Empirical evaluation of change impact predictions using a requirements management tool with formal relation types

R.S.A. van Domburg
r.s.a.vandomburg@student.utwente.nl

ABSTRACT

A quasi-experiment was conducted to evaluate the impact of using TRIC, a software tool with support for formal requirements relationship types, on the performance of change impact predictions. It was revealed that empirical experiments cannot provide a solution validation to new software tools because there are not enough experts in the new tool. Using a group of non-experts as participants, the results for a set of change scenarios on a low-complexity software requirements specification indicated that TRIC resulted in slower change impact predictions in three out of five cases without any changes in quality.

Keywords

Change impact prediction, formal requirements relationship types, requirements management, software tool.

1. INTRODUCTION

The QuadREAD Project aims at a better alignment between analysts and architects [20]. It has contributed a requirements metamodel with formal requirements relationship types and a prototype software tool called TRIC that supports it. Earlier case study results concluded that TRIC supports a better understanding of mutual dependencies between requirements, but that this result could not be generalized pending a number of industrial and academic case studies with empirical results [11].

The problem that this research deals with is the lack of solution validation of the requirements metamodel, which can inhibit its adoption because the benefits are not clear.

Using the goal template from the Goal-Question-Metric approach, the research objective is formulated as follows:

To analyze the real-world impact of using a software tool with formal requirements relation- ship types; for the purpose of the evaluation of effectiveness of tools; with respect to the quality of change impact predictions; in the context of software requirements management; from the viewpoint of system maintenance engineers.

This research is conducted at the laboratory of the Software Engineering Group from March 2009 up to and including November 2009. It takes place within the context of the QuadREAD Project, which is a joint research project of the Software Engineering and Information Systems research groups at the Department of Computer Science in the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente.

2. BACKGROUND AND RELATED WORK

Requirements evolution both during the requirement engineering process and after a system has gone into service is inevitable [24]. System maintenance engineers analyze requested changes as part of the requirements management cycle [17]. Using the requirements to understand the system and the relationship between its parts, they predict the impact that a requested change in a particular requirement will have on other requirements [24]. Increased understanding about a software requirements specification helps them to perform this activity effectively [11].

Requested changes can take the form of change scenarios, which describe possible change situations that will cause the maintenance organization to perform changes in the software [6]. Several scenario-based methods have been proposed to evaluate software architectures with respect do desired quality attributes such as maintainability, performance, and so on [5]. There has been little focus on change scenarios themselves, which poses a weakness in methodologies that depend on them [6].

Change impact predictions enumerate the set of objects estimated to be affected by the change impact analysis method. Change impact analysis is the identification of potential consequences of a change, or estimating what needs to be modified to accomplish a change [4]. See Table 1.

Table 1. Change impact prediction sets [2]

Set	Abbr.	Description
System	-	Set of all objects under consideration.
Estimated Impact Set	EIS	Set of objects that are estimated to be affected by the change.
Actual Impact Set	AIS	Set of objects that were actually modified as the result of performing the change.
False Positive Impact Set	FPIS	Set of objects that were estimated by the change impact analysis to be affected, but were not affected during performing the change.
Discovered Impact Set	DIS	Set of objects that were not estimated by the change impact analysis to be affected, but were affected during performing the change.

Table 1 shows that the Estimated Impact Set may not be equal to the Actual Impact Set. Thus, there is a quality attribute to change impact predictions. This may be captured using a binary classifier; see the confusion matrix in Table 2.

Table 2. Confusion matrix [10]

		Actual Impact	
		Changed	**Not changed**
Estimated Impact	**Changed**	True Positive	False Positive
	Not changed	False Negative	True Negative

Binary classifiers are also used in the domain of information

retrieval. Metrics from this domain may be used to measure the quality of change impact predictions [2]. See Table 3.

Table 3. Change impact prediction quality metrics [2]

Metric	Equation	Also known as
Recall	$\frac{\lvert EIS \cap AIS \rvert}{\lvert AIS \rvert}$	Hit rate, sensitivity, true positive rate
Precision	$\frac{\lvert EIS \cap AIS \rvert}{\lvert EIS \rvert}$	Positive predictive value
Fallout	$\frac{\lvert FPIS \rvert}{\lvert System \rvert - \lvert AIS \rvert}$	False alarm rate, false positive rate

A popular measure that combines precision and recall is the weighted harmonic mean of precision and recall, also known as the F_1-measure because recall and precision are evenly weighted [2]. See Equation 1.

Equation 1. F_1 measure

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

The F_1 measure is referred to as the *F*-measure. Results on the *F*-measure are referred to as *F*-scores.

Software maintenance engineers may produce better quality change impact predictions through increased understanding about the requirements and their relations. One requirement in a software requirement may be related to one or more other requirements in that specification. Relationships can be of a certain type that more precisely defines how requirements are related. Using imprecise relationship types may produce deficient results in requirements engineering [11].

The formal relationship types defined by the QuadREAD Project are based on first-order logic and can be used for consistency checking of relationships and inferencing. Consistency checking is the activity to identify the relationships whose existence causes a contradiction. Inferencing is the activity of deriving new relationships based solely on the relationships that a requirements engineer has already specified. The available relationship types are requires, refines, partially refines, contains and conflicts [11].

TRIC is a prototype software tool for requirements management which supports the creation, updating, viewing and deletion of relations with the formal relationship types between requirements, effectively adding traceability support [11]. Other software tools that can be used for requirements management include but are not limited to Microsoft Excel, a general-purpose spreadsheet application, and IBM Rational RequisitePro, a dedicated requirements management tool that is well-known in industry [12]. RequisitePro only defines *traceTo* and *traceFrom* relationship types, which are very generic. It only supports inferencing based on transitivity [11].

Related experiments: An experimental study was conducted in which participants perform impact analysis on alternate forms of design record information. Here, a design record is defined as a collection of information with the purpose to support activities following the development phase, which would include traceability artifacts. These results suggest that design records have the potential to be effective for software maintenance but training and process discipline is needed to make design recording worthwhile. The experiment observed a lack of focus with the participants and unreliable tutoring of participants. The results of an analysis of variance on the quality attributes of change impact prediction were non-significant [1].

A requirements management tool, the Storymanager, was developed to manage rapidly changing requirements for an eXtreme Programming team. As part of action research, the tool was used in a case project where a mobile application for real markets was produced. The tool was dropped by the team after only two releases. The principle results show that the tool was found to be too difficult to use and that it failed to provide as powerful a visual view as paper-pen board method [14].

A trace approach was introduced that focuses on impact analysis of system requirements changes and that is suited for embedded control systems. With significant statistical significance, an empirical study showed that the trace approach allows a more effective impact analysis of changed on embedded systems than non-trace approaches; the additional information helped in getting a more complete and correct set of predicted change impacts [25]. It is plausible that this is caused by the higher complexity of the research object that was used in that experiment.

A prototype software tool called traceMAINTAINER was developed to automate traceability maintenance tasks in evolving UML models. The research yielded two conclusions with limited generalizability. First, the group using traceMAINTAINER required significantly fewer manual changes to perform their change management. Second, there was no significant difference between the quality of the change predictions of the two groups [18].

TRIC was illustrated using a single fictional case study featuring a course management system. Based on the case study results, it was concluded that TRIC supports a better understanding of mutual dependencies between requirements, but that this result could not be generalized pending a number of industrial and academic case studies with empirical results [11].

3. EXPERIMENTAL DESIGN

3.1 Goal

The goal of this experiment is to analyze the real-world impact of using a software tool with formal requirements relationship types for the purpose of the evaluation of the effectiveness of tools with respect to the quality of change impact predictions.

3.2 Hypothesis

It is hypothesized that using TRIC, a software tool with formal requirements relationship types, will positively impact the quality of change impact predictions.

Hypothesis 1. The *F*-scores of change impact predictions of system maintenance engineers using TRIC will be equal to or less than those from system maintenance engineers not using TRIC.

Hypothesis 2. The time taken to complete change impact predictions of system maintenance engineers using TRIC will be equal to or greater than those from system maintenance engineers not using TRIC.

The statistical significance is 5% (α=0,05).

3.3 Design

Different groups will be assigned to perform change impact analysis using a different software tool. This research setup involves control over behavioral events during change impact analysis with administrator selection, for which experimental research is the most appropriate [26] and quasi-experimental in particular [21].

This research follows a synthetic design with three treatments. This allows controlling the level of tool support while still being feasible for execution within a limited amount of time. The treatment is the administration of Excel, RequisitePro and Excel. The observation is the change impact prediction quality as measured by *F*-score and time taken to complete the prediction.

3.4 Parameters

A single real-world software requirements specification will be selected as research object. Predetermined groups of participants will perform change impact prediction on the requirements that are present in this specification.

3.5 Variables

The dependent variables that are measured in the experiment are those that are required to compute the *F*-score:

- Size of the Estimated Impact Set
- Size of the False Positive Impact Set
- Size of the Discovered Impact Set

The precision, recall and finally *F*-scores can then be computed.

One independent variable in the experiment is the supplied software tool during change impact analysis. This is measured on a nominal scale: Microsoft Excel, IBM Rational RequisitePro or TRIC.

It would be a threat to internal validity to only study the impact of using Microsoft Excel and TRIC, because such an experimental design would be biased in favor of TRIC. When assuming that requirements relationships play an important role in the results of change impact prediction, it would be logical that a software tool with dedicated support would score higher than a software tool without such support. By also studying an industrially accepted tool such as IBM Rational RequisitePro, concerns to validity regarding the bias in tool support are addressed.

The following covariate variables were expected to influence the *F*-scores of change impact predictions and time taken to complete them [16, 23]:

- Level of formal education
- Nationality
- Gender
- Current educational program
- Completion of a basic requirements engineering course
- Completion of an advanced requirements engineering course
- Previous requirements management experience

3.6 Planning

The experiment takes place from 13:45 to 17:30 on June 11, 2009. The participants register pre-experiment and provide responses to the covariables. Groups are created by first assigning the participants at random. Groups are equalized on covariates by manually moving participants from group to group.

During the experiment, the participants receive an equal and general instruction about change management for 15 minutes. They then receive an instruction specific to their tool for 30 minutes. Following that, they receive an equal kick-off instruction with the experimental procedure and prizes to be won for 5 minutes. Participants are then granted 60 minutes to review the software requirements specification in any way they see fit. Following a 15-minute break, they are granted 60 minutes to perform change impact prediction for five change scenarios. Change scenarios are distributed to the participants in random order to compensate for learning effects [15].

The instructions are provided by the team of researchers.

3.7 Participants

Participants will be master students following the Software Management master course at the University of Twente. The experiment is not strictly part of the course and students are encouraged to participate on a voluntary basis. For each software tool group, there is a first prize of € 50 and a second prize of € 30. Everyone is presented with a USB memory stick.

3.8 Objects

The research object is a software requirements specification titled "Requirements for the WASP Application Platform" version 1.0 by the Telematica Instituut [9]. This is a public, real-world requirements specification in the context of context-aware mobile telecommunication services, with three scenarios, 16 use cases and 71 requirements. The page count including prefaces is 62. The WASP requirements specification features inter-level tracing from scenarios to use cases and from use cases to scenarios. The requirements are functionally decomposed and ordered in hierarchies. For each function, there is a tree with a calculated tree impurity of 0.

Scenarios were created to cover a range of change scenario cases. Five separate cases can be discerned in the theory on formal requirements relationships [11]. See Table 3.

Table 3. Change scenarios cases and tasks

Case	Task
Add part	1
Remove part	2, 4
Add detail to part	3
Add whole	-
Remove whole	5

No change scenario was created for the "add whole" case because that does not impact other requirements. A replacement scenario was created for "remove part". This was convenient because many requirements in the WASP specification have multiple parts.

3.9 Instrumentation

All participants are handed out a printout of all slides that were shown to them, a copy of the software requirements specification and a USB memory stick. The memory stick contains the requirements specification in PDF format and a digital

requirements document that can be opened with their software tool. It is pre-filled with all requirements but contains no relations. The participants are told to treat the introduction, scenario and requirements chapters as leading and the use case chapter as informative.

3.10 Data collection

A web application is created to support the registration of participants, distribution of experiment tasks and collection of data. The Actual Impact Set is to be determined as a golden standard from experts.

3.11 Analysis procedure

The web application has built-in support to calculate the *F*-scores. SPSS will be used to perform an analysis of variance using planned comparisons to test if participants in the TRIC group had significantly different *F*-scores and times than those in the Microsoft Excel or IBM Rational RequisitePro groups. A similar test will be performed for analysis of covariance. Finally, a multiple analysis of variance will be used to test if there are interaction effects between the *F*-scores and times.

3.12 Validity evaluation

This research features a limited sample set and statistical power will be low as a result. The observed power and required sample size for proper power will be calculated as part of the analysis.

The used tools are not fully comparable in terms of functionality, maturity and usability. Any inferences will only be valid as they pertain to these tools specifically, not to similar applications.

The setup of the instruction is not any fairer by assigning equal slots of time. While an equal amount of time is given to all groups for the lecture, the complexity of the tools is very much different. By compressing more required knowledge into a shorter timeframe, the intensity of the lecture decreases and participants cannot be expected to understand the software tools equally well. Using a pre-test and post-test to compensate for learning effects would allow accurately measuring the influence of the instruction on the results [15], although ways to reliably measure aptitude are not directly available and would be a study in itself.

A lack of theory about what a proper change scenario should be has caused the change scenarios to be developed in a rather ad-hoc fashion.

The number of constructs and methods that are used to measure the quality of change impact prediction is monogamous; only the *F*-score is truly a measure of "product" quality, with the time taken being more of a measure of "process" quality. This may underrepresent the construct of interest, complicate inferences and mix measurements of the construct with measurement of the method [21].

This experiment is subject to Hawthorne effects [21] because of participants reacting differently in experimental conditions.

Inferences will only be valid as they pertain to the WASP requirements specification and the specific participants.

Participants may not represent real-world system maintenance engineers. Finally, the instructors are three different people that may not have equal instructing aptitude.

4. EXECUTION

The experiment was conducted with 22 participants. 21 of these participants completed the online registration before the start of the experiment to score the covariates and facilitate group matching. 2 participants did not pre-register. Their responses to the registration were added after the execution of the experiment. All participants who registered also showed up.

The participants were distributed over three groups. 6 participants were in the Microsoft Excel group, 7 in the IBM Rational RequisitePro group and 8 in the TRIC group.

5. PREPARATION

Three locations were booked with the facility management of the University of Twente; one location per group. Two of the three assigned locations were computer clusters in a single large room with a total of four clusters. The third location was a computer cluster in a room on the first floor. The rooms were not comparable in terms of environment or layout. No three neutral rooms were available.

Five slideshows were created: one for the general instruction, three for the specific instruction (one per group) and one for the general kick-off.

Data collection performed: All 22 participants submitted estimated impact sets for six change scenarios. Consequently 132 estimated impact sets were collected. Of these, 22 were the result of warm-up scenarios and were not used in statistical analysis.

Validity procedure: There was construction work ongoing in two of the experimental locations. The rooms were also occupied by other people who were working aloud. This caused the rooms to be noisy.

Not all students were as focused on the task as expected, in spite of the monetary rewards offered. One student was actively listening to music and seen watching YouTube videos during the experiment. Nothing was about this.

Many students were not finished with adding relationships before the break. After the break, some of them tried catching up by adding more relationships. Others started change impact prediction with the unfinished set of relationships. When this was noticed, the supervisors jointly decided to provide an extra 15 minutes. The extra time was not enough for many students.

Not all students used the tool to full effect and some did not use them at all. Nothing was about this, because the participants were told to use the software tool and documents in any way they saw fit.

Some participants did not check the initially changed requirement as part of their Estimated Impact Set, even though they were instructed to do so both during the lecture and by the web application. The data set was corrected to include the initially changed requirement for all participants. The underlying assumption is that this has been an oversight by the participants.

6. ANALYSIS

6.1 Change scenario representativeness

One of the original authors of the WASP specification was asked to rate the representativeness of the change scenarios on an ordinal scale of low, medium or high. See Table 5.

Table 5. Representativeness of change scenarios

Scenario	Represenativeness
1	Medium

2	Low
3	High
4	Medium
5	Low

6.2 Golden standard reliability

The establishment of a golden standard was initiated after the experiment was conducted. Four people created a golden standard individually; one expert (another original author from the WASP specification still with Novay) and three academics with the software engineering department and the QuadREAD Project: a postdoc, a PhD candidate and a master student.

The inter-rater reliability was calculated as a measure of the level of agreement between the golden standards [13]. The type of inter-rater reliability is case 2 (the same raters rate each case) [22] using an absolute agreement definition (the ordering of ratings matters) [3]. See Table 6.

Table 6. Inter-rater reliability analysis

Task	Impacted set size	Mean	Standard error	Intraclass correlation
1	3	58,1%	9,1%	0,832
2	9	78,6%	4,2%	0,936
3	1	100,0%	0,0%	1,000
4	1	100,0%	0,0%	1,000
5	6	44,9%	9,7%	0,712

6.3 One-way between-groups ANOVA

An analysis of variance on each task is conducted separately for the *F*-score and time taken.

Testing for assumptions to perform an analysis of variance, a Kolmogorov-Smirnov test for normality revealed non-normality for several results of tasks 2, 4 and 5. It was decided to analyze these tasks using a non-parametric test.

Table 7. One-way between-groups ANOVA on *F*-score

	F-score (higher is better)			ANOVA
Task	Group	Mean	Standard deviation	Significance
1	Excel	0,498	0,232	0,866
	RequisitePro	0,658	0,187	
	TRIC	0,593	0,176	
	Total	0,588	0,198	
3	Excel	0,407	0,321	0,629
	RequisitePro	0,468	0,290	
	TRIC	0,507	0,325	
	Total	0,465	0,300	

Table 7 presents the results of a one-way between-groups analysis of variance to explore the impact of using three different software tools on the quality of change impact predictions, as measured by the *F*-score. Using a planned comparison for the TRIC group, there were no statistically significant differences at the $p<0,05$ level in the *F*-scores of the three groups in either task 1 [$F(1, 18)=0,030$; $p=0,866$] or task 3 [$F(1, 18)=0,242$; $p=0,629$].

Table 8. One-way between-groups ANOVA on time taken

	Time (lower is better)			ANOVA
Task	Group	Mean	Standard deviation	Significance
1	Excel	193	89	0,000
	RequisitePro	137	53	
	TRIC	368	117	
	Total	241	136	
3	Excel	172	70	0,219
	RequisitePro	239	121	
	TRIC	314	219	
	Total	249	161	

Table 8 presents the results of a one-way between-groups analysis of variance to explore the impact of using three different software tools on the time taken to complete predicting change impact, as measured in seconds. There was a statistically significant difference at the $p<0,05$ level in the times of the three groups for task 1 [$F(1, 18)=24,04$; $p=0,000$]. The effect size, calculated using $\eta 2$, was 0,572. In Cohen's terms, the difference in mean scores between the groups is large [7]. The TRIC group performs change impact predictions 48% slower than the Microsoft Excel group and 63% slower than the IBM Rational RequisitePro group.

There was no statistically significant difference at the $p<0,05$ level in the times of the three groups for task 3 [$F(1, 18)=1,753$; $p=0,219$].

The attained statistical power is 56% for detecting effects with a large size, $p<0,05$; sample size 21 and 18 degrees of freedom.

6.4 Non-parametric testing

Table 9 and Table 10 display the results of $\chi 2$ test for tasks 2, 4 and 5, which did not meet the requirements for analyzing them using a more sensitive analysis of variance.

Table 9. $\chi 2$ test for goodness of fit on *F*-score

	F-score (higher is better)			$\chi 2$
Task	Group	Mean	Standard deviation	Significance
2	Excel	0,499	0,319	0,584
	RequisitePro	0,517	0,129	
	TRIC	0,424	0,275	
	Total	0,476	0,242	
4	Excel	0,407	0,182	0,717
	RequisitePro	0,524	0,230	
	TRIC	0,461	0,161	
	Total	0,467	0,188	
5	Excel	0,423	0,160	0,444
	RequisitePro	0,528	0,100	
	TRIC	0,573	0,151	

	Total	0,515	0,146	

Table 9 presents the results of a $\chi 2$ test to explore the impact of using three different software tools on the quality of change impact predictions, as measured by the *F*-score. There were no statistically significant differences at the $p<0,05$ level in the F-scores of the three groups in task 2 [$\chi 2=1,077$; df=2; p=0,584], task 4 [$\chi 2=0,667$; df=2; p=0,717] or task 5 [$\chi 2=1,625$; df=2; p=0,444].

Table 10. $\chi 2$ test for goodness of fit on time

	Time (lower is better)			$\chi 2$
Task	Group	Mean	Standard deviation	Significance
2	Excel	133	83	0,000
	RequisitePro	154	76	
	TRIC	222	137	
	Total	174	107	
4	Excel	213	111	0,000
	RequisitePro	300	81	
	TRIC	467	248	
	Total	339	196	
5	Excel	324	274	0,000
	RequisitePro	170	64	
	TRIC	342	133	
	Total	280	181	

Table 21 presents the results of a $\chi 2$ test to explore the impact of using three different software tools on the time taken to complete change impact predictions, as measured in seconds. There were statistically significant differences at the $p<0,05$ level in the times between the three groups in task 2 [$\chi 2=414$; df=2; p=0,000], task 4 [$\chi 2=102$; df=2; p=0,000] or task 5 [$\chi 2=612$; df=2; p=0,000].

A post-hoc comparison using a Mann-Whitney U test revealed that the time taken to complete task 4 was significantly different between the Microsoft Excel and TRIC groups, p=0,020. The TRIC group performs change impact predictions 54% slower than the Microsoft Excel group.

A similar post-hoc comparison revealed that the time taken to complete task 5 were significantly different between the IBM Rational RequisitePro and TRIC groups, p=0,011. The TRIC group performs change impact predictions 50% slower than the IBM Rational RequisitePro group.

No other combination of groups yielded a significant difference in times results in the post-hoc test, including task 2.

The attained statistical power for the $\chi 2$ tests is 52% for detecting effects with a large size, $p<0,05$, sample size 21 and two degrees of freedom.

6.5 Analysis of covariance

The reliability of the covariates, as measured by Cronbach's alpha, is only 0,310 which indicates poor reliability. Attempts to eliminate one or more weak covariables resulted in a Cronbach's alpha of 0,585, which is too low to warrant an analysis of covariance [19] and was therefore not executed.

6.6 Multivariate analysis of variance

An assessment of the linearity of *F*-scores and times using a Pearson product-moment correlation calculation revealed no linearity. Transformation strategies in an attempt to attain linearity over a skewed data set did not yield linearity. A multivariate analysis of variance was therefore not warranted [19] or executed.

7. INTERPRETATION

7.1 Change scenario representativeness

Not all change scenarios were deemed to be representative, which is a reliability issue.

7.2 Golden standard reliability

Statistical testing for tasks 1 up to and including 4 did not reveal any significant differences between the golden standards and suggested excellent inter-rater reliability.

Statistical testing for task 5 indicates a statistically significant difference between the golden standards. However, the more precise intraclass correlation score does suggest good inter-rater reliability.

The high inter-rater reliability means that the design of the tasks is feasible. Had they been too ambiguous, then it would have been likely that the inter-rater reliability would have been much lower.

7.3 One-way between-groups ANOVA

The quality of change impact predictions is not impacted by the software tool that is being used for tasks 1 or 3. A similar conclusion can be drawn about the time taken to complete task 3.

The time taken to complete task 1 is significantly different for the group that used TRIC. They performed change impact prediction of scenario 1 slower than the other groups.

7.4 Non-parametric testing

The quality of change impact predictions is not impacted by the software tool that is being used for tasks 2, 4 or 5.

The time taken to complete tasks 4 and 5, who respectively remove a part and remove a whole, are significantly different for the group that used TRIC. For task 4, the TRIC group was slower than the Microsoft Excel group. For task 5, the TRIC group was slower than the IBM Rational RequisitePro group.

The time taken to complete task 2 was indicated to be significantly different for the group that used TRIC by the $\chi 2$ test, but an ensuing post-hoc comparison using a Mann-Whitney U test indicated that this result is a false positive, likely caused by a small sample size [8].

7.5 Analysis of covariance

The reliability of the covariates was too low to conduct an analysis of variance. Of the strongest covariates, the first three somehow measure the same construct. The completion of a basic requirements engineering course, completion of an advanced requirements engineering course, and months of experience, are in fact all a measure of experience with requirements management. Statistical testing detects correlations amongst these variables of medium effect size.

7.6 Multivariate analysis of variance

The assumption of linearity between the *F*-score of change impact predictions and time taken to complete them was violated, because of which a multivariate analysis of variance could not be executed. One hypothesis to explain the longer time taken yet equal *F*-score of the TRIC group is that TRIC is a more complex tool. It offers more visualization opportunities and is not as mature as the other software tools. If the benefits of TRIC are to better cope with complexity, then those may only be reaped with an appropriately complex software requirements specification.

8. CONCLUSIONS AND FUTURE WORK

8.1 Summary

The background for this research was to evaluate the impact of TRIC, a software tool that supports the formal requirements relationship types that were developed in the QuadREAD Project, on the quality of change impact predictions. It was hypothesized that using TRIC would positively impact that quality. A quasi-experiment was systematically designed and executed to empirically validate this impact.

The research design revealed that there were not enough TRIC experts in existence to participate in the experiment. This meant that non-expert participants had to be trained to play the role of expert. This posed two important threats to validity. First, this threatens internal validity because the lecture effect is difficult to control. Second, it threatens external validity because the non-experts may not be representative for experts or even system maintenance engineers in general. This is an inherent problem when attempting to empirically provide a solution validation to new software tools.

The object used in the experiment was the WASP specification, a software requirements specification which was found to be clear and of low complexity. Recognizing the benefit of TRIC to deal with complex specifications yet being unable to acquire one of ample complexity meant that the WASP specification was likely to cause non-significant results.

A group of experts created a golden standard to compare participants' change impact predictions against. The inter-rater reliability of these golden standards was high, indicating that the experimental instrumentation is reliable in spite of reliability issues concerning the change scenarios.

8.2 Results

The following conclusions can be drawn with respect to the combination of participants, change scenarios and software requirements specification that were used in this experiment:

- Null hypothesis 1 stated that the *F*-scores of change impact predictions of system maintenance engineers using TRIC will be equal to or less than those from system maintenance engineers. This null hypothesis was accepted.
- Null hypothesis 2 stated that the time taken to complete change impact predictions of system maintenance engineers using TRIC will be equal to or longer than those from system maintenance engineers not using TRIC. This null hypothesis was also accepted.

No differences in the quality of change impact predictions between using Microsoft Excel, IBM Rational RequisitePro or TRIC were detected.

8.3 Limitations

The results of this research are subject to the following limitations:

- Lack over lecture control effect. Participants require training to work with the software tools and play the role of expert. This is difficult to do reliably.
- Low participant representativeness. There is no strong evidence to assume that master students are representative for actual system maintenance engineers.
- Lack of control over change scenarios. It is likely that change scenarios have influence over the results of change impact predictions, but the lack of theory surrounding change scenarios is a cause of reliability problems.
- Small sample size. The sample size of the research is too small to attain the generally accepted statistical power of 80%.
- Limited comparability of software tools. No statistical adjustments have been made for the functionality, maturity and usability of either Microsoft Excel, IBM Rational RequisitePro or TRIC.
- Monogamous metrics. Having more measures of quality would improve the reliability of the results.
- Low participant reliability. Not all participants were as focused on the task as expected. This may have led to suboptimal change impact predictions.
- Limited research object representativeness. An intelligent tool such as TRIC is likely to only show its benefits when tasked with a complex software requirements specification, which was not used here.
- Limited control over environment. The experiment locations were not comparable in terms of layout or noise.

8.4 Future work

The following can be recommended to further pursue the solution validation of the requirements metamodel and TRIC:

- Further the state-of-the-art in change scenario theory, so that it is clear how a certain change scenario can impact change impact prediction.
- Create multiple change scenarios of the same class to test the effect of change scenario classes on change impact predictions.
- Find a number of real-world software requirements specifications of high complexity to test if TRIC's intelligence will then reap benefits.
- Consider organizing an online experiment, where experts can participate from behind their own computer. This allows more time for experimentation and lowers the barrier to entry.
- Consider organizing multiple action research projects, where researchers can apply the techniques in practical cases that are currently running with clients.

9. ACKNOWLEDGEMENTS

Thanks go out to the participants for their participation in the experiment and Johan Koolwaaij and Martin Wibbels for their expert insight into the WASP requirements specification.

10. REFERENCES

1. Abbattista, F., et al. *An Experiment on the Effect of Design Recording on Impact Analysis*. in *International conference on Software Maintenance*. 1994. Victoria, BC: IEEE Computer Society Press. pp. 253-259.
2. Abma, B.J.M., *Evaluation of requirements management tools with support for traceability-based change impact analysis*. 2009, Master's thesis, University of Twente: Enschede.
3. Alexander, I. & Robertson, S., *Understanding project sociology by modeling stakeholders.* IEEE Software, 2004. **21**(1): pp. 23-27.
4. Arnold, R.S. & Bohner, S.A. *Impact Analysis - Towards A Framework for Comparison*. in *Conference on Software Maintenance*. 1993. Montreal, Quebec: IEEE Computer Society. pp. 292-301.
5. Babar, M.A. & Gorton, I. *Comparison of Scenario-Based Software Architecture Evaluation Methods*. in *11th Asia-Pacific Software Engineering Conference*. 2004. Busan, Korea: IEEE Computer Society. pp. 600-607.
6. Bengtsson, P. & Bosch, J. *Architecture Level Prediction of Software Maintenance*. in *Third European Conference on Software Maintenance and Reengineering*. 1999. Amsterdam, The Netherlands: IEEE Computer Society. pp. 139-147.
7. Cohen, J., *Statistical power analysis for the behavioral sciences*. 1988, Hillsdale, New Jersey: Erlbaum.
8. Dawson, B. & Trapp, R.G., *Basic & Clinical Biostatistics*. 4th ed. 2004: McGraw-Hill.
9. Ebben, P., et al., *Requirements for the WASP Application Platform*, in *WASP/D2.1*. 2002, Telematica Instituut: Enschede, The Netherlands.
10. Fawcett, T., *ROC Graphs: Notes and Practical Considerations for Researchers*. 2004, Technical report, HP Laboratories: Palo Alto, California.
11. Goknil, A., et al., *Semantics of Trace Relations in Requirements Models for Consistency Checking and Inferencing*. 2009, to be published in Software and Systems Modeling, University of Twente: Enschede.
12. IBM Corporation. *Rational RequisitePro*. 2009 [cited 2009 October 4]; Available from: http://www-01.ibm.com/software/awdtools/reqpro/.
13. Jedlitschka, A. & Pfahl, D. *Reporting Guidelines for Controlled Experiments in Software Engineering*. in *International Symposium on Empirical Software Engineering*. 2005. Noosa Heads, Australia: IEEE Computer Society Press. pp. 95-104.
14. Kaariainen, J., et al., *Improving Requirements Management in Extreme Programming with Tool Support – an Improvement Attempt that Failed*, in *30th EUROMICRO Conference*. 2004, IEEE Computer Society Press: Rennes, France. pp. 342-351.
15. Kampenes, V.B., et al., *A systematic review of quasi-experiments in software engineering.* Information and Software Technology, 2009. **51**: pp. 71-82.
16. Katz, S., et al., *Gender and Race in Predicting Achievement in Computer Science*, in *IEEE Technology and Society Magazine*. 2003.
17. Lauesen, S., *Software Requirements: Styles and Techniques*. 2002, Harlow: Pearson Education Limited.
18. Mäder, P., Gotel, O. & Philippow, I. *Enabling Automated Traceability Maintenance through the Upkeep of Traceability Relations*. in *5th European Conference on Model Driven Architecture*. 2009. Enschede, The Netherlands: Springer. pp. 174-189.
19. Pallant, J., *SPSS Survival Manual: A Step By Step Guide to Data Analysis Using SPSS for Windows*. 2001, Buckingham, UK: Open University Press.
20. QuadREAD Project. *QuadREAD Project - Project Description*. 2009 [cited 2009 March 14]; Available from: http://quadread.ewi.utwente.nl/index.php?option=com_content&task=view&id=13&Itemid=29.
21. Shadish, W.R., Cook, T.D. & Campbell, D.T., *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. 2002, Boston, New York: Houghton Mifflin Company.
22. Shrout, P.E. & Fleiss, J.L., *Intraclass Correlations: Uses in Assessing Rater Reliability.* Psychological Bulletin, 1979(2): pp. 420-428.
23. Sjøberg, D.I.K., et al. *Conducting Realistic Experiments in Software Engineering*. in *International Symposium on Empirical Software Engineering*. 2002. Nara, Japan: IEEE Computer Society Press. pp. 17-26.
24. Sommerville, I., *Software Engineering*. 7th ed. International Computer Science Series. 2004, Essex, England: Pearson Education.
25. von Knethen, A. *A Trace Model for System Requirements Changes on Embedded Systems*. in *International Conference on Software Engineering*. 2001. Vienna, Austria: ACM. pp. 17-26.
26. Yin, R.K., *Case Study Research: Design and Methods*. 4th ed. Applied Social Research Methods, ed. L. Bickman and D.J. Rog. 2009, Thousand Oaks, California: SAGE Publications.