Practical Relevance of Experiments in Comprehensibility of Requirements Specifications

Nelly Condori-Fernández, Maya Daneva, Klaas Sikkel
University of Twente
Drienerlolaan 5, 7522 NB Enschede, The Netherlands
{n.condorifernandez, m.daneva, ksikkel}@utwente.nl

Andrea Herrmann
Technical University Braunschweig
Mühlenpfordtstr. 23
Braunschweig, Germany
AndreaHerrmann3@gmx.de

***Abstract*—Recently, the Requirements Engineering (RE) community has become increasingly aware of the importance of carrying out industry-relevant research. Researchers and practitioners should be able to evaluate the relevance of their empirical research to increase the likely adoption of RE methods in software industry. It is in this perspective that we evaluate 24 experimental studies on comprehensibility of software requirements specifications to determine their practical value. To that end a checklist based on Kitchenham's approach was operationalized from a practitioner's perspective and an analysis with respect to the main factors that affecting on comprehensibility was carried out. Although 100% of the papers reviewed reported statistically significant results, and 96% of them take examples from a real-life project. 80% of the papers do not scale to real life, 54% of the papers do not specify the context in which the results are expected to be useful. We also found that there is a lack of underlying theory in the formulation of comprehensibility questions.**

Keywords—practioner's checklist; comprehensibility; requirements specifications; experiments

I. INTRODUCTION

The Evidence Based Software Engineering (EBSE) paradigm aims to improve decision-making related to software development and maintenance by integrating current best evidence from research with practical experience and human values [8]. Evidence is defined as a synthesis of the best quality of empirical studies on a specific topic or research question. However, an important problem acknowledged by the supporters of EBSE is that there is not yet any consensus on the definition of "quality".

In the Cochrane handbook [12], the best known reference in EBSE, quality is defined as the extent to which the study minimizes bias and maximizes internal and external validity. However, we consider that construct validity is also a particularly important quality issue to be considered, since higher quality studies are only possible if we use constructs that we understand well and are able to communicate precisely.

Dybå et al. [9] identified three key aspects to assess the quality of primary studies in SE: (i) rigorousness (Has a thorough and appropriate approach been applied to key research methods in the study?), (ii) credibility (Are the findings well-presented and meaningful?), and (iii) relevance (How useful are the findings to the software industry and the research community?).

In our study on comprehensibility of software requirement specifications (SRS), we focus on one of the criteria identified by Dybå et al., namely the relevance. We analyze 24 studies that report on experimental results on comprehensibility of SRS. The 24 sources were identified in a previous systematic mapping study [2], in which comprehensibility was identified as the quality attribute that is studied most by RE researchers.

A secondary purpose of the paper, in addition to the result of the study on comprehensibility, is to contribute to the EBSE methodology. We used a practitioner's perspective-based checklist [17], which was inspired by checklist-based review from multiple perspectives proposed by Kitchenham et al. [5].

Comprehensibility, also called understandability, can be defined as the degree to which information contained in a SRS can be easily understood by a reader of that SRS [22]. It is an essential quality attribute of SRS. For example Krogstie [18] considers comprehension as the only goal of pragmatic quality [1] and the comprehensibility (understandability) as an important factor in achieving the empirical quality.

Comprehensibility is the quality attribute that has most frequently been empirically studied, yet Shima et al. [19] warn that it is difficult to evaluate it in practice. Comprehension is an internal human process, which requires a research that is more multidisciplinary in nature (e.g. by using cognitive theories). Motivated by Shima's affirmation, we asked ourselves how the empirical evaluations on comprehensibility have been carried out so far. If we evaluate how relevant the empirical results on comprehensibility, published until 2008 are for the software industry, then this contribution will allow us to know what kinds of empirical studies could be useful to the software industry and the research community.

The remainder of this paper is organized as follows: Section II. discusses an analysis of the 24 studies with respect to the main factors that affecting on comprehensibility. Section III summarizes the practitioner's checklist used and the reliability analysis that we carried out. Section IV. reports on our results for each of the checklist questions analyzed. Finally conclusions and further work are presented.

[1] It relates to the effect that the model has on the participants and the world.

II. EXPERIMENTS REPORTED ON COMPREHENSIBILITY OF REQUIREMENTS SPECIFICATIONS

An important feature of a good RE process is its ability to facilitate effective communication of requirements among different stakeholders, who, more often than not, have only limited background in computing [1]. Thus, requirements notations and processes must help RE professionals to maintain a delicate balance between the suitability of requirements descriptions for a non-computing audience and the level of precision needed for the downstream developers.

Numerous SRS techniques have been proposed with the purpose of improving the quality of requirements documents and consequently increasing user satisfaction with the final product [2],[16].

This paper focuses on the evaluation of 24 articles reporting experiments on comprehensibility of SRS, which were selected of out 46 articles identified by means of a Mapping Study realized by [2]. The comprehensibility according to this mapping study was one of the aspects most studied in the RE literature from 1976–2008. The references of these studies are in the Appendix of this paper.

A description of these 24 studies in terms of the main factors that affecting on comprehensibility of requirements specifications is presented in Table I. These factors are: type of task, language expertise, domain expertise, and problem size [15].

Type of task: The different tasks that readers perform with a representation are facilitated or hindered to varied degrees. For example, comprehensibility for information search or for information retention can lead to different evaluation results.
Language expertise: The user's previous expertise with the modeling language under study.
Domain expertise: The user's previous expertise with the domain being modelled.
Problem size: Size of the specification. Different modeling languages scale up with variable degrees of success.

We list also the ***objects of study,*** the object being analyzed in each of the 24 empirical studies. This can be a language, method, technique (guidelines, diagrams, etc.), or tool for SRS (See Table I).

TABLE I. DESCRIPTION OF EMPIRICAL STUDIES SELECTED (1976-2008)

ID	Object of Study	Type of Task	Language expertise	Domain expertise	Problem size
S1	Stereotypes of UML sequence diagrams	Answering questions concerning the semantic of the diagrams, and describing the functionality of diagrams.	Students had knowledge about the use of stereotypes in general. They were taught about the specific stereotypes used	Not indicated	Not indicated.
S2	UML statecharts and/or sequence diagrams	Answering questions concerned several aspects of the system requirements.	Students had received classes on the modelling of OO software systems using UML	Familiarized with the domain of both systems used	Not indicated.
S3	Black-box (SCR) and White-box (UML) requirements specifications	Answering questions about the semantics of specifications	Training session of 3 hours	Familiarized with the domain	Black box specification: 12 pages White box specification: 20 pages
S4	Use Case Models	Answering questions about the Use case read	A half-day seminar on use cases.	Not indicated	Not indicated
S5	Multimedia requirements specification	Answering questions about the semantics of specifications	Training period is not indicated	Not indicated.	Not indicated
S6	Object oriented models and process oriented models	Answering questions based on the model provided.	Students had some prior experience with process-oriented modeling;	Not indicated	The problem size is not indicated
S7	Use Case Models	Answering questions about the functionality specified in the use case model with different guidelines.	None of the students were familiarized with the guidelines. Training session of two hours for basic use case modeling, and one hour for the use of guidelines.	Few subjects with application domain experience.	Use case model consisted of 7 actors and 9 use cases.
S8	Data flow diagrams and textual specification.	Identification of the functions that are performed by the system and answering questions about the problem-domain scenarios.	Programmers were not familiarized with data flow diagram	Familiarized with the problem domain	"Toy problems"
S9	saUML sequence diagrams	Answering questions formulated with two levels of abstraction: comprehensibility and application.	Students were mostly Java juniors with basic concurrency concepts.	Students receive a brief introduction to the readers/writers problem.	Not indicated
S10	UML sequence,	Answering questions about	Students received a course	Not indicated.	Not indicated

ID	Object of Study	Type of Task	Language expertise	Domain expertise	Problem size
	collaboration and state diagrams	the semantics of specifications	about UML.		
S11	UML sequence and collaboration diagrams	Answering five multiple-choice questions to express the comprehensibility of interaction diagram.	Familiarized with both types of interaction diagrams	Small examples of both MIS and real-time systems were used previous to the experiment.	MIS system Real-time system
S12	Language UML-B language B	symbols interpretation, modification on models	Training session of 8 hours for Language B, and 1 hour for UML-B.	Not indicated	"Toy problems"
S13	Z specification	Finding relevant part of the specification Understanding the notation Modifying the specification by writing an extra feature.	Students that had just finished one semester course in Z.	Not indicated	Specification A: monolithic, 121 lines Specification B: modularized, about 20 lines per module. Specification C: highly modularized
S14	Object oriented (OO) and Process oriented (PO) methods	Problem solving with behavioral and structural features	Training session of 3 hours for OO method, 3 hours for PO method	Familiarized with the domain	Each problem description was about 1 page for accounts payable and employee benefits systems.
S15	UML interaction diagrams (Sequence and collaboration)	Answering questions on diagrams comprehension and perceived comprehensibility.	Students had received a course where both types of interaction diagrams were taught. A test was applied before the experiment execution	Familiarized with the domains	Library MIS: 1 actor and 9 interactions. Security system: 2 actors and 11 interactions.
S16	Stereotypes of UML models (class and collaboration diagrams)	Answering questions related to the syntaxes of the stereotypes used.	Students had enough knowledge of UML models. The notion of stereotypes was introduced previous to the experiment.	Not familiarized with the Telecommunication domain	30 objects in collaboration diagrams, and 14 classes in class diagram.
S17	UML Statechart diagrams	Answering questions Completing the functional specification.	Informative seminar on UML Statechart diagrams for professionals	Familiarized with the domain	Size and complexity "representative" of a real-life case
S18	UML Statechart diagrams	Answering questions about the semantics of specifications	Skill of the second group of students using UML statechart diagrams, was much lower than first group.	Not indicated	Simple models of 9 states for the phone call system.
S19	UML sequence, collaboration, and state diagrams.	Answering questions about the semantics of specifications	Training time is not indicated. 6 groups according to the knowledge level in UML.	Not indicated.	Average operation size for Make Phone Call:8, Lend Item (Library):9, and Play message: 16
S20	The Object-Process Method (OPM) The Object Modeling Technique (OMT)	Answering questions about the semantics of specifications	4 weeks for OMT 3 weeks for OPM	Not indicated.	10 lines of description textual. Home Heating System.
S21	UML interaction and collaboration diagrams	Answering questions related to ordering and activity information.	Some previous experience with UML diagrams	Familiarized with both ATM and Lift scenarios	Four diagrams, each one of 30 interactions.
S22	Nesting level of UML Statechart diagrams	Answering questions about the semantics of specifications	short training of UML statechart diagrams	Not indicated	Not indicated
S23	UML class, sequence, and use case diagrams	Answering questions about the semantics of specifications	Students-> training time: five weeks for UML. Professionals with more than two years using UML.	Not indicated	Model between three and nine classes.
S24	Stereotypes of UML models (collaboration and class diagram)	Answering questions about the semantics of specifications	Students-> 45 min for introducing the notion, and usage of stereotypes. Professionals experienced with UML and stereotypes.	Students not familiarized. Professionals familiarized with the domain.	30 objects in collaboration diagrams, and 11 or 14 classes in class diagram.

As shown in Figure 1, UML diagrams were the most studied (e.g. use case, sequence, collaboration, state, object diagrams) by the researchers. 17% of the studies focused on the analysis of the effect of stereotypes on comprehensibility of requirements specifications. Comparisons between UML and non-formal languages has received slightly more attention than comparisons with formal languages.

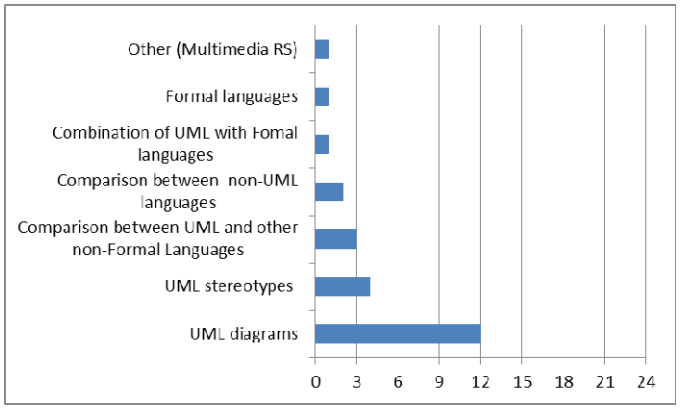

Figure 1. Distribution of the objects analyzed in the 24 studies

With respect to the *type of task* performed, 87.5% of the studies externalized the comprehensibility by means of questions formulated about the semantic of specifications. However, quality of these questions is not discussed by the majority of the researchers. Only four studies (S1, S9, S12, S14) considered an underlying theory in the formulation of comprehensibility questions. For example, in S9, according to Bloom's taxonomy[20] the questions were formulated at two levels of abstraction: comprehensibility and application.

The following three factors are very strongly related to the external validity[2].

With respect to *language expertise*, only 1 of out 24 studies involved professionals experienced with UML and familiarized with the usage of stereotypes (S24). The rest of studies had to conduct a training process to teach the modeling language under study. The training period observed in these studies was very varied. It oscillates from 45 minutes until 5 weeks as maximum. Furthermore, only 1 study (S8) did not require that the subjects (Programmers) were familiarized with the modelling language (data flow diagram).

With respect to *domain expertise*, 50% of the studies do not mention any level of the subjects' familiarization with the problem domain. S7 mentioned that few subjects had some experience with the domain under study, which were randomly assigned to the treatments with the purpose of blocking the influence of this variable (domain) on comprehensibility. Only S11 analysed this variable for evaluating the difference in diagrams comprehension for the Management Information Systems (MIS) and Real Time domains.

Considering the combination of these two factors (language and domain expertise) we have three main potential types of subjects, summarized as follow:

- A subject has less knowledge on the modeling language than knowledge of the domain (e.g. clients);
- A subject has an equal knowledge on the modeling language and knowledge of the domain (e.g. requirements engineers); and
- A subject has more knowledge on the modeling language than knowledge of the domain (e.g. analysts).

Although the 24 studies involved students as subjects, the majority of them evaluated the SRS comprehensibility from "analyst" perspective. Training sessions were more focused on the modelling language than the problem domain.

With respect to the *problem size*, we note a diversity of units of measure to express the problem size (e.g. number of elements of model, number of lines, number of pages). The majority of these studies (62%) recognized the lack of external validity due to use of "toy problems". Only 1 study (S17) affirmed that the problem size was "representative" of a real-life case. However, the "representative" term was not expressed in a quantitative way. On the other hand 38% of studies described the problem but its size is not mentioned.

III. CHECKLIST-BASED EVALUATION ON RELEVANCE OF EXPERIMENTAL STUDIES ON COMPREHENSIBILITY

The review process included two practitioners, the second and the forth authors of this paper. Prior to becoming university researchers, they worked as RE consultants. The first has 10 years of RE experience in industry and the second 8 years. Both are active members of the RE community in their areas of experience. The two practitioners worked in two different locations, used a checklist independently from each other, and had no communication between them. Next, the checklist used to evaluate the relevance of these 24 studies is described.
We make the note that using two practitioners only is not enough to obtain generalizable results, and that including more practitioners is important. However, we thought that the learning experience in the evaluation can still be valuable because of its potential to open up new questions for discussion and reflection on our future work.

A. *Practitioner's checklist*

Based on the approach of Kitchenham [5], a checklist was designed to be used in the RE community. It consists of 19 items formulated as questions concerning the information required for the practitioner's perspective [17]. A practitioner is someone who provides summary information for use in industry and wants to know whether the results in a paper are likely to be of value to his/her company or clients. Each question is rated using a 3-point ordinal scale ("2" when a question can be answered as "yes", "1" when a question can be answered "partially", and "0" when the answer is "no").

We had two raters who answered the 19 questions in the checklist while reading all 24 papers. Calculating the inter-rater reliability [13] of the questions of the checklist, 9 of out 19 questions showed an "acceptable agreement" ($\kappa_w > 0.21$) [17]. Based on this we discarded 10 questions because the agreement level among the answers of the reviewers was too low.

[2] It refers to the approximate truth of conclusions to generalize the results of our experiment to industrial practice [21].

Table II shows the 9 questions that were used in this study , as well as the respective rationale for each question.

TABLE II. QUESTIONS OF THE PRACTITIONER' CHECKLIST

Item	Question	Rationale/ Consultants need	Inter-rater Reliability [13]
Q1	Is the claim supported by believable evidence?	To be sure that any claims are supported by evidence	1
Q2	Is it claimed that the results are industry-relevant?	To clearly see whether the conclusions/results have practical relevance and why the authors think so	0.476
Q3	How can the results be used in practice?	Guidance by the authors on how the results would be used in industry	0.427
Q4	Is the result/claim useful/ relevant for a specific context?	To know the context in which the results are expected to be useful	0.474
Q5	Is it explicitly stated what the implications of the results/conclusion for practice are?	To get explicit information on the implications of the authors' work for practice	0.560
Q6	Are the results of the paper viable in the light of existing research topics and trends?	To know how the current work in the paper relates to current research trends.	0.660
Q7	Is the application type specified?	To know what type of applications the results apply to. In particular whether the results are specific to particular types of application (e.g. finance, or command and control etc.)	0.220
Q8	Do the authors show that the results scale to real life?	To be sure that the results scale to real life	0.514
Q9	Is the experiment based on concrete examples of use/application or only theoretical models?	To be sure that the results have a clear practical application.	0.9

B. Validity evaluation of the results obtained in using the checklist

Conclusion validity. Although our evaluation is based on the questions in the checklist that had an acceptable level of inter-rater reliability, there is a threat that the number of questions used cannot be enough for evaluating the relevance of the empirical studies from practitioner perspective. To reduce this threat our evaluation was complemented with the analysis of each one of the 24 studies in terms of the variables that affecting the comprehensibility (problem size, type of task, user expertise respect to the domain and modeling language) which are strongly related with the external validity.

Construct validity. Evaluator expectancies can bias the results of a study unconsciously, more even if we include feedback of two practitioners only. However, including more evaluators – by this we mean engaging more RE-practitioners from companies, was infeasible. We did have access to practitioners interested in the study, however there were resource constraints in their organizations that companies that impeded their participation.

We know that bringing more practitioners as evaluators it's better. However, we also note that our evaluations (see the next Section) brought some strong conclusions (e.g. that 80% of the reported studies do not scale up to real life, see Q8 and Figure 5 on the next page). Therefore, given the strength of the conclusions, in this case we think that more precision would not add much more value because bringing in a 3rd and 4th evaluator would not change much the results.

IV. RESULTS OBTAINED FROM CHECKLIST

Next, we discuss each one of the questions that had an acceptable agreement for the 24 empirical studies reported.

Q1. Is the claim supported by believable evidence?

This question asks whether the reader can buy into it. If pieces of evidence build upon each other, the reader is likely to deem the evidence worthy of being accepted as true or reasonable. In our findings, the evaluators had 100% agreement regarding believability of the papers that were reviewed.

Q2. Is it claimed that the results are industry-relevant?

This question checks (i) whether the research was motivated by some real-life problem experienced in industry, e.g. in software projects, in carrying our certain tasks by professional requirements engineers, (ii) whether the conclusions relate to what was claimed in the motivation regarding the practical problems, and (iii) what analytical argument the authors use to convince the readers that their conclusion have relevance. We found that 58% of the studies, from a practitioner' viewpoint, reported results not relevant for industry (See Figure 2)

Q3. How can the results be used in practice?

This question checks whether the authors clearly indicate how the results can be used in a real-life project. We found that 62% of the 24 articles propose a plan on how can be the results used in practice. (See Figure 2).

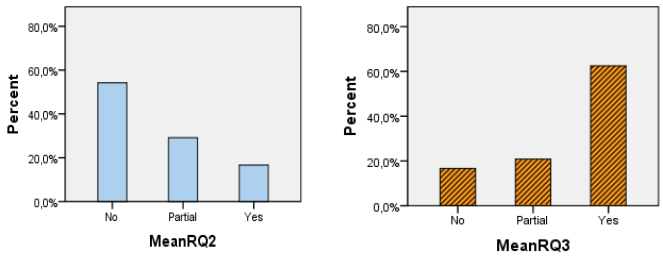

Figure 2. A frequency Distribution for Q2 and Q3

Q4. Is the result/claim useful/relevant for a specific context?

This question refers to the extent to which authors characterize the context where their proposals should work best. Context details are instrumental for practitioners to make a judgement about whether or not the proposal is relevant to their specific work settings. As shown in Figure 3, 54% of the articles do not specify the context in which the results are expected to be useful.

Q5. Is it explicitly stated what the implications of the results/conclusion for practice are?

This question checks whether or not the authors include a section that discusses the implications of their research for practice. We found that 67% of the articles reviewed did not address the implications of their results for practice (See Figure 3).

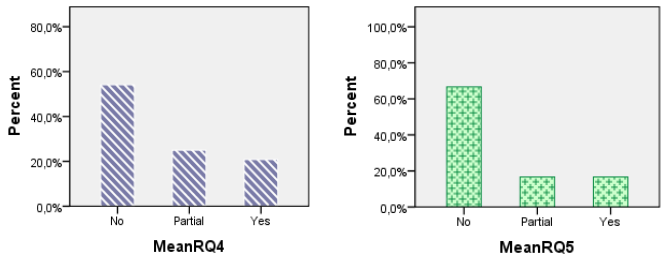

Figure 3. A frequency Distribution for Q4 and Q5

Q6. Are the results of the paper viable in the light of existing research topics and trends?

This question is to judge how the results are positioned with respect to other findings known in literature on the same topic (comprehensibility of SRS) with respect to the same artifact (e.g. an UML model, a statechart). Usually, the answers to this question are in the Related Work section, or in the Discussion section of a paper. As shown in Figure 4, 62% of the articles discussed their contribution with respect other findings on similar topic.

Q7. Is the application type specified?

This question checks whether or not the authors give enough details about the application type and the implications of this for their research. As shown in Figure 4, 50% of the articles specified partially the type of application to be used.

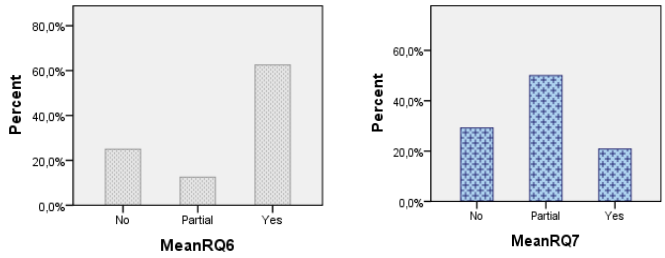

Figure 4. A frequency Distribution for Q6 and Q7

Q8. Do the authors show that the results scale to real life?

This question checks whether or not the authors discuss the assumptions they make about using their proposal in real life projects. We found that 80% of the article do not scale up to real life (See Figure 5).

Q9. Is the experiment based on concrete examples of use/application or only theoretical models?

This question asks whether the authors take examples from a real-life project as a starting point in their empirical work or they deliberately take a simplified example used in text books, or in their own teaching (theoretical models). As shown in Figure 5, all articles except one take examples from a real-life project.

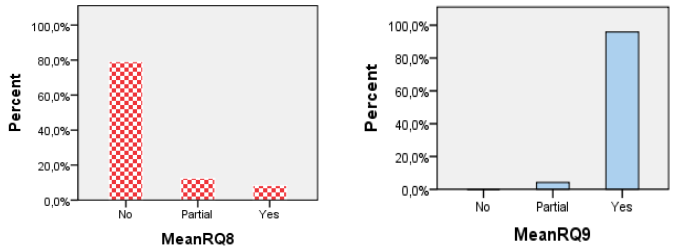

Figure 5. A frequency Distribution for Q8 and Q9

V. CONCLUSIONS AND FUTURE WORK

In this paper we report on a preliminary evaluation of the relevance of results in empirical studies on comprehensibility of requirements specifications. This evaluation was carried out using a checklist that was adapted from the Kitchenham approach [5].

With respect to our primary research goal, we found that a majority of experiments reported have significant limitations with respect to the artifacts and subjects utilized, which does not facilitate the application of research results to industry (generalizability of results). This clearly indicates the need for more research to evaluate SRS techniques in real-life settings.

On the other hand, we found that 88% of the studies externalized the comprehensibility by means of questions about the semantic of specifications. However, there is a lack of underlying theory in the formulation of these comprehensibility questions. This means that we could, and should, bring the insights of other research areas to our empirical evaluations on

comprehensibility (e.g. cognitive psychology). This lack of underlying theory affects unfavourably the construct validity, which is also related to generalizing the results of the experiment.

With respect to our secondary goal, we found that while the checklist is in no way perfect, we found it to fit the purpose of our study and yet be inexpensive to use. This encourages us using in follow up research, for example, in evaluating the relevance of empirical studies on comprehensibility of SRS published between after 2008. This will both complement the results we have now regarding the studies themselves and regarding the checklist.

We also welcome RE experiments with more relevance to reality. If we as a community generate and continually collect experience reports that relate to the challenges and the possible solutions of RE practitioners in their day to day work, our knowledge about how to leverage our research output to software project or process success would increase.

We plan to carry out a second and much larger evaluation of the relevance of empirical studies in RE, once the checklist is validated. To do this validation, we are going to adjust the rest of questions of the checklist [17], and to evaluate the effectiveness of the checklist by means of independent researchers/practitioners. For this purpose, we consider those practitioners from companies in the Netherlands who already participate in industry-university projects with the University of Twente. We also wanted to compare our experiences with other researchers/practitioners that could possibly use the checklists. We therefore invite other researchers from the empirical SE and RE communities to share ideas about how we could improve both our checklist and our checklist-based evaluation approach.

VI. ACKNOWLEDGEMENT

This work was in part funded by the EU Marie Curie Fellowship Grant 50911302 PIEF-2010.

APPENDIX : STUDIES INCLUDED

S1. Genero M., Cruz-Lemus J., Caivano D., Abrahão S., Insfrán E., Carsí J., Assessing the Influence of Stereotypes on the Comprehension of UML Sequence Diagrams: A Controlled Experiment. Model Driven Engineering Languages and Systems. 11th International Conference, MoDELS 2008, Toulouse, France, September 28 - October 3, 2008. Proceedings, pp 280-294.

S2. Gravino C., Scanniello G. and Tortora, G. An Empirical Investigation on Dynamic Modeling in Requirements Engineering. Model Driven Engineering Languages and Systems Conference, MODELS 2008, pp. 615-629

S3. Kamsties E., Antje von Knethenb, Reussner R.. A controlled experiment to evaluate how styles affect the understandability of requirements specifications, Information and Software Technology Journal, 45 (2003) 955–965.

S4. Cox K., Phalp K., Shepperd M., Comparing Use Case Writing Guidelines, 7th Int. Workshop on Requirements Engineering: Foundation for Software Quality. REFSQ 2001. Interlaken, Switzerland.

S5. Overmyer S., A Methodology for Constructing User-Oriented Requirements Specifications for Large-Scale Systems Using Electronic Hypermedia. Requirements Engineering (1999) 4:1–18, Springer Verlag.

S6. Agarwal R., Prabuddha D, and Sinha A.. Comprehending Object and Process Models: An Empirical Study, IEEE Transactions on Software Engineering, 25(4):541-556, August 1999.

S7. Anda B., Sjøberg D., and Jørgensen M., Quality and Understandability of Use Case Models. ECOOP 2001, Object-Oriented Programming, LNCS, Springer Berlin, 402-428

S8. Cioch F., Measuring Software Misinterpretation, Journal Systems Software, Elsevier, 1991, 14:85-95

S9. Xie S., Kraemer E., and Stirewalt R., Empirical Evaluation of a UML Sequence Diagram with Adornments to Support Understanding of Thread Interactions. 15th IEEE International Conference on Program Comprehension (ICPC'07), 2007

S10. Otero M., Dolado J., An Initial Experimental Assessment of the Dynamic Modelling in UML, Empirical Software Engineering Journal, 7, 27–47, 2002. Kluwer Academic Publishers.

S11. Glezer Ch., Nahmani E., Shoval P., Experimental Comparison of Sequence and Collaboration Diagrams in Different Application Domains, Proceedings of the Workshop on Evaluating Modeling Methods for Systems Analysis and Design, (EMMSAD'05), Portugal, 2005, pp. 463-476.

S12. Rozilawati R., Snook C., Poppleton M., Comprehensibility of UML-based Formal Model – A Series of Controlled Experiments, WEASELTech'07, November 5, 2007, Atlanta Georgia, USA, Copyright ACM

S13. Finney K., Fenton N., Fedorec A., Effects of structure on the comprehensibility of formal specifications, IEE Software. 146(4): 193-202, August 1999

S14. Agarwal R.; Sinha A.; Tanniru M. Cognitive fit in requirements modeling: A study of object and process method, Journal of Management Information Systems; Fall 1996. pg. 137

S15. Glezer Ch., Last M., Nachmany E., Shoval P.. Quality and comprehension of UML interaction diagrams-an experimental comparison, Information and Software Technology 47 (2005) 675–692.

S16. Kuzniarz L., Staron M., Wohlin C., An Empirical Study on Using Stereotypes to Improve Understanding of UML Models, Proceedings of the 12th IEEE International Workshop on Program Comprehension (IWPC'04)

S17. Cruz-Lemus J., Genero M., Morasca S., and Piattini M., Using Practitioners for Assessing the Understandability of UML Statechart Diagrams with Composite States, J.-L. Hainaut et al. (Eds.): ER Workshops 2007, LNCS 4802, pp. 213–222, 2007. Springer-Verlag Berlin Heidelberg 2007.

S18. Cruz-Lemus J., Genero M., Manso E., and Piattini M. Evaluating the Effect of Composite States on the Understandability of UML Statechart Diagrams. L. Briand and C. Williams (Eds.): MoDELS 2005, LNCS 3713, pp. 113-125, 2005. Springer-Verlag Berlin Heidelberg 2005.

S19. Otero M., Dolado J., Evaluation of the comprehension of the dynamic modeling in UML. The Journal of Information and Software Technology 46(2004) 33–53.

S20. Peleg M., Dori D., The Model Multiplicity Problem: Experimenting with Real-Time Specification Methods. IEEE Transactions on Software Engineering, 26(8)742-759, August 2000.

S21. Swan, T. Barker, C. Britton, M. Kutar, "An empirical study of factors that affect user performance when using UML interaction diagrams," ISESE, pp.10 pp., 2005 International Symposium on Empirical Software Engineering, 2005.

S22. Cruz-Lemus J., Genero M., Piattini M., and Toval A., An Empirical Study of the Nesting Level of Composite States Within UML Statechart Diagrams, J. Akoka et al. (Eds.): ER Workshops 2005, LNCS 3770, pp. 12 – 22, 2005.

S23. Lange Ch., Chaudron M., Effects of Defects in UML Models – An Experimental Investigation. ICSE'06, May 20–28, 2006, Shanghai, China Copyright 2006 ACM.

S24. Staron M., Kuzniarz L., Wohlin C., Empirical assessment of using stereotypes to improve comprehension of UML models: A set of experiments, Journal of Systems and Software, 2006, pp. 727-742.

REFERENCES

[1] B. Cheng, J.M. Atlee, Research Directions in Requirements Engineering, Future of Software Engineering (FOSE) - ICSE 2007, IEEE CS Press, pp. 285–303

[2] Condori Fernandez N., Daneva M., Sikkel K., Wieringa R.J., Dieste O., Pastor O., A Systematic Mapping Study on Empirical Evaluation of Software Requirements Specifications Techniques. In: ESEM2009. IEEE CS Press, pp. 503–505.

[3] Davis A., Dieste O., Hickey A., Juristo N., Moreno A., Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review, In: Int. Requirements Engineering Conference, 2006, pp. 176–185.

[4] Davis A., Hickey A., Dieste O., Juristo N., Moreno A., A Quantitative Assessment of Requirements Engineering Publications - 1963-2006. In: REFSQ, 2007, Springer, 129–143.

[5] Kitchenham, B.A., Al-Kilidar H., Babar, M.A., Berry, M. Cox, K., Keung L., Kurniawati, F. Staples M., Zhang H., Zhu L, Evaluating guidelines for reporting empirical software engineering studies. Empirical Software Engineering 13(1), 2008, pp 97–121.

[6] Kitchenham, B., Procedures for Undertaking Systematic Reviews, Joint Technical Report, Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd. (0400011T.1), 2004.

[7] Racheva, Z., Daneva, M., Sikkel, K., Value Creation by Agile Projects: Methodology or Mystery? In: Int. Conf. on Product-Focused Software Process Improvement, Springer, 2009, pp. 141–155.

[8] Kitchenham B.A., T. Dybå, and M. Jørgensen, "Evidence-Based Software Engineering," Proc. 26th Int'l Conf. Software Eng, IEEE CS Press, 2004, pp. 273–281.

[9] Dybå, T. and Dingsøyr, T. (2008) Empirical Studies of Agile Software Development: A Systematic Review, Information and Software Technology, 50(9-10): 833–859.

[10] Sjøberg, D.I.K., Dybå, T., and Jørgensen, M. (2007) The Future of Empirical Methods in Software Engineering Research, FOSE'07, IEEE CS Press, pp. 358–378.

[11] Jedlitschka, A., Ciolkowski, M. & Pfahl, D. (2008), Reporting experiments in software engineering, in F. Shull, J. Singer & D. Sjøberg, eds, 'Guide to Advanced Empirical Software Engineering', Springer, London, chapter 8.

[12] Higgins, J.P.T., S. Green, Cochrane Handbook for Systematic Reviews of Interventions, Wiley, 2008.

[13] Cohen J. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. Psychol Bull.1968 ;70:213–220

[14] Gwet, K. L. Computing inter-rater reliability and its variance in the presence of high agreement, British Journal of Mathematical and Statistical Psychology, 61, 2008, pp 29–48.

[15] Aranda, J., Ernst, N., Horkoff, J., Easterbrook, S.: A framework for empirical evaluation of model comprehensibility. In: MISE 2007: Proceedings of the International Workshop on Modeling in Software Engineering,Washington, DC, USA, p. 7. IEEE Computer Society, Los Alamitos (2007)

[16] Wieringa R. 1998. A survey of structured and object-oriented software specification methods and techniques, ACM Computing Surveys (CSUR); 30 (4) : 459–527

[17] Daneva M. , Condori-Fernandez N., Sikkel K., Herman A.: Experiences in Using Practitioner's Checklists to Evaluate the Relevance of Experiments Reported in Requirements Engineering. Technical Report TR-CTIT-11-16, Centre for Telematics and Information Technology, University of Twente, Enschede, 2011

[18] Krogstie J.: Using a Semiotic Framework to Evaluate UML for the Development of Models of High Quality. Unified Modeling Language: Systems Analysis, Design and Development Issues 2001: 89–106

[19] Shima K., Takemura Y., Matsumoto K., "An Approach to Experimental Evaluation of Software Understandability," ISESE, pp.48, 2002 International Symposium on Empirical Software Engineering (ISESE'02), 2002

[20] Bloom's Taxonomy: A Forty-Year Retrospective. Anderson & Sosniak, 1994

[21] Wohlin C, Runeson P, Höst M., Ohlsson M.C., Regnell B., Wesslén A. (2000) Experimentation in Software Engineering: an introduction. Kluwer, Dordrecht. 228 p.

[22] Reijers, H.A.; Mendling, J.; , "A Study Into the Factors That Influence the Understandability of Business Process Models," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.41, no.3, pp.449–462, May 2011.

.