.45	0.39	0.57	0.53	0.38	0.69	0.37	0.40	0.46
Correct H_0^2 [%]	56	51	46	68	53	58	46	48	34
Missed H_0^3 [%]	50	57	34	56	59	31	44	55	37
Incorrect H_0^4 [%]	58	54	30	85	58	56	31	50	27
Precision H_0^5 [%]	42	39	50	40	37	56	45	42	44
Accuracy H_0^6 [%]	43	41	49	39	39	53	47	45	46

E denoted that both pair members are experienced, *E and U* denoted one experienced and inexperienced person working together and *U* denoted that both pair members were inexperienced

following two interpretations: (1) the method is better in this matter, and (2) working in pairs has a minimum or equal impact on the two methods when it comes to the number of correctly linked requirements.

The results for the number of missed requirements links (research question Q1c) confirm the results of the original experiment. The t-test confirms that the *assisted* method can help to miss fewer requirements links than the *manual* method. Missing fewer links may be important when large sets of requirements have to be analyzed, which is a reasonable practical interpretation of this result. This result also confirms the interpretation that in the case of the *assisted* method, showing a list of similar requirements candidates limits the solution space for the analyst which results in a smaller number of missed requirements links.

Similarly to the original experiment, the results from the experiment can also not reject hypotheses H_0^4, H_0^5 and H_0^6 (research question Q1). The lack of statistically significant differences in these cases may be interpreted as the possible existence of additional factors that affect the consolidation of requirements process which were not controlled in the experiment. For example, since it is much easier to make a link in ReqSimile than in Telelogic DOORS this may affect the number of incorrect links, precision and accuracy. This threat to construct validity is described in Section 4.9 and is considered as one of the topics for further work.

The fact that subjects worked in pairs may also influence the results. Even though working in pairs has generally been considered having a positive impact on the task, for example in pair programming (Begel and Nachiappan 2008), the results among researchers are inconsistent (Hulkko and Abrahamsson 2005; Parrish et al. 2004). Therefore, assessing the impact of working in pairs in more decision-oriented software engineering tasks is even more difficult. Thus, it can be assumed that working in pairs may sometimes influence the performance of these types of tasks positively, and sometimes negatively. In this case, we assume that subjects were similarly affected by this phenomenon both in the *assisted* and in the *manual* method.

The influence on the results of fluency in reading and reviewing requirements in the English language can be interpreted in the following way. Since subjects reported either "very good knowledge" or "'fluent knowledge' in reading and writing English our interpretation of this fact is that this aspect equally influenced all subjects. Moreover, the subjects were allowed to ask questions for clarification, including understanding the requirements during the experiment. However, it remains an open question what can be the results of the experiment when performed on native English language speaking subjects.

The analysis of the influence of the industrial experience of the subjects of the results achieved is depicted in Table 7. The data has been analyzed for all pairs of subjects, as well as for subjects using the same method. Four out of the total 11 pairs of subjects using the *assisted* method reported having some industrial experience. For the *manual* method, 5 out of 9 pairs of subjects included in the data analysis reported having some industrial experience. Subjects were paired in a way that minimizes the difference in the experience of pair members. Moreover, only in two cases (pairs M5 and M6) were pairs formed of one experienced and one inexperienced subject (see Section 4.2 for more detail about the subjects).

The analysis of the relationship of industrial experience to the results is based on the arithmetic average values of the results achieved. Table 7 shows that in most cases industrial experience negatively influenced the results (and tested hypotheses). The

cells colored gray in Table 7 indicate cases where industrial experience has a positive effect on the analyzed aspect. For all hypotheses for subjects using the *manual* method, the industrial experience had a negative impact on the results achieved. In the case of the *assisted* method, the experienced subjects made fewer incorrect links and had better precision and accuracy. The results from comparing all subjects show the same pattern as the results for the *assisted* method. While the results are implausible, they may be an indicator that general software engineering industrial experience may not be useful in the task of analyzing requirements, at least when the experience is minimal to small. Thus, we state a hypothesis that experience in reviewing requirements and the domain knowledge should significantly help in achieving better results by our subject. Keeping in mind the scarceness of the data, we provide some examples that we used to support the hypothesis in the text that follows.

Six pairs of subjects reported having experience analyzing and reviewing requirements outside of the courses taken in their education. In one case (M8), a person with more than one year of industrial experience was paired with a person with no experience. Although it may be expected that an experienced pair member can significantly influence the results of this pair, the efficiency of this pair was only 0.02 higher than the median value of the efficiency for the subjects using the *manual* method. The number of correct links is 10% lower, the number of incorrect links is 10% higher while precision and accuracy are very close to the average values for the entire group using the *manual* method. In the case of pair M7, a person with experience only from courses was paired with a person with less than a year of industrial experience in analyzing and reviewing requirements. The results achieved by this pair are higher than the average in terms of efficiency (7% higher), close to the average for the share of correct links, and below the average for the remaining attributes (30% more incorrect links than the average, 16% lower than the average for the precision and 12% lower than the average for the accuracy). The remaining 3 pairs of subjects (A1 and M5) were composed of a person with only academic experience with a person with less than a year of industrial experience. Pair M5 analyzed 0.4 requirement per minute which is close to the average value (0.41), achieved 40% of correct links (the average value is 43% for this group) and 44% of incorrect links. The precision achieved by pair M5 is 50% which is 4% higher than the average. When it comes to the accuracy, pair M5 achieved 46% accuracy (the average value was 46%). The results for efficiency for pairs (A1, A3 and A11) were below or about the average values, and these data points could have been partly responsible for the fact that hypothesis H_0^1 could not be rejected. The results for these pairs for the number of correct links and the share of missed links were also below the average and the median values. The results for the number of incorrect links were around the mean value and above the median value. Finally, the results for the precision and accuracy were below the median values. To summarize, the influence of experience in analyzing and reviewing requirements can't be clearly defined and statistically confirmed, as subjects report both results above and below the average values.

As the last step of the analysis, we investigate whether prior experience in the course that originated the requirements in some manner influences the results achieved by the subjects. We can assume that this experience can somehow be compared to specific domain knowledge. In the course model, all team members are actively involved in analyzing and reviewing requirements. Thus, we only distinguish

between subjects that took and did not take the course. For six pairs of subjects, both subjects have experience from the course, for seven other pairs only one pair member had experience from the course. Finally for six pairs, both subjects reported no experience from the course. The pairs where both members had experience and where both members had no experience were equally distributed between the methods (three pairs for each method). The *assisted* method had four pairs with experience and lack of experience. The analysis is depicted in Table 8. Pairs where both pair members had experience from the course are abbreviated with the letter "E", where only one pair member had experience are abbreviated with "E and U" and where none of the two pair members had any experienced from the course is abbreviated with the letter "U".

Analyzing the differences between the average results for all three sub-groups for both methods we can see that the expected behavior can only be confirmed for the number of correct links. Pairs with experience in the course achieved better correctness than inexperienced pairs, independent of whether one or both members had the experience. Another interesting observation here is that inexperienced subjects missed on average only 34% of requirements links, while experienced subjects respectively missed 50% (both pair members experienced) and 57% (when one of the pair members had some experience). The lowest average precision and accuracy levels were recorded for pairs where one pair member had experience from the course from which the requirements were taken. The analysis of the pairs working with the same method confirms the analysis of all data points. For the *manual* method experienced pairs turned out to be more correct than the inexperienced pairs. For the *assisted* method, pairs where both members where experienced were more correct, but pairs where only one member had experience where not as correct as the inexperienced pairs. Finally, the pairs where only one person was experienced performed worse in all aspects for both methods analyzed than the pairs where both persons were experienced.

7.2 Interpretation of the Results from both Experiments

In this section, we provide the analysis and discussion of the results achieved in both the original and the replicated experiments. We have used standard t-tests to test if there are any significant differences between the two experiments. From the results of the t-tests between the same methods depicted in Table 9, we can see no significant difference for any of the cases (research question Q2). However, some interesting

Table 9 The results of the t-tests for the original and the replicated experiments for the same methods

Hypotheses	Assisted old/new (p-value) (research question Q2a)	Manual old/new (p-value) (research question Q2b)
H_0^1 Efficiency	0.48	0.27
H_0^2 Correct links	0.93	0.30
H_0^3 Missed links	0.37	0.20
H_0^4 Incorrect links	0.21	0.73
H_0^5 Precision	0.81	0.45
H_0^6 Accuracy	0.90	0.41

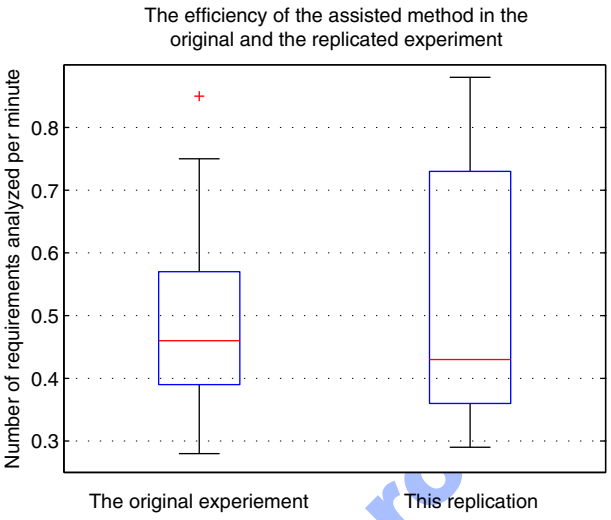

Fig. 7 The result of comparing the efficiency of the *assisted* method in two experiment sessions

differences between the two experiments for the efficiency of the subjects using the *assisted* method can be seen from the box-plot visualization in Fig. 7. As can be seen in Fig. 7, the results for the efficiency of the *assisted* method in this experiment have a much larger range of values, which may be the reason why the hypothesis H_0^1 could not be rejected (research question Q2a). As described in Section 7.1 the two lowest values of performance were achieved by the pairs having either one year of industrial experience, including experience from reviewing requirements (pair A3), or experience from the course from which the requirements originate (pair A4).

However, one of the possible explanations for the difference between the performance in this experiment and in the original experiment differs may be that more advanced searching and filtering functionalities have been used in the current *manual*

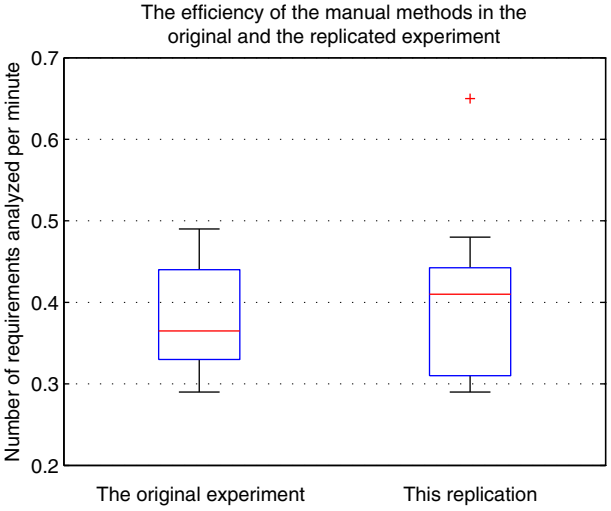

Fig. 8 The results of the efficiency achieved by the manual method in the original and the replicated experiments

method. Contrary to the original experiment, the *manual* method in this experiment uses advanced searching and filtering functionalities which may (to some extent) be comparable to the lexical similarity analysis because they also present only a subset of analyzed requirements to the analyst. The analyst using the filtering and searching functionality has to provide a meaningful search string to filter out similar requirements, while in the lexical similarity case the analysis is done automatically. Our interpretation is supported by Fig. 8 which depicts the results of the performance for the *manual* method between the original and the replicated experiments. The median value for this replication study is 0.41 and is 0.05 higher than in the original experiment (0.36). However, the second quartile has more diverse values than in the original experiment. Moreover, we can assume that the filtering method has a higher degree of uncertainty which is shown by the results for the accuracy.

7.3 Discussion

In this section, we discuss alternative designs for this experiment as well as the differences between the two experiment sessions (the replication has been run in two sessions). Using alternative design could be beneficial for the comparative analysis of the original and the replicated experiment (RQ2). For example the paired t-test could have been used to support comparison between this replication study and the original study (Kachigan 1991). However, the it remains an open question if paired units are similar with respect to "noise factors" for both the *assisted* and the *manual* methods used. It could also have been beneficial to the validity if the subjects answered the pre-questionnaire before running the study and were then assigned to treatments based on the result of this questionnaire. The manual analysis of the two experiment sessions did not reveal any significant differences between the Tuesday and the Friday sessions. However, to fully address this threat to validity, additional statistical tests should be used. Finally, changing the design of the study to use random samples of 30 and 160 requirements for each subject, generated from a much large dataset of requirements is one of the options for further work.

During this experiment six hypotheses were tested using the same data set, and more tests were performed comparing the data from the original and replicated experiments. The result of performing multiple comparisons on the same data is increased probability of Type I error, which in case of only one comparison is equal to the obtained p-value (Arcuri and Briand 2011). Thus, the Bonferroni correction should be discussed here. In this case, we performed 18 tests, 6 tests comparing the assisted and manual method (to answer the research question Q1), 6 tests comparing the old/new experiments with regard the assisted method (to answer the research question Q2a) and 6 tests comparing the old/new experiments with regard to the manual method (to answer the research question Q2b). This yields a significance level of $0.05/18 = 0.0027$ according to Bonferroni. In this case it is no longer possible to reject hypotheses H_0^2 and H_0^3. The correction has no impact on the results of the tests between the original and the replicated experiments. However, the correction has not been used in the original experiment and has been criticized by a number of authors (Arcuri and Briand 2011; Perneger 1998; Nakagawa 2004) where some of them do not recommend using the Bonferroni adjustment (Arcuri and Briand 2011). In the light of this criticism, it is an open question for this work as to whether or not this correction should be used. Therefore, we report the obtained p-values for all

performed tests in case the readers want to evaluate the results using the Bonferroni correction or other adjustment techniques (Arcuri and Briand 2011).

The relationship between the efficiency and the practical experience of the subjects may have been investigated using multivariate analysis. For example, understanding if the efficiency of the subjects was related to their accuracy could have been investigated by recording the efficiency of linking requirements at random as a reference point. Since this has not been done, we consider this analysis as possible future work and thus outside the scope of this article.

8 Conclusions

Large market-driven software companies face new challenges that emerge due to their extensive growth. Among those challenges, a need for efficient methods to analyze large numbers of requirements, issued by various customers and other stakeholders, has emerged (Regnell and Brinkkemper 2005). The result is an increasing effort dedicated to analyzing incoming requirements against those requirements already analyzed or implemented. This task is also called requirements consolidation. The core of the requirements consolidation process is finding the similarities between requirements and recording them by making links between them (Natt och Dag et al. 2006).

In this paper, we present a replicated experiment that aims to assess whether a linguistic method supports the requirements consolidation task better than a searching and filtering method. In this experiment, two methods implemented in two different tools were compared for the requirements consolidation task. The *assisted* method, which utilizes natural language processing algorithms to provide a similarity list for each analyzed requirements, was compared with the *manual* method, which utilizes searching and filtering algorithms to find similar requirements. After deciding which requirements were similar, the subjects assigned links between the requirements. The conclusions of this paper are as follows:

- Subjects using the *assisted* method were statistically not more efficient in consolidating requirements than the subjects using the *manual* method (research question Q1a), which is a different result compared to the original study
- The *assisted* method was confirmed as significantly more correct in consolidating requirements than the *manual* method (the manual method was changed from the original experiment) (research question Q1b), which is inline with the original study.
- The *assisted* method helps to miss fewer requirements links than the *manual* method (research question Q1c), which is the same result as in the original study.
- The hypotheses that could not be rejected in the original study (in terms of the number of incorrect links, precision and accuracy related to research question Q1) could also not be rejected in this experiment. Further investigation is required to understand the reasons for these results.
- The analysis of the results achieved for the same method (assisted or manual) between the original and the replicated study shows no significant difference in any of the cases. However, some differences in favor of the searching and filtering method have been observed between the results of the performance of the subjects using the *manual* methods.

To summarize, for two of our hypotheses the results reported in this replication study confirm the results achieved in the original experiment. The first confirmed hypothesis (H_0^2) is that the *assisted* method helps to make more correct links than the *manual* method. The second confirmed hypothesis (H_0^3) indicates that the *assisted* method helps to miss fewer requirements links than the *manual* method. The statistical significance in performance of the *assisted* method achieved over the *manual* method (hypothesis H_0^1) is not confirmed in this study. The remaining three hypotheses regarding the number of incorrect links (hypothesis H_0^4), precision (hypothesis H_0^5) and accuracy (hypothesis H_0^6) could not be rejected, which is the same situation as reported in the original experiment (Natt och Dag et al. 2006). In order to investigate the possible reasons for the difference in the remaining case, this paper provides a cross-case analysis of the same methods across the two experiment sessions as well as detailed analysis of the relations between the experience of the subjects and their results.

The analysis revealed that the pairs of subjects with experience in the course that originated the requirements achieved better correctness than inexperienced pairs, independent of whether one of both members had the experience. At the same time, the pairs of subjects without any experience missed on average fewer requirements links than the experienced pairs of subjects. Furthermore, the pairs where only one person had experience performed worse in all aspects than the pairs where both persons were experienced. The analysis revealed no statistical difference for any of the cases (which refers to the research question RQ2). However, the analysis of the efficiency of the subjects using the *assisted* method in the two experiments, depicted in Fig. 7, revealed that the values for the efficiency achieved in this replication have a much higher range. The results for the efficiency of the manual methods in the two experiments, depicted in Fig. 8, shows similar range of values, but different medians. It should be noted that there are validity threats to the study as described in Section 4.9, e. g. experience of the subjects and their incentives, the number of subjects participating, requirements used in the experiment, and the multiple comparison threats.

Performing a third experiment with experienced practitioners and requirements sets from industry is left for future work. Moreover, it would be interesting to further investigate the influence of working in pairs on the requirements consolidation task as well as to analyze the influence of the construction of the user interfaces on the efficiency of correctness of the subjects.

Acknowledgements This work is supported by VINNOVA (Swedish Agency for Innovation Systems) within the UPITER project. Special acknowledgments to Richard Berntsson-Svensson for participating in the pilot study and reviewing the paper. We are also thankful to Lars Nilsson and David Callele for reviewing the paper and excellent language comments.

References

Aguilera C, Berry D (1991) The use of a repeated phrase finder in requirements extraction. J Syst Softw 13:209–230. doi:10.1016/0164-1212(90)90097-6

Antoniol G, Canfora G, Casazza G, De Lucia A, Merlo E (2002) Recovering traceability links between code and documentation. IEEE Trans Softw Eng 28(10):970–983. doi:10.1109/TSE.2002.1041053

Arcuri A, Briand L (2011) A practical guide for using statistical tests to assess randomized algorithms in software engineering. In: Proceeding of the 33rd international conference on Software engineering. ACM, New York, ICSE '11, pp 1–10. doi:10.1145/1985793.1985795
Baddoo N, Hall T, Jagielska D (2006) Software developer motivation in a high maturity company: a case study. Software process: improvement and practice, pp 219–228
Basili V, Shull F, Lanubile F (1999) Building knowledge through families of experiments. IEEE Trans Softw Eng 25(4):456–473. doi:10.1109/32.799939
Begel A, Nachiappan N (2008) Pair programming: what's in it for me? In: ESEM '08: proceedings of the second ACM-IEEE international symposium on empirical software engineering and measurement. ACM, New York, pp 120–128. doi:10.1145/1414004.1414026
Berenbach B, Paulish DJ, Kazmeier J, Rudorfer A (2009) Software & systems requirements engineering: In: Practice. Pearson Education Inc.
Breaux TD (2009) Exercising due diligence in legal requirements acquisition: a tool-supported, frame-based approach. Atlanta, GA, pp 225–230. doi:10.1109/RE.2009.46
Cleland-Huang J, Chang CK, Ge Y (2002) Supporting event based traceability through high-level recognition of change events. In: COMPSAC '02: proceedings of the 26th international computer software and applications conference on prolonging software life: development and redevelopment. IEEE Computer Society, Washington, pp 595–602
Cleland-Huang J, Settimi R, Duan C, Zou X (2005) Utilizing supporting evidence to improve dynamic requirements traceability. In: Proceedings of the 13th IEEE International Conference on Requirements Engineer (RE 2005), pp 135–144
Cleland-Huang J, Berenbach B, Clark S, Settimi R, Romanova E (2007) Best practices for automated traceability. Computer 40(6):27–35. doi:10.1109/MC.2007.195
Cleland-Huang J, Czauderna A, Gibiec M, Emenecker J (2010) A machine learning approach for tracing regulatory codes to product specific requirements. In: ICSE '10: proceedings of the 32nd ACM/IEEE international conference on software engineering. ACM, New York, pp 155–164. doi:10.1145/1806799.1806825
ECTS (2010) The ECTS grading systems defined by European Commission. http://en.wikipedia.org/wiki/ECTS_grading_scale. Accessed 7 Sept 2011
Fabbrini F, Fusani M, Gnesi S, Lami G (2001) An automatic quality evaluation for natural language requirements. In: Proceedings of the 7th international workshop on Requirements Engineering Foundation for Software Quality (REFSQ 2001), pp 4–5. doi:10.1.1.9.7525
Fantechi A, Gnesi S, Lami G, Maccari A (2003) Applications of linguistic techniques for use case analysis. Requir Eng 8(3):161–170. doi:10.1007/s00766-003-0174-0
Fricker S, Gorschek T, Byman C, Schmidle A (2010) Handshaking with implementation proposals: negotiating requirements understanding. IEEE Softw 27:72–80. doi:10.1109/MS.2009.195
Gacitua R, Sawyer P, Gervasi V (2010) On the effectiveness of abstraction identification in requirements engineering. In: 2010 18th IEEE International Requirements Engineering conference (RE), pp 5–14. doi:10.1109/RE.2010.12
Gervasi V (1999) Environment support for requirements writing and analysis. PhD thesis, University of Pisa
Gervasi V, Nuseibeh B (2000) Lightweight validation of natural language requirements: A case study. In: Proceedings Fourth International Conference on Requirements Engineering. ICRE 2000. IEEE Comput. Soc., pp 113–133. doi:10.1.1.28.9876
Goldin L, Berry DM (1997) Abstfinder, a prototype natural language text abstraction finder for use in requirements elicitation. Autom Softw Eng 4:375–412. doi:10.1.1.26.8152
Gorschek T, Garre P, Larsson SBM, Wohlin C (2007) Industry evaluation of the requirements traction model. Requir Eng 12(3):163–190. doi:10.1007/s00766-007-0047-z
Gotel O, Finkelstein C (1994) An analysis of the requirements traceability problem. In: Proceedings of the first international conference on requirements engineering, 1994, pp 94–101. doi:10.1109/ICRE.1994.292398
Hayes J, Dekhtyar A, Osborne J (2003) Improving requirements tracing via information retrieval. In: Proceedings 11th IEEE international requirements engineering conference, 2003, pp 138–147. doi:10.1109/ICRE.2003.1232745
Hayes J, Dekhtyar A, Sundaram S, Holbrook E, Vadlamudi S, April A (2007) Requirements tracing on target (retro): improving software maintenance through traceability recovery. Innovations Syst Softw Eng 3:193–202. doi:10.1007/s11334-007-0024-1
Hayes JH, Dekhtyar A, Sundaram SK (2006) Advancing candidate link generation for requirements tracing: the study of methods. IEEE Trans Softw Eng 32(1):4–19. doi:10.1109/TSE.2006.3

Higgins S, Laat M, Gieles P, Geurts E (2003) Managing requirements for medical it products. IEEE Softw 20(1):26–33
Höst M, Wohlin C, Thelin T (2005) Experimental context classification: Incentives and experience of subjects. In: Proceedings of the 27:th International Conference on Software Engineering (ICSE), pp 470–478
Hulkko H, Abrahamsson P (2005) A multiple case study on the impact of pair programming on product quality. In: ICSE '05: proceedings of the 27:th international conference on software engineering. ACM, New York, pp 495–504. doi:10.1145/1062455.1062545
IBM (2010a) Rational doors (former telelogic doors) product description. http://www-01.ibm.com/software/awdtools/doors/productline/. Accessed 7 Sept 2011
IBM (2010b) Rational doors product description (former telelogic doors). http://www-01.ibm.com/software/awdtools/doors/productline/. Accessed 7 Sept 2011
IEEE (2010) The IEEE keyword taxonomy webpage. http://www.computer.org/mc/keywords/software.htm. Accessed 7 Sept 2011
Ivarsson M, Gorschek T (2009) Technology transfer decision support in requirements engineering research: a systematic review of REj. Requir Eng 14(3):155–175. doi:10.1007/s00766-009-0080-1
Jackson P, Moulinier I (2002) Natural language processing for online applications. Text retrieval, extraction and categorization, natural language processing, vol 5. Benjamins, Amsterdam, Philadelphia
Jarke M (1998) Requirements tracing. Commun ACM 41(12):32–36. doi:10.1145/290133.290145
Kachigan SK (1991) Multivariate statistical analysis: a conceptual introduction. Radius Press
Kamsties E, Berry DM, Paech B (2001) Detecting ambiguities in requirements documents using inspections. In: Proceedings of the first Workshop on Inspection in Software Engineering (WISE 2001), pp 68–80. doi:10.1.1.93.4870
Karlsson L, sa G Dahlstedt, Natt Och Dag J, Regnell B, Persson A (2002) Challenges in market-driven requirements engineering—an industrial interview study. In: Proceedings of the eighth international workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2002)
Kitchenham B, Pfleeger SL, Pickard LM, Jones PW, Hoaglin DC, Emam E, Rosenberg J (2002) Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng 28(8):721–734. doi:10.1109/TSE.2002.1027796
Konrad S, Gall M (2008) Requirements engineering in the development of large-scale systems. In: Proceedings of the 16th international Requirements Engineering conference (RE 2008), pp 217–222
Kotonya G, Sommerville I (1998) Requirements engineering. Wiley
Lauesen S (2002) Software requirements—styles and techniques. Addison–Wesley
Leuser J (2009) Challenges for semi-automatic trace recovery in the automotive domain. In: TEFSE '09: proceedings of the 2009 ICSE workshop on traceability in emerging forms of software engineering. IEEE Computer Society, Washington, pp 31–35. doi:10.1109/TEFSE.2009.5069580
Lin J, Lin CC, Huang J, Settimi R, Amaya J, Bedford G, Berenbach B, Khadra O, Duan C, Zou X (2006) Poirot: A distributed tool supporting enterprise-wide automated traceability. In: 14th IEEE international conference Requirements Engineering, pp 363–364. doi:10.1109/RE.2006.48
Lormans M, Van Deursen A (2006) Can lsi help reconstructing requirements traceability in design and test? In: CSMR '06: proceedings of the conference on software maintenance and reengineering. IEEE Computer Society, Washington, pp 47–56
Lucia AD, Fasano F, Oliveto R, Tortora G (2007) Recovering traceability links in software artifact management systems using information retrieval methods. ACM Trans Softw Eng Methodol 16(4):13. doi:10.1145/1276933.1276934
Lund University (2011a) The requirements engineering course (ets170) at the lund university. http://www.cs.lth.se/ETS170/. Accessed 7 Sept 2011
Lund University (2011b) The software development of large systems course page at lund university. http://cs.lth.se/etsn05/. Accessed 7 Sept 2011
Macias B, Pulman SG (1995) A method for controlling the production of specifications in natural language. Comput J 48(4):310–318
Manning CD, Schütze H (2002) Foundations of statistical natural language processing. MIT Press
Marcus A, Maletic JI (2003) Recovering documentation-to-source-code traceability links using latent semantic indexing. In: ICSE '03: proceedings of the 25th international conference on software engineering. IEEE Computer Society, Washington, pp 125–135
Mich L, Mylopoulos J, Nicola Z (2002) Improving the quality of conceptual models with nlp tools: an experiment. Tech. rep., University of Trento. doi:10.1.1.62.6397

Nakagawa S (2004) A farewell to bonferroni: the problems of low statistical power and publication bias. Behav Ecol 15(6):1044–1045. doi:10.1093/beheco/arh107, http://beheco.oxfordjournals.org/content/15/6/1044.short, http://beheco.oxfordjournals.org/content/15/6/1044.full.pdf+html

Natt och Dag J (2006a) Managing natural language requirements in large-scale software development. PhD thesis, Lund University, Sweden

Natt och Dag J (2006b) The reqsimile tool website. http://reqsimile.sourceforge.net/

Natt och Dag J, Gervasi V, Brinkkemper S, Regnell B (2004) Speeding up requirements management in a product software company: linking customer wishes to product requirements through linguistic engineering. In: Proceedings of the 12th international requirements engineering conference, pp 283–294. (RE 2004)

Natt och Dag J, Thelin T, Regnell B (2006) An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development. Empir Softw Eng 11(2):303–329. doi:10.1007/s10664-006-6405-5

Panis MC (2010) Sucessful deployment of requirements traceability in a commercial engineering organization ... really. In: Proceedings of the 18th IEEE international requirements engineering conference, pp 303–307

Parrish A, Smith R, Hale D, Hale J (2004) A field study of developer pairs: productivity impacts and implications. IEEE Softw 21(5):76–79. doi:10.1109/MS.2004.1331306

Perneger TV (1998) What's wrong with Bonferroni adjustments, vol 316

Pohl K, Bockle G, van der Linden FJ (2005) Software product line engineering: foundations, principles and techniques. Springer

Ramesh B, Jarke M (2001) Toward reference models for requirements traceability. IEEE Trans Softw Eng 27(1):58–93. doi:10.1109/32.895989

Ramesh B, Powers T, Stubbs C, Edwards M (1995) Implementing requirements traceability: a case study. In: RE '95: proceedings of the second IEEE international symposium on requirements engineering. IEEE Computer Society, Washington, p 89

Rayson P, Emmet L, Garside R, Sawyer P (2001) The revere project: Experiments with the application of probabilistic NLP to systems engineering. In: Natural Language Processing and Information Systems, vol 1959. Lecture Notes in Computer Science, Springer, pp 288–300. doi:10.1007/3-540-45399-7_24

Regnell B, Brinkkemper S (2005) Engineering and managing software requirements, Springer, chap market–driven requirements engineering for software products, pp 287–308

Regnell B, Beremark P, Eklundh O (1998) A market-driven requirements engineering process: results from an industrial process improvement programme. Requir Eng 3(2):121–129. doi:10.1007/BF02919972

Rolland C, Proix C (1992) A natural language approach for requirements engineering. In: Advanced information systems engineering, vol 593. Lecture Notes in Computer Science, Springer, pp 257–277. doi:10.1007/BFb0035136

Rupp C (2000) Linguistic methods of requirements engineering (NLP). In: Proceedings of the EuroSPI 2000, pp 68–80

Ryan K (1993) The role of natural language in requirements engineering. In: Proceedings of the IEEE international symposium on requirements engineering. IEEE Computer Society Press, San Diego, California, pp 240–242. doi:10.1.1.45.7180

Samarasinghe R, Nishantha G, Shutto N (2009) Total traceability system: A sustainable approach for food traceability in smes. In: 2009 international conference on Industrial and Information Systems (ICIIS), pp 74–79. doi:10.1109/ICIINFS.2009.5429887

Sawyer P, Cosh K (2004) Supporting measur-driven analysis using nlp tools. In: Proceedings of the 10th international workshop on Requirements Engineering: Foundations of Software Quality (REFSQ 2004), pp 137–142

Sawyer P, Rayson P, Garside R (2002) REVERE: support for requirements synthesis from documents. Inf Syst Front 4(3):343–353. doi:10.1023/A:1019918908208

Sawyer P, Rayson P, Cosh K (2005) Shallow knowledge as an aid to deep understanding in early phase requirements engineering. IEEE Trans Softw Eng 31(11):969–981. doi:10.1109/TSE.2005.129

Shull FJ, Carver JC, Vegas S, Juristo N (2008) The role of replications in empirical software engineering. Empir Softw Eng 13(2):211–218. doi:10.1007/s10664-008-9060-1

Sjøberg DIK, Hannay JE, Hansen O, Karahasanovic VB, Liborg A, Rekdal NK (2005) The survey of controlled experiments in software engineering. IEEE Trans Softw Eng 31(9):733–753. doi:10.1109/TSE.2005.97

Sommerville I, Sommerville I, Sawyer P, Sawyer P (1997) Viewpoints: principles, problems and a practical approach to requirements engineering. Ann Softw Eng 3:101–130

Strens M, Sugden R (1996) Change analysis: a step towards meeting the challenge of changing requirements. In: Proceedings IEEE symposium and workshop on engineering of computer-based systems, 1996, pp 278–283. doi:10.1109/ECBS.1996.494539

Wiegers KE (2003) Software requirements, 2nd edn. Microsoft Press. http://www.worldcat.org/isbn/0735618798

Wilkinson L (1999) Statistical methods in psychology journals: guidelines and explanations. Am Psychol 54(8):594–604

Wilson W, Rosenberg LH, Hyatt L (1997) In: ICSE '97: Proceedings of the 19th international conference on software engineering. ACM, New York, pp 161–171. doi:10.1145/253228.253258

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslen A (2000) Experimentation in software engineering an introduction. Kluwer Academic Publishers

Zowghi D, Offen R (1997) A logical framework for modeling and reasoning about the evolution of requirements. In: Proceedings of the third IEEE international symposium on requirements engineering 1997, pp 247–257. doi:10.1109/ISRE.1997.566875

Krzysztof Wnuk is a doctoral candidate in Software Engineering at Lund University's Department of Computer Science, Sweden. He received his M.Sc. Degree from Gdansk University of Technology, Poland in 2006. His research interests include market-driven software development, requirements engineering, software product management, decision making in requirements engineering, large-scale software, system and requirements engineering and management and empirical research methods.

Martin Höst is a Professor in Software Engineering at Lund University, Sweden. He received an M.Sc. degree from Lund University in 1992 and a Ph.D. degree in Software Engineering from the same university in 1999. His main research interests include software process improvement, software quality, and empirical software engineering. The research is mainly conducted through empirical methods such as case studies, controlled experiments, and surveys. He has published more than 40 papers in international journals and conference proceedings.

Björn Regnell is a professor of Software Engineering in Lund University's Department of Computer Science and Vice Dean of Research at the Faculty of Engineering, LTH. His research interests include market-driven software development, requirements engineering, software quality, software innovation, software product management, and empirical research methods. He received his PhD in software engineering from Lund University.