

Requirements Reuse and Requirement Patterns: A State of the Practice Survey

Cristina Palomares¹, Carme Quer², Xavier Franch³

1. Group of Software and Service Engineering (GESSI)
Universitat Politècnica de Catalunya (UPC), Barcelona (Catalonia, Spain)
Jordi Girona num.1-3, Omega building, room S206
cpalomares@essi.upc.edu
Phone: +34 93.413.7174, Fax: +34 93.413.7833
2. Group of Software and Service Engineering (GESSI)
Universitat Politècnica de Catalunya (UPC), Barcelona (Catalonia, Spain)
Jordi Girona num.1-3, Omega building, room 124
cquer@essi.upc.edu
3. Group of Software and Service Engineering (GESSI)
Universitat Politècnica de Catalunya (UPC), Barcelona (Catalonia, Spain)
Jordi Girona num.1-3, Omega building, room 122
franch@essi.upc.edu

Abstract —

Context. Requirements engineering is a discipline with numerous challenges to overcome. One of these challenges is the implementation of requirements reuse approaches. Although several theoretical proposals exist, little is known about the practices that are currently adopted in industry.

Objective. Our goal is to contribute to the investigation of the state of the practice in the reuse of requirements, eliciting current practices from practitioners, and their opinions whenever appropriate. Besides reuse in general, we focus on requirement patterns as a particular strategy to reuse.

Method. We conducted an exploratory survey based on an online questionnaire. We received 71 responses from requirements engineers with industrial experience in the field, which were analyzed in order to derive observations.

Results. Although we found that a high majority of respondents declared some level of reuse in their projects (in particular, non-functional requirements were identified as the most similar and recurrent among projects), it is true that only a minority of them declared such reuse as a regular practice. Larger IT organizations and IT organizations with well-established software processes and methods present higher levels of reuse. Ignorance of reuse techniques and processes is the main reason preventing wider adoption. From the different existing reuse techniques, the simplest ones based on textual copy and subsequent tailoring of former requirements are the most adopted techniques. However, participants who apply reuse more often tend to use more elaborate techniques. Opinions of respondents about the use of requirement patterns show that they can be expected to mitigate problems related to the quality of the resulting requirements, such as lack of uniformity, inconsistency, or ambiguity. The main reasons behind the lack of adoption of requirement patterns by practitioners (in spite of the increasing research approaches proposed in the community) are related to the lack of a well-defined reuse method and involvement of requirement engineers.

Conclusion. The results of our paper are interesting for practitioners since we highlight relevant observations from the survey participants' experiences when deciding to implement requirements reuse practices. We also suggest future lines of research based on the needs pointed out in the results.

Keywords — Requirements engineering; requirements reuse; requirement patterns; empirical studies; surveys; online questionnaire.

1 Introduction

The final quality of a software system greatly depends on the quality of its requirements, which are documented in its software requirements specification. However, evidence shows that the state of the practice for acquiring these requirements is still far from satisfactory (Mich et al. 2004, Sadraei et al. 2007, Cox et al. 2009, Solemon et al. 2009, SwissQ 2012a, Schirpenbach 2014, Young 2015). Mich et al. (2004), for instance, asked more than 150 participants working in software development to name the two things in their job that they would like to do more efficiently. The activity that was named most frequently (46%) was “identify user requirements”.

Without a proper set of requirements, any software project has a great chance of failing, no matter how well the rest of the project is executed. A study of non-outsourced projects (Hall et al. 2002) found that out of 268 documented development problems, requirements-related problems accounted for 48%. In a literature review performed by Arnuphaptrairong (2011), five out of eight lists of top ten software project risks analyzed had the misunderstanding of requirements as a frequent risk. A study performed by SwissQ (2012a) asked over 300 participants, among other issues, for problems related to requirements engineering (RE). The results showed that 75% of the participants usually made linguistic mistakes (i.e. ambiguousness, incomprehensibility and unquantifiability), 74% made content mistakes (i.e. incompleteness and wrong statements), and 61% made logical mistakes (i.e. inconsistency and redundancy). These observations seem not to be related to a particular development method, for example, 75% of all respondents in the survey worked with Agile methods, which shows that RE is still high risk even when adopting this approach. Other studies have also identified requirements as being an important risk factor in project failures (Beecham et al. 2003, Tiwana 2004, Sommerville et al. 2005, Galorath 2012, Calleam Consulting 2015). It has also been reported that the cost of fixing requirements-based problems increases rapidly the farther into the software development they are discovered (Boehm and Basili 2001).

Eliciting the suitable requirements produces benefits, such as preventing errors, improving quality, and reducing risk in software projects (Procaccino et al. 2002, Jones 2014). Requirements reuse has been proposed as an advanced requirements elicitation technique that can be a key asset in obtaining software requirements specifications of better quality through more effective engineering processes (Sawyer et al. 1997, Sommerville and Sawyer 1997, Niazi et al. 2008, Wiegers and Beatty 2013, Jones 2014, Méndez and Wagner 2015). In a nutshell, requirements reuse is the concept of picking up requirements that have been written for previous projects and using them in a new one.

Owing to the proven benefits of reuse in different software engineering areas, requirements reuse, its success factors, barriers and processes, are increasingly attracting the interest of researchers. In a study about research fields in RE and future trends, Nuseibeh and Easterbrook (2000) highlighted the need to

establish efficient methods for requirements reuse. Several industrial applications have showed benefits of reuse in the requirements engineering area (Issa et al. 2010, Goldin and Berry 2013, Pacheco et al. 2015). Major journals and conference proceedings include publications of scientific works with reuse proposals and studies (see Section 2). Furthermore, comprehensive RE textbooks even include full chapters focused on software reuse (e.g., see Wiegers and Beatty (2013)).

In addition to requirements reuse in general, some approaches seem to be particularly emergent recently. Several research indicators point out the emergence of one of those approaches, namely requirement patterns (Withall 2007). Among these indicators, we can mention: the existence of a series of workshops in the IEEE RE conference (RePa, five editions at RE 2012–2016¹); the inclusion of tutorials on this topic on the ACM/IEEE ICSE program (three tutorials at ICSE 2013–2015); 61 out of the 105 papers presenting pattern proposals for reuse in requirements engineering were published between 2010 and 2015 (see Section 2.1).

In spite of this interest by the research community, it is well-known that research interests do not always match industry needs (Lo et al. 2015). A natural question then is: to what extent are IT companies reusing requirements? Do they see advantages in a pattern-based approach? Some indicators seem to point out that requirements reuse is becoming a regular practice in industry: the latest releases of requirement management tools are including requirements reuse as part of their functionality (e.g., Doors², Jama³, and Visure Requirements⁴) and we are witnessing the creation of social media groups dedicated to the topic (e.g., *Reuse in Requirements Engineering* LinkedIn group). Nevertheless, these indicators are not enough to answer the question above and this is the main motivation of our work.

The goal of this paper is to report the results of a survey that we conducted in order to explore the current situation of requirements reuse practices in IT companies, both in general and with respect to patterns. The survey was implemented as an online questionnaire that was available from April 2013 to July 2014, and for which we obtained 71 responses from requirements engineers with industrial experience in the field. Once the results were analyzed, we derived some observations about the state of the practice of the respondents' organizations that may be interesting for RE practitioners and researchers. The analysis undertaken in this paper goes into much greater depth than the preliminary frequency analysis made on a subset of 50 responses presented as work in progress at the REFSQ conference (Palomares et al. 2014a).

The contents of the paper are as follows: Section 2 introduces the background and related work on existing academic proposals for software reuse and requirement patterns, as well as reported industrial applications of these proposals and existing surveys or interviews regarding the state of the practice. Section 3 presents the research questions addressed in this paper and the research approach used (i.e., the survey conducted). Section 4 states the relevant variables that characterize the respondents of the survey. Sections 5, 6, and 7 present the observations obtained regarding each research question, which in some cases are related to the characteristics of Section 4. Section 8 summarizes and discusses the observations derived from the survey results, and Section 9 describes the threats to the validity of the survey results. Finally, Section 10 presents the conclusions.

¹ <http://www.utdallas.edu/~supakkul/repal6/>

² <http://www-03.ibm.com/software/products/en/ratidoor/>

³ <http://www.jamasoftware.com/>

⁴ <http://www.visuresolutions.com/requirements-engineering-tool>

2 Background and Related Work

Subsection 2.1 presents a literature-based background on the subject of investigation, requirements reuse and requirement patterns. The other two subsections report existing related work. Subsection 2.2 focuses on technology transfer by reporting applications of academic proposals in industry. Subsection 2.3 reports on published surveys or interviews that provide an evidence-based state of the practice, as we aim at in our paper. Finally, Subsection 2.4 summarizes the results and justifies the need for the survey presented in this paper.

2.1 Background on Requirements Reuse and Patterns

The background that we include in this subsection is based on a literature search of publications from 1980 to 2015. We ran an automatic search of papers published not later than 2014 in several major digital libraries (IEEE Xplore, ACM Digital Library, Springer Link and Science Direct). The search looked for publications that include the words “reuse” and “requirements” either in the title or in the abstract and it was tailored to the capabilities of each library. We complemented the results with a manual search in the main venues that regularly include requirements engineering papers to make sure we included all relevant publications in 2015. After filtering by title, abstract, and quick read, 294 publications were selected. From them, 105 used requirement patterns, showing its importance as a reuse vehicle, especially in the last 6 years (2010–2015), where the proportion had grown to 61 out of the 129 publications in requirements reuse published in that period (47%). The 105 publications can be grouped into 80 proposals, since there were several papers that dealt with the same proposal.

In Table 1, we show a representative sample of the publications found. The sample has been chosen to be able to illustrate each value of each characteristic of a requirement reuse and patterns approach:

- Artifacts to be reused. They are mainly requirements written in natural language (e.g., Withall 2007), use cases (Chung and Supakkul 2006), and domain models (Haeng-Kon 2014). Other artifacts that may be reused, which are not strictly requirements, are ontologies (e.g., Salini et al. 2012), classifications of requirements in a requirements specification (e.g., Panis 2015, Franch et al. 2013a), or the relationships or dependencies among requirements or other reuse artifacts (Srivastava, 2013, Franch et al. 2013a, Konrad and Cheng 2002).
- Size of the artifacts to reuse. The size varies from individual requirements (e.g., Daramola et al. 2012) to requirement clusters (e.g., Chen et al. 2005) to requirement specification parts or even complete requirement specifications (e.g., Chung and Supakkul 2006).
- Process of reuse. Several proposals describe the process for applying requirements reuse, for instance those by Pacheco et al. (2015), Carrillo-de-Gea et al. (2013), and Renault et al. (2009).
- Repository. Although most proposals give some ideas about how to structure the repositories of reusable artifacts, we highlight Franch et al. (2013a) and Naish and Zhao (2011), which give detailed descriptions of how to construct such repositories and how to classify and identify artifacts that are suitable for reuse.
- Scope. Most of the published proposals are general even though the papers give examples of reuse in specific domains. Other proposals are specific for a particular domain, such as seismological systems (Li et al. 2012), embedded systems (Konrad and Cheng 2002), or security requirements (Daramola et al. 2012, Jensen et al. 2009), without any aim at generalization.

Table 1: Background on requirements reuse and patterns

	Process	Repository	Scope	Artifact	Size	Abstraction	Purpose
Carrillo-de-Gea et al. 2013	Specific process	Yes	General	Natural language requirements	Requirements	Patterns	Elicitation, specification
Chen et al. 2005	-	Yes	General	Natural language requirements	Requirement clusters	Feature models	Elicitation, specification
Chung and Supakkul 2006	-	-	General	Use cases, NFR diagrams	Requirement specifications	Patterns	Specification
Daramola et al. 2012	Specific process	Yes	Security	Natural language requirements, classification, ontology	Requirements	Templates	Elicitation, specification
Franch et al. 2013a, Renault et al. 2009	Specific process	Yes	General	Natural language requirements, classification, dependencies	Requirement clusters	Patterns	Elicitation, specification
Haeng-Kon 2014	-	Yes	Adaptive human management systems	Domain models	Requirement clusters	Patterns	Specifications analysis
Hauksdottir et al. 2012	Specific process	Yes	General	Natural language requirements	Requirement clusters	Templates variability models	Elicitation, specification
Jensen et al. 2009	Specific process	Yes	Healthcare systems security	Natural language requirements	Requirements	No	Elicitation, specification
Konrad and Cheng 2002, 2005	-	-	Embedded systems	Diagrams, relationships	Requirement clusters	Patterns	Specification
Li et al. 2012	Specific process	-	Seismology systems	UML diagrams	Requirement clusters	Patterns	Elicitation, specification
Mannion and Kaindl 2008	Specific process	Yes	General	Natural language requirements	Requirement clusters	Feature models	Elicitation, specification
Massonet and van Lamsweerde 1997	Analogical queries	Yes	General	KAOS models	Requirement specifications	No	Specifications analysis
Naish and Zhao 2011	Specific process	Yes	General	-	-	Patterns	Elicitation, specification
Pacheco et al. 2015	Specific process	Yes	General	Natural language requirements, classification	Requirements	Templates	Elicitation, specification
Panis 2015	Specific process	Yes	General	Natural language requirements, classification	Requirements	Templates	Specification
Robertson and Robertson 2000	-	-	General	Requirements specification, classification	-	Document templates	Specification
Salini et al. 2012	-	Yes	Security	Natural language requirements, ontologies	Requirement clusters	Patterns	Elicitation, specification
Srivastava 2013	Specific process	Yes	Online examination systems	Natural language requirements, dependencies	Requirements	Patterns	Elicitation, specification
von Knethen et al. 2002	Specific process	Yes	General	Reusable components relationships	Groups of related components	Document templates	Elicitation, specification
Withall, 2007	Specific process	Yes	General	Natural language requirements	Requirements	Patterns	Elicitation, specification

- Abstraction. Some approaches reuse specific requirements or complete requirements specifications without any abstraction. Other proposals add abstraction to the reuse knowledge by using templates, patterns, or feature models. The lower level of abstraction is applied with the following techniques: templates about the content or structure of requirements specifications (Robertson and Robertson 2000, von Knethen et al. 2002); templates that are natural language sentences with no required structure (Hauksdottir et al. 2012); templates with a basic structure that may include parameters (e.g., Daramola et al. 2012, Pacheco et al. 2015); or requirements with a required structure that is compliant with language grammar (e.g. Konrad and Cheng 2005 applied in Post et al. 2011). At the highest level of elaboration we find patterns (e.g., Withall 2007, Li et al. 2012, Franch et al. 2013a) and feature models proposed in

domain engineering (Chen et al. 2005, Mannion and Kaindl 2008), which are those approaches based upon the notion of variability of requirements. As stated before, it is important to note the fact that patterns are proposed in 47% of the requirements reuse proposals identified between 2010 and 2015.

- Purpose. As represented by the references included in Table 1, one of the purposes of most proposals is the elicitation and/or specification of requirements. Another purpose is requirement specification analysis (Haeng-Kon 2014, Massonet and Van Lamsweerde 1997), where knowledge from past specifications is transferred to new specifications that share some significant similarity with the past ones.

2.2 Industrial Case Studies on Requirements Reuse and Patterns

The low percentage of papers on software engineering that include industrial validation was already reported by Lam et al. (1997). More recently, the results of the systematic mapping of reusable knowledge on security requirements by Souag et al. (2015) corroborate the results, since in their study only 10.5% of the identified papers performed an experimental validation in industry. In the case of the 294 publications reported in the previous subsection, only 72 conducted some empirical study to endorse the proposal and just a few of these studies were carried out in industry (e.g., Goldin and Berry 2013, Eriksson et al. 2009 and Rine and Nada 2000). Considering only the 80 proposals that use patterns (see previous subsection), 28 included some empirical study, and only 11 conducted the study in industry. In seven of the 11 proposals, the study consisted of obtaining practitioners' opinions or in constructing patterns for reuse in systems developed in specific companies. In the remaining four proposals, the studies consisted of the industrial application of requirements reuse artifacts and the measurements of the benefits and drawbacks of the reuse.

Table 2 summarizes these four pattern-based proposals. All of them show benefits in using patterns, being the aspects that they measured: the time required for requirements specification, the quality of the resulting specifications, and the completeness of the repository of knowledge to reuse considering the number of requirements reused. Three of them showed that the time required for the requirements engineering phase decreased because of the use of patterns (Issa et al. 2010, Pacheco et al. 2015, Toval et al. 2002). Two of them also reported an improvement in the quality of the resulting specifications (Karatat et al. 2014, Pacheco et al. 2015) and the other three did observations about the repository of patterns to reuse and the amount of requirements reused (Issa et al. 2010, Karatat et al. 2014, Toval et al. 2002).

Table 2: Industrial application of requirements reuse through patterns

	Company	Context	Application	Evaluation
Issa et al. (2010)	TestWarehouse, software warehouse company	Requirements elicitation and modeling	Approach used over 18 months in 6 projects	Time saved in the requirements engineering phase
Karatas et al. (2014)	Aselsan Inc., technology systems development company	Software product line development	Software engineers check requirement specifications for two projects derived from a requirements repository	Improvement of the quality of requirement specifications
Pacheco et al. (2015)	RAGASOFT, software development company	Development of products, sales and stock management system	Two teams did the specification of requirements, one that just was allowed to query past project specification and the other with the catalog and using the authors' proposal	Time saved in the requirements engineering phase Improvement of the quality of requirements specifications
Toval et al. (2002)	Spanish regional government	Risk security analysis	Development of the SIREN repository that helps to make security issues explicit from the early steps of a system development process	Time saved in the requirements engineering phase

2.3 Surveys and Interviews on Requirements Reuse

In order to look for surveys and interviews on the state of the art of requirements reuse in organizations, it was not suitable to use the same search results as those used in Subsections 2.1 and 2.2. In these two subsections, the search was for publications that include the words “reuse” and “requirements” in the title or in the abstract, and this probably excluded publications reporting interviews or surveys focusing on requirements engineering in general that also reference reuse practices. Therefore, we conducted a new search looking for papers published from 2006 that a) have the word “requirements” in the title and b) in the abstract talk about surveys, interviews, or questionnaires and about companies, organizations, or firms. We used the same digital libraries as used in the previous search. This new search yielded 184 papers and, after filtering them, we selected 17 papers presenting surveys, questionnaires, or interviews on the state of the art of requirement engineering practices used in organizations⁵. We applied snowballing using the same search criteria over this initial set and we identified 15 additional papers published since 2000 (which we considered a reasonable year to start the search taking into account the subject of papers we were searching). These 32 publications were read and, as a result, we identified 22 with no reference to requirements reuse, seven that were addressing requirement engineering practices and that had some question about requirements reuse, and only three that were fully focused on requirements reuse. Next we describe the results of the last two groups.

Surveys and interviews on RE in general. Table 3 is dedicated to the seven papers including some questions about requirements reuse. Except for one, the papers obtained data only from companies in a specific country, and not globally. Another fact to note is that most of the papers allowed participants with different roles, and not only people in charge of eliciting and specifying requirements. The results are that between the 40% and 82% of participants, depending on the study, state requirements reuse in their organizations as a practice that is always or widely followed.

⁵ We excluded our REFSQ 2014 paper since it was presenting some preliminary results of the study reported in this paper.

Table 3: Surveys and interviews on RE in general

	Type of study	Participants	Companies	Country	Percentage of participants that state requirements reuse as a practice always or widely followed
Cox et al. (2009)	Interviews	10 RE Experts	10	Australia	40%
Iqbal et al. (2013)	Survey	108 Different roles	18	Malaysia	82%
Khankaew and Riddle (2014)	Interviews	10 Different roles	11	Thailand	75%
Matulevicius (2005)	Survey	28 Different roles	28	Lithuania	50%
Niazi et al. (2012)	Survey	39 Req. engineers	39	Global	61%
Solemon et al. (2010)	Survey	64	Not stated	Malaysia	97% ⁶
Tahir and Ahmad (2010)	Survey/ interviews	27/5 Different roles	25	Malaysia	77%

Surveys and interviews fully focused on requirements reuse. Table 4 is dedicated to the three publications that belong to this category (we include only data and opinions that are relevant to our paper).

Bakar and Kasirun (2014) conducted a survey on requirements reuse in Malaysian software development, IT consultancy, research, and education companies. They obtained 36 responses. The majority of participants reused requirements in an ad hoc manner and just 19.4% were involved in systematic reuse processes. They reported benefits in the reuse of requirements but they stated the barriers to reuse as the low quality and incompleteness of requirements that are available for reuse. Although the survey included questions about the implications for project team members and project management when requirement reuse is applied, the authors neither reported nor analyzed the answers to these questions in the paper.

Chernak (2012) analyzed the state of the practice and benefits of reuse by means of an online survey. The survey received 82 valid responses from participants that were contacted through professional e-mail groups, IT-related websites, and direct e-mailing. One of the most relevant results of this survey was that, although only 59% of participants stated having adopted requirements reuse, 93% believe that reusing requirements is important and can provide benefits. Other results indicated that the respondents also thought that the way of implementing reuse in practice is not clear and that the main obstacles may be related to the maintenance and difficulty in use of the requirements repository and the low quality of reusable knowledge that companies have. Finally, it can be mentioned that 83% of participants (40 of 82 participants) mentioned regular reuse practices when deploying a new release of an existing application.

Hoffmann et al. (2013) interviewed five requirements analysts to solicit their opinions on the advantages and drawbacks of reuse through patterns. The five analysts had not used requirement patterns before the interviews. In general, they foresaw advantages in employing requirement pattern approaches within a company. Four of them foresaw efficiency in elicitation, three of them understandability and completeness of requirement specifications, and two of them requirements quality, comparability, and traceability. Specifically, they considered that it is advisable to define requirement patterns for non-

⁶ Includes participants that state the practice as occasionally used

functional requirements and recurrent functional ones. Finally, among factors that are relevant for the success of the practical application of patterns, they highlighted the following: the existence of a well-defined reuse method and its application process that should be introduced in a specific way to each organization; the reduction of the effort required thanks to the use of this method and application process; the definition of the requirement patterns in an appropriate language and detail level; the acceptance and implication of requirements engineers and managers of the reuse process; the guidance and support provided to requirement patterns users; the clarity of who is responsible for adding and maintaining patterns; and the existence of tool support.

Table 4: Surveys and interviews fully focused on requirements reuse

	Type of study	Participants	Companies	Country	Results
Bakar and Kasirun (2014)	Survey	36 Different roles with more than 1 year of experience in RE	-	Malaysia	<ul style="list-style-type: none"> • Requirements reuse (72.2% participants apply requirements reuse = 19.4% involved in systematic requirements reuse + 52.8% requirements reuse in an ad hoc manner). • Benefits (requirements to be reused are easy to understand as compared to defining new requirements, reuse gives positive impact to the requirements engineering performance, reuse increases the productivity of the development team). • Why not reuse requirements (requirements from previous projects are incomplete or do not exist, existing requirements are poorly structured, existing requirements are not kept updated). • Critical factors (existence of a tool that facilitates the search and selection of requirements to reuse).
Chernak (2012)	Survey	82 50% Business analyst 50% different roles	82	Global	<ul style="list-style-type: none"> • Requirements reuse (59% reused requirements in their latest projects). • Benefits (faster time-to-market, lower development cost). • Why not reuse requirements (low maintenance of the reuse repository, requirements to be reused are incomplete, difficulty of identifying requirements to reuse).
Hoffman et al. (2013)	Interviews	5 Requirements analysts	5	-	<ul style="list-style-type: none"> • Requirements reuse (convenient for recurring requirements). • Benefits expected (efficiency in elicitation, understandability of requirements, completeness of requirements specification, requirements quality, comparability of requirements, traceability). • Critical factors (adaptation of the strategy of introduction to each organization, reduction of the effort required by analysts and stakeholders, patterns with suitable language and detail, organizational changes accepted by all people implied, provision of information and support to pattern users, guide users using requirement patterns, clarity in who is responsible for patterns maintenance, existence of tool support).

2.4 Summary of Related Work

As a summary of the three subsections above, we can say that:

- There are a considerable number of publications that deal with reusing requirements and a considerable percentage are papers that propose the use of patterns as the abstraction level of the reuse knowledge. The

existent proposals differ in multiple aspects, one of them being the artifacts used and the level of abstraction in these artifacts.

- Few academic proposals of requirements reuse with patterns have been assessed or validated in industrial applications. The proposals that have been validated in an industrial context showed benefits in measuring the results of applying their artifacts, techniques, and/or tools.
- There are different surveys and interviews on the state of the practice of requirements engineering that address reuse. Only three of them are specifically focused on requirements reuse. The one that reports interviews on requirements reuse with patterns does not study the state of the practice, but the opinion of five professionals about the possible benefits/drawbacks of the use of patterns in requirements reuse.

The conclusion, with regard to related work, is that there is a need for work that gives more evidence on the benefits and drawbacks of the different abstractions and artifacts used in the reuse proposals. Although there are some studies that have conducted validation in industry, there are not enough to support the validity of the results. The surveys and interviews that ask practitioners about the requirement reuse state of the practice and that ask about their experience or opinion of the different techniques are interesting for requirement engineers that work on proposals to understand the potential usability of their approaches and the acceptance that they may have in industry.

3 Research Approach

Goal: The main goal of our study was to conduct an exploratory study of the practices in requirements reuse that are currently being used in organizations and to study in more depth the possible benefits and drawbacks of the use of patterns as a requirements reuse technique.

Research questions: The research questions (RQs) that drove our study are as follows:

- *RQ1: What is the current state of the practice of requirements reuse?* Here we investigated the current situation of requirements reuse practices in organizations (i.e., the level of requirements reuse, the type of requirements that are more likely to be reused, and the techniques used to implement the concept).
- *RQ2: Why are existing requirements reuse proposals not being used in industrial practice?* Given the results of the state of the art and the practice, we decided to explore the reasons that may hamper requirements reuse adoption in organizations and to report all of the identifiable barriers for to such adoption.
- *RQ3: What benefits and drawbacks can emerge from the use of a catalogue of software requirement patterns?* In this RQ, we asked the survey participants for their opinion about whether (and how) RE problems could be mitigated by the existence of a software requirements pattern catalogue (SRPs catalogue for short) and about critical aspects and barriers for its introduction in an organization.

Research method. In order to achieve our goal, we could use different empirical research methods, such as survey questionnaires, survey interviews, data aggregation of evidence from industrial case studies. Data aggregation would not be a good approach owing to the few publications regarding applications in industrial case studies (see Subsection 2.2). Interviews have clear advantages over questionnaires (since an interviewer can clarify doubts about questions and it is possible to extract more detailed data), but they have two clear disadvantages: the time required to collect the same number of answers and the less diverse population that it is possible to reach. Therefore, we decided to use an exploratory survey questionnaire (more specifically an

online questionnaire) in order to collect more data and obtain answers from a wider scope of countries, companies, project types, etc. (Dillman et al. 2014). We know that questionnaires of this kind are usually rigid and the choices proposed in the answers to questions may influence the results obtained. Therefore, knowing these intrinsic problems, we tried to design the questionnaire to minimize their influence on the results.

Survey design. Surveys collect qualitative and quantitative information to provide a snapshot of the current status related to a phenomenon (Wohlin et al. 2012). To ensure rigor and repeatability and to reduce researcher bias, we designed the survey protocol following the template proposed for evidence-based software engineering⁷. It included 33 questions structured into eight sections, which can be publicly accessed⁸. The questions were chosen with the goal of covering the three RQs (see the relationship between survey sections and RQs in Table 5). We tried to cover all of the possible answers in multiple-choice questions in order not to influence the results, always allowing the respondent to state alternative choices that were not explicitly offered. In order to cover the most frequent possible answers, we collected them from the main books and publications on RE (e.g., Pohl 2010, Hull 2011).

All the questions in the survey referred either to the RE experience that participants had or to their beliefs according to their RE experience. Sections 2, 3, 4, and 5 of the survey had questions that were related to the participants' background or to general RE practices used in the participants' daily work (such as elicitation techniques or problems faced during elicitation). These questions allowed the sample of respondents to be characterized and allowed interesting observations to be made when they were related to the answers of the questions in Section 6, 7, and 8, which were directly related to requirements reuse.

Table 5: Relation between survey sections and RQs

Section		Topic	Relation to RQ
Id	Title		
1	Welcome page	Explains the purpose of the survey, who will analyze the results, and how they will be communicated	---
2	Context and work experience	Gathers personal and experience data relevant to the analysis of the survey results	RQ1, RQ2, RQ3
3	RE practices	Surveys general aspects of RE practices followed by the participants	RQ1, RQ2, RQ3
4	RE problems	Includes questions related to the RE problems encountered by the participants in their professional work	RQ1, RQ2, RQ3
5	Observations on requirements	Presents questions about difficulties found in some specific types of requirements derived from the ISO/IEC-25010 quality standard	RQ1, RQ2, RQ3
6	Reuse during RE	Elicits current practices of participants on requirements reuse	RQ1
7	Reuse through patterns	Surveys participants' opinions about the benefits and barriers of using patterns as a requirements reuse technique	RQ3
8	Barriers to reuse adoption	Explores the participants' opinions on the failure to implement reuse practices in RE	RQ2

Protocol. We piloted the survey questionnaire at REFSQ 2013 (April 2013), where it was presented as part of the Empirical Track⁹. The conference attendees were encouraged to respond during the conference. During this pilot of the survey questionnaire, we observed a high percentage of participants that abandoned

⁷ <http://community.dur.ac.uk/ebse/resources/templates/SurveyTemplate.pdf>

⁸ <http://www.upc.edu/gessi/PABRE/SurveyQuestionsJournal.pdf>

⁹ <http://refsq.org/2013/empirical-track/>

the questionnaire when they were answering Section 5. As a consequence, we implemented some changes in this section. They were changes in the presentation of the survey, not in the survey questions. For instance, during the pilot proces we had a table at the start of Section 5 containing the definitions of the different types of requirements; one of the changes was to delete that table and incorporate the definitions as tooltips wherever a specific type of requirement was mentioned. As a result, the percentage of participants that abandoned the questionnaire without completing it, decreased considerably. The questionnaire was available from April 2013 to July 2014.

Channel. The survey questionnaire was implemented using LimeSurvey¹⁰, which offers support for developing online questionnaires and collecting and managing their data.

Population, sampling frame, and sample. The theoretical population for the survey was IT professionals with industrial experience in RE. Finding a suitable sampling frame (i.e., the actual population) is very difficult in surveys for which no exhaustive register of the target population exists (Dillman et al. 2014). Thanks to the Internet, we had access to groups that would be difficult, if not impossible, to reach through other channels (Wright 2005) as requirements engineers that belonged to LinkedIn RE groups (*Requirements Engineering, Requirements Engineering Specialists, Reuse of Requirements Engineering, Requirements Management and Analysis*) and readers of online RE magazines and communities (the IREB magazine, the Spanish Software Quality Community). In addition to the participants contacted through the Internet, other participants were encouraged to participate during their attendance at RE-related conferences thanks to publicity through papers, posters, and demos (REFSQ 2013, RE 2013, REFSQ 2014), tutorials at international conferences (RCIS 2013, ICSE 2013, ICSE 2014), and seminars at two universities (UFES, Brazil, July 2013; U. Oulu, Finland, March 2014), which were taught by one of the authors of this paper. To boost participation, we proactively reached out to the population: we started new discussions every once in a while to disseminate the survey on the LinkedIn groups, as recommended by Dillman et al. (2104); we sent reminder messages to tutorial attendees after the conferences and we resent e-mails to RE communities.

In order to avoid bias, and keeping in mind that the composition of the surveying frame could include people that did not fit the population, we included a question in the first section of the survey to discard such people. Finally, we obtained 77 valid responses from 142 respondents who started to answer the survey. Of the 77 valid responses, 71 of them belonged to the population that we wanted to reach, and the other six were discarded since they correspond to academic RE researchers with no industrial experience in RE.

The potential number of requirements engineers that the survey announcement effectively reached is impossible to say. This problem happens, currently, in all surveys announced through the Internet (e.g., it is currently impossible to know how many people read our posts in the LinkedIn groups). Because of that, it is not possible to truly know the percentage of responses from people to which the survey announcement arrived, but just the percentage of responses from people who opened the survey. This situation also arises in other papers (e.g. Chen et al. 2015, Dietrich et al. 2015) that present surveys open through Internet channels and are only able to give the same participation data.

Data analysis. To ensure the quality of the data obtained from the questionnaire, we applied sanity checks to find obvious errors in the data. We used descriptive statistics to analyze the data (Kitchenham and Pfleeger 2003). We performed content analysis of the free text answers (Krippendorff 2012); we

¹⁰ <http://www.limesurvey.org/>

coded these answers into categories, then we classified them and analyzed their frequencies. Finally, we carried out cluster analyses (Matloff 2009) to find relationships among the results by running these tests over each pair of questions made in the survey. We only present in this paper those correlations and cluster analyses that provided significant conclusions.

4 Characterization of the Respondents

In this section, we describe the 71 responses of the survey regarding certain aspects related to the respondents' background and experience.

Industrial experience as requirements engineer. Figure 1 shows the distribution of the 71 participants according to their level of industrial experience in RE together with their affiliation as industry or academy professionals. The majority of them were industry professionals with RE experience (40 participants; 56.34%). The rest were researchers who declared some degree of industrial experience in RE (15 participants declared significant industrial experience and 16 some; 21.13% and 22.54% respectively). It is worth noting that the total of participants with significant industrial experience in RE represented more than 75% of the respondents.

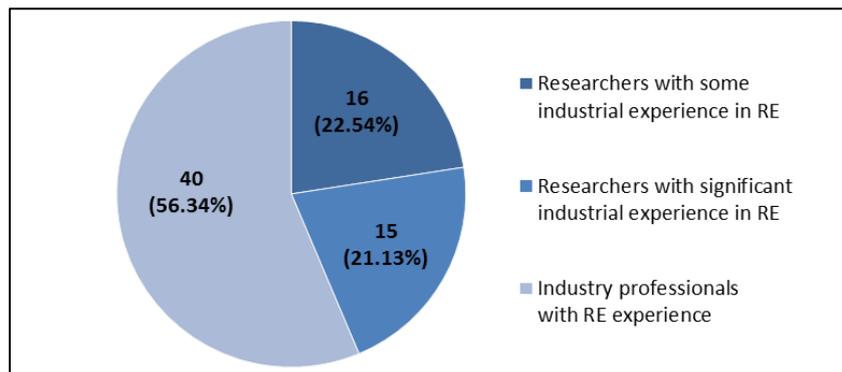


Figure 1: Distribution of responses for the level of industrial experience as requirements engineer

Worldwide distribution. Since the survey was conducted online, requirements engineers from all over the world were able to participate. Figure 2 shows the allocation of the participants on the world map, with responses coming from 27 countries from all continents, with a special focus on Europe (31 participants; 43.66%) and North America (18 participants; 25.35%).



Figure 2: Global view of survey participant locations

Educational background. Figure 3 shows the distribution of participants by their educational background. More than 75% of the participants had a *MSc* or even a *PhD in Science*. Of the 6 participants that selected the option *Other*, 4 had education in business analysis, while the other 2 did not explicitly stated their level of studies. Most MSc or BSc participants knew about the survey in LinkedIn (18 MSc participants out of 41, 10 BSc participants of a total of 11). Instead, most PhD participants knew about the survey in their attendance to REFSQ or after being contacted in conferences (9 participants out of 13 PhD participants attended REFSQ or were contacted in conferences). Taking into account the demographic data of internet users that use LinkedIn published by the Pew Research Centers (Duggan 2015), it is not surprising the high share of participants with an MSc or BSc.

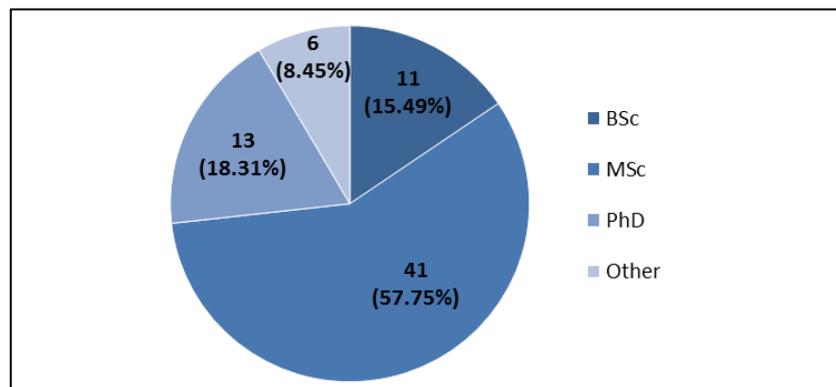


Figure 3: Distribution of responses for educational background

Years of experience. As Figure 4 shows, most of the survey participants had more than 5 years of experience (62; 87.32%); specifically, 31 of them (43.66%) had more than 15 years of experience.

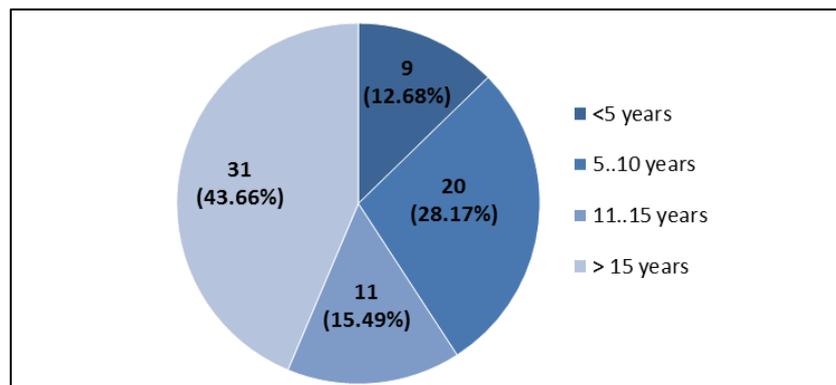


Figure 4: Distribution of responses for years of experience

Organization size. Figure 5 contains the distribution of the respondents according to the size of the organizations in which they acquired industrial experience related to RE. The participants were allowed to select more than one size of organization (i.e. multiple answer question). As can be observed, the percentages in all the categories are quite similar (except for the companies with less than 10 employees), assuring a good coverage of all the possible organization sizes.

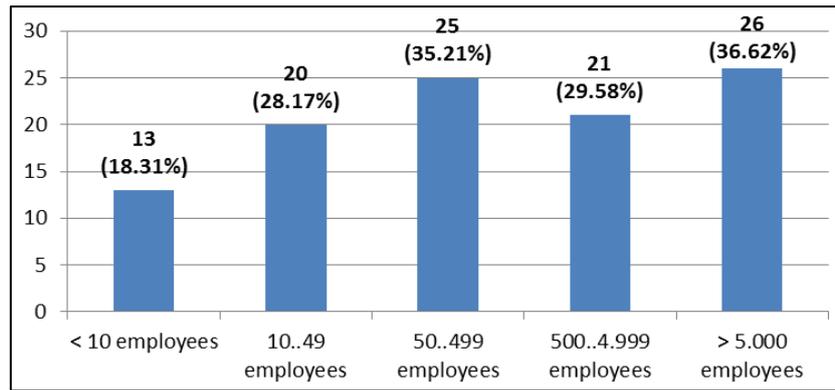


Figure 5: Distribution of responses for organization size (multiple-answer question)

Organization sector. The sectors of the organizations in which participants acquired industrial experience related to RE (Table 6) were presented as a multiple-answer question. The initial list of domains was based on Neill and Laplante (2003) and refined based on our own experience. The most common sectors were: *Consulting* (25; 35.21%), *IT Provider* (18; 25.35%), *Telecommunication* (16; 22.54%), and *Embedded Systems* (13; 18.31%). All of the other sectors (except *Human Resources*) were represented in the survey but were selected by less than 10 participants.

Table 6: Distribution of responses for project domain (multiple-answer question)

Domain	# Respondents	Percentage
Consulting	25	35.21 %
IT Provider	18	25.35 %
Telecommunication	16	22.54 %
Embedded Systems	13	18.31 %
Manufacturing	9	12.68 %
Education	8	11.27 %
Healthcare	7	9.86 %
Insurance	7	9.86 %
Public administration	7	9.86 %
Transportation	7	9.86 %
E-commerce	6	8.45 %
Finance	6	8.45 %
Automotive	5	7.04 %
Customer relationship management	4	5.63 %
Travel	2	2.82 %
Power distribution	1	1.41 %
Human resources	0	0.00 %

Languages to specify requirements. A multiple-answer question was used to determine the languages used to specify requirements (Figure 6). Pohl (2010) was used as the main source of choices offered to the participants and, based on our experience and knowledge, we selected those values which we considered to be the most common. According to the results, the largest share of responses (57; 80.28%) used

requirements in *Natural Language*, closely followed by *Use Cases or other scenario-based approaches* (55; 77.46%), and *UML* (38; 53.52%). It is important to note that of the 12 respondents (16.90%) that selected the option *Other*, 4 used BPMN to write requirements.

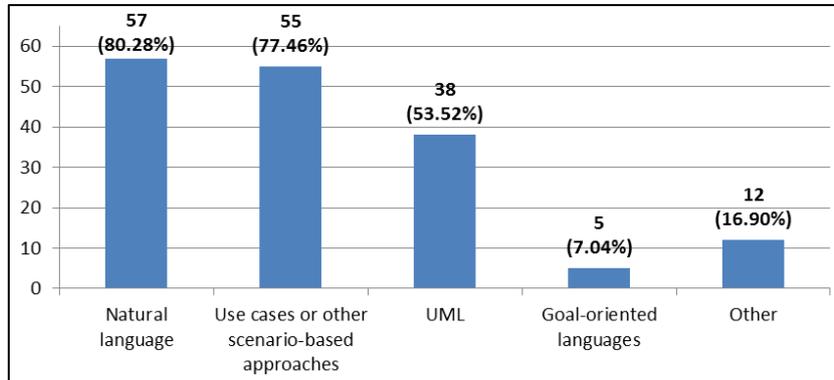


Figure 6: Distribution of responses for languages used to specify requirements (multiple-answer question)

Requirements elicitation methods. A multiple-answer question was used to determine the methods used to elicit requirements (Figure 7). We used the elicitation techniques presented in Pohl (2010) and Hull (2011) as sources for the proposed answers. Again, based on our experience, we selected the most common ones for the list presented to the user. The results showed that 59 participants (83.10%) used *Interviews*, 50 used *Workshops* (70.42%), 38 used *Questionnaires* (53.52%), 37 used *Observations* (52.11%), 29 used *Focus Groups* (40.85%), and 22 used *Perspective-Based Reading* (30.99%). Other elicitation methods used by 10 of the participants (14.08%) included business-form analysis, prototyping, and their own patented methods. Four participants (5.63%) *Never or rarely used an elicitation method*.

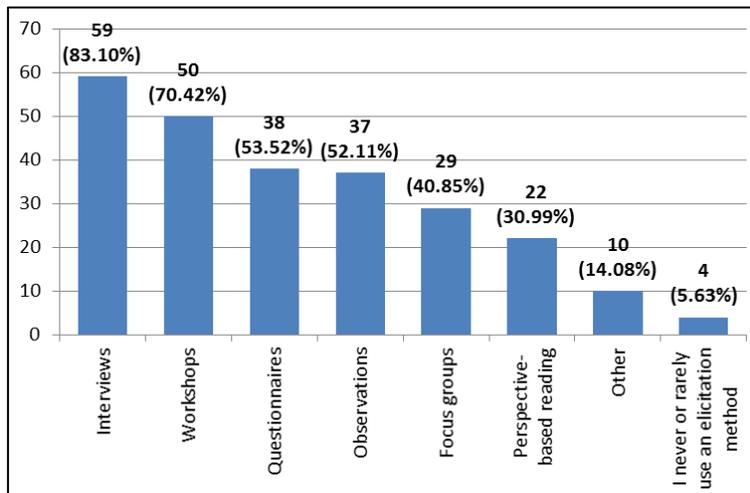


Figure 7: Distribution of responses for requirements elicitation methods (multiple-answer question)

5 RQ1: On Requirements Reuse Adoption

In order to answer RQ1, we asked the participants about: the level of requirements reuse they had in their projects, the techniques they implemented to achieve requirements reuse, and the types of requirements that were the most similar among their projects.

Current state of requirements reuse. We required to measure the level of requirements reuse in the participant projects using a Likert Scale ranging from *1-Nonexistent or Very Low* to *5-Very High*. The results show (Figure 8) that a majority of participants (78.87%) stated some kind of requirements reuse (i.e., the level was declared equal to or greater than *2-Low*). However, reuse does not seem to be an established practice in IT projects since only 18 of the participants (25.35%) marked it as equal to or greater than *4-High*.

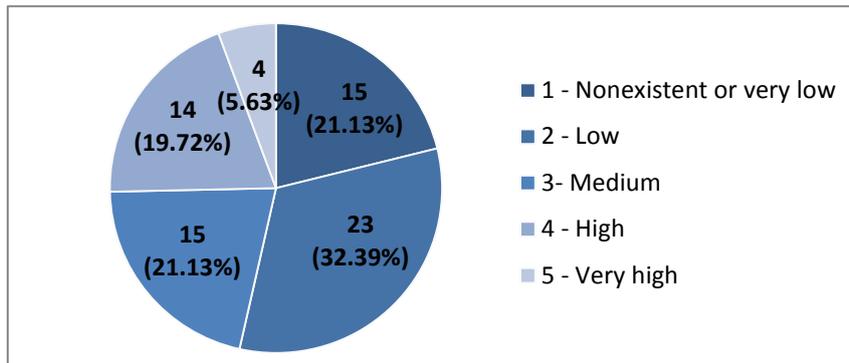


Figure 8: Distribution of responses for requirements reuse level

Table 7 contains the cross-tabulation between the requirements reuse level stated by the participants and their organization size. Most of the Chi-Square exact tests led to p-values that were smaller than 0.05, which means that there is a statistically significant relationship between these two variables. Specifically, the results in Table 7 show that there is a trend towards a higher level of requirements reuse level the larger the organization is. It is worth noting that this correlation is not due to the domain of the projects carried out by the organizations since these sectors do not show any correlation with the organization size or with the requirements reuse level (i.e., a size or level represented a variety of sectors and not a single one).

Table 7: Cross-tabulation of requirements reuse level and organization size

		Organization Size (#employees)				
		<10	10..49	50..499	500..4.999	>5.000
Reqs. Reuse Level	Nonexistent or very low	7 (9.86%)	7 (9.86%)	4 (5.63%)	0 (0.00%)	4 (5.63%)
	Low	4 (5.63%)	8 (11.27%)	14 (19.72%)	8 (11.27%)	5 (7.04%)
	Medium	2 (2.82%)	3 (4.23%)	3 (4.23%)	9 (12.68%)	5 (7.04%)
	High	0 (0.00%)	2 (2.82%)	3 (4.23%)	1 (1.41%)	10 (14.08%)
	Very high	0 (0.00%)	0 (0.00%)	1 (1.41%)	3 (4.23%)	2 (2.82%)
TOTAL		13 (18.31%)	20 (28.17%)	25 (35.21%)	21 (29.58%)	26 (36.62%)
Chi-Square (p-value)		0.015	0.169	0.041	0.001	0.033

We did not find any other significant relationship between any other variable and the requirements reuse level except the one related to requirements reuse techniques (see the next point in this section). Remarkably, we did not find any statistically significant relationship that one might have thought about. Among them, we highlight that we could not establish a relationship among the requirements reuse level and years of experience of the respondent.

Requirements reuse techniques. Using a multiple-answer question, the participants were provided with a list of requirements reuse techniques; as usual, they had the possibility to add any missing value with an open field value option. This question was only asked to those participants that implemented some degree of requirements reuse in their projects, i.e., those participants stating the requirements reuse level in the first question as being greater than or equal to 2-Low (56 participants; 78.87%).

The results for this question are shown in Figure 9. The most common techniques (35 participants; 62.50%) are those based on the textual copy and subsequent modification of requirements from previous projects (also known as *clone* and *own reuse* by Eriksson et al. 2009, Bosch 2000). More specifically, we are referring to the answers: *Copy and paste of groups of requirements*, *Copy and paste of individual existing requirements*, and *Duplicate of a full existing requirements specification* and work in its parts as needed. Less common techniques were *Fill in predefined templates* and the *Use of a requirement patterns catalogue*; this last technique was the least used (only 6 participants; 10.71%). We also found that for 20 participants (35.71%) the reuse technique used could be different depending on the project.

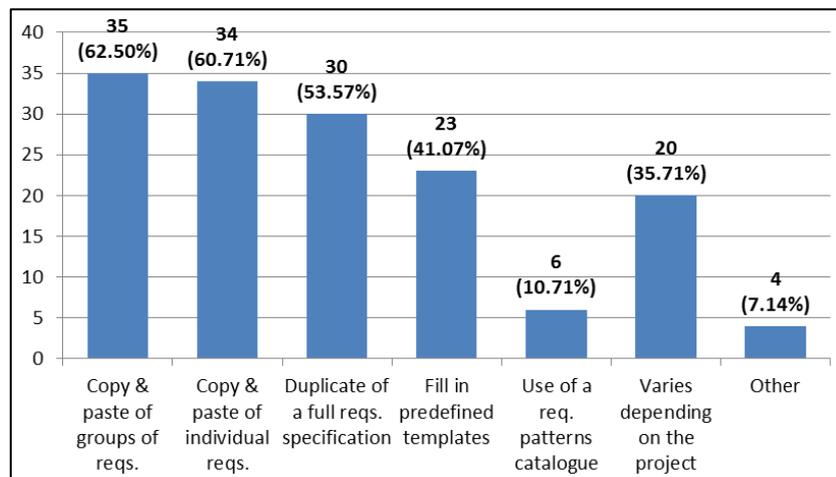


Figure 9: Distribution of responses for requirements reuse techniques (multiple-answer question)

At a first glance, the number of participants that stated to use patterns could seem too large since it seems a technique closer to research than to industry. However, it is worth to remark that in the survey we stated before this question that "A Software Requirement Pattern consists of natural language templates for generating those requirements that are related to a specific objective (goal), as well as some information to identify its adequacy to a particular project and how it may be tailored to the project". Thus, the existence of a set of reusable natural language requirement templates or the existence of a reusable document with natural language requirements could have induced some respondent to classify their approach as pattern-based. We recognize that another possible justification is that the title of the survey, which was

‘requirements reuse and patterns’, may have attracted more people interested in reuse; consequently, we add this fact to the threats to validity in Section 9.

An in-depth look into the relationship between the reuse level and the reuse techniques (Table 8) shows that there is a statistically significant relationship between these two variables (Chi-Square exact test). The techniques for which this relationship could not be stated is for *Use of a Requirement Patterns Catalogue* and *Other*; the reason is that there were not enough data points for these techniques to make the results of the tests reliable. Even when this is taken into account, we could observe the trend that the more elaborated reuse techniques were (*Fill-in predefined templates*, *Use of a requirement patterns catalogue*), the higher the reuse level was, while simpler reuse techniques (*Copy & paste of groups of requirements*, *Copy & paste of individual requirements*, *Duplicate of a full requirements specification*) were mostly used in organizations with low and medium reuse levels. Finally, it is important to note that the respondents that participated in projects with lower reuse level declare that they used different requirements reuse techniques depending on the project, indicating that reuse techniques seem to be more established in higher reuse levels.

Table 8: Cross-tabulation of requirements reuse level and requirements reuse techniques

		Requirements Reuse Techniques						
		Copy & Paste of groups of requirements	Copy & Paste of individual requirements	Duplicate of a full reqt. specification	Fill in predefined templates	Use of a reqt. patterns catalogue	Varies depending on the project	Other
Requirements Reuse Level	Low	14 (25.00%)	16 (28.57%)	13 (23.21%)	2 (3.57%)	0 (0.00%)	11 (19.64%)	1 (1.79%)
	Medium	11 (19.64%)	7 (12.50%)	10 (17.86%)	6 (10.71%)	1 (1.79%)	5 (8.93%)	0 (0.00%)
	High	8 (14.29%)	8 (14.29%)	5 (8.93%)	12 (21.43%)	3 (5.36%)	3 (5.36%)	1 (1.79%)
	Very high	2 (3.57%)	3 (5.36%)	1 (1.79%)	3 (5.36%)	2 (3.57%)	1 (1.79%)	2 (3.57%)
TOTAL		35 (62.50%)	34 (60.71%)	29 (51.79%)	23 (41.07%)	6 (10.71%)	20 (35.71%)	4 (7.14%)
Chi-Square (p-value)		0.029	0.015	0.008	0.014	0.34	0.011	0.57

From all the other correlation analyses that were carried out over the results, we found another interesting fact related to the languages that are used to specify requirements and the techniques that are used to reuse requirements. The results highlight a strong statistical relationship (Chi-Square test p-value = 0.018) among the respondents that use natural language to specify requirements and the duplication of requirements specifications for reusing requirements. This indicates that the respondents that use natural language to specify requirements acquired the duplication of specifications as main reuse technique in more than half of the cases, in contrast to other notations, which use duplication of specifications only in 32% in average.

Types of requirements likely to be reused. In the survey, we included questions to ask about the similarity between requirements of the same type in different projects (based on the respondents’ experience). These questions used the Likert Scale, ranging from *1-Totally Agree* to *5-Totally Disagree*.

The requirement types proposed were based on the characteristics of the quality models proposed in the ISO/IEC-25010 and in the extended version Carvallo et al. (2006) of its predecessor (i.e. the ISO/IEC 9126-1 quality standard). The respondents were allowed to add other requirement types that might be relevant and that were not included in the list provided in the survey. The types that were ranked with a higher reuse rate were: *Reliability*, *Maintainability*, *Usability*, and *Security* (see Table 9). For the other types of requirements, the results do not highlight any significant difference in the level of recurrence, with most of them being around 3 (equivalent to Neutral value). Based on the classification proposed in (Franch et al. 2013a, Palomares et al. 2013, Renault et al. 2009) of these requirement types as Functional Requirements (FRs), Non-Functional Requirements (NFRs), or Non-Technical Requirements¹¹ (NTRs), it is important to note that the requirements that are more likely to be reused are NFRs.

Table 9: Average response for requirement types likely to be more similar between projects (1 – Totally agree, 5 – Totally disagree)

Classification	Requirement Types	Likert Scale Average ¹²
NFR	Reliability	1.75
NFR	Maintainability	1.93
NFR	Usability	2.17
NFR	Security	2.35
NFR	Performance efficiency	2.75
NTR	Business suitability	2.85
NTR	Project suitability	2.86
NFR	Compatibility	2.93
NFR	Portability	2.94
FR	Functional suitability ¹³	2.96
NTR	Product non-technical suitability	3.11
NTR	Supplier suitability	3.13

6 RQ2: On Barriers to the Adoption of Requirements Reuse

In order to answer RQ2, we asked for the opinion of those participants who declared a level of reuse as nonexistent or very low (15 participants; 21.13%) about two aspects: the possible problems in requirements reuse proposals that prevent them from adopting reuse in organizations; and what is missing in requirements reuse proposals made by researchers that would facilitate the incorporation of reuse in industry.

Problems in requirements reuse proposals that hinder their adoption by organizations. The participants were provided with a list of problems for which they could select one or more options. In that list, we decided to include the option *Organizations consider the incorporation of requirements reuse to be too complex*. Although it may seem a trivial answer, we knew from our informal pilot studies that sometimes this is the only reason preventing reuse. Besides the list provided, participants also had the possibility to add

¹¹ Non-Technical Requirements are the ones not related to technical aspects of the software but to contextual aspects as prices, licenses or suppliers.

¹² See *Face Validity* paragraph in Section 9.2 for a discussion on calculating averages over Likert Scale variables.

¹³ *Functional Suitability* refers to the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions

any missing value with an open value option. Figure 10 shows that the most common opinion was that *Organizations do not know how to do this incorporation* (14 participants; 93.30%). Three other issues that were considered relevant for almost half of the respondents were the following: *Even if their incorporation may provide benefits, the initial investment is too high* (8 participants; 53.33%); *Organizations never thought about incorporating requirements reuse proposals* (7 participants; 46.67%); and the opinion that *Organizations consider the incorporation of requirements reuse to be too complex* (7 participants; 46.67%). In summary, it can be concluded that the reasons are based on the ignorance about reuse or reuse elicitation processes and on doubts with regard to its return-on-investment.

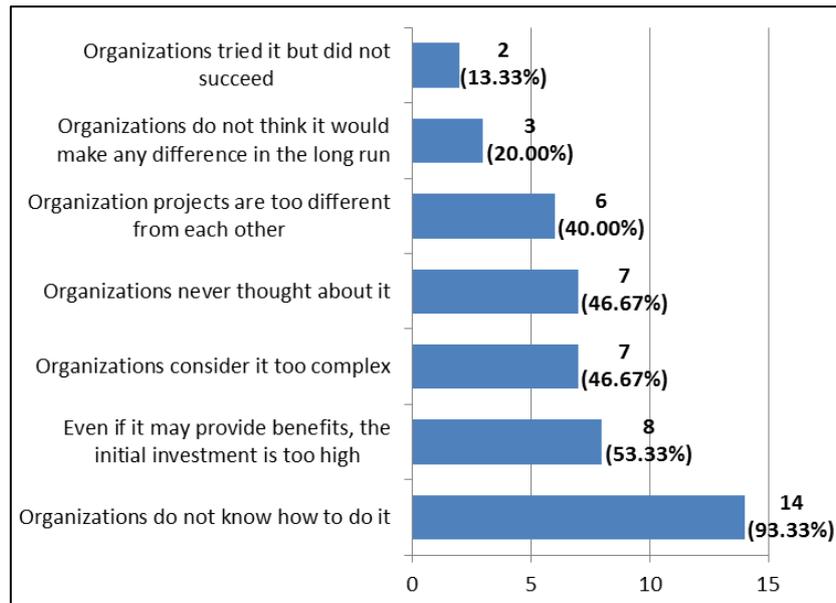


Figure 10: Distribution of responses for problems in requirements reuse proposals (multiple-answer question)

What is missing in researchers’ requirements reuse proposals to be adopted by practitioners. Given that this issue was asked with an optional free text question, we only got 4 answers to it. Despite this, the respondents made some good points that are worth discussing. On the one hand, two respondents agreed on the fact that what is missing in requirements reuse proposals is a solid business case behind them that can convince a CIO to make the investment necessary to incorporate them into the RE process: “[*What is missing is*] *Presenting successful cases on the existing requirements reuse proposals for new requirements reuse clients*” and “*A solid business case is needed for requirements reuse to be widely used. Since there is a lack of solid business cases, another approach that could be used is Technology Maturation. However, there are few companies that are big enough, with deep pockets, and the required imagination to bankroll technology maturation for requirements reuse*”. On the other hand, the other two respondents declared that the reason for not incorporating requirements reuse in industry is the lack of process maturity in organizations: “*Nothing [is missing], apart from the maturity of the organization*” and “*With [my] limited knowledge, I do not believe anything is missing; however, the maturity of the company may be the reason.*”

7 RQ3: On Requirements Reuse Through Patterns

To know the benefits and drawbacks of using a catalogue of software requirement patterns to elicit requirements, in Section 8 of the questionnaire, we provided the 71 participants with a short explanation of what an SRP and an SRPs catalogue are. We then asked the participants their opinion about a list of common RE problems that could be mitigated by using an SRPs catalogue and two lists of critical factors and barriers that could influence its successful adoption. In the three lists, the participants had the opportunity to add new items that might be missing. We extracted the values of the first list by looking at the IEEE Recommended Practice for Software Requirements Specifications (2009) and Pohl (2010). To state the values in the second and third lists, we used our own experience and knowledge as well as the values obtained in Hoffman et al.'s survey (2013) and relevant general requirements reuse barriers and success factors stated by Wiegers and Beauty (2014) in their book.

Table 10: Average response for problems mitigated by the use of an SRPs catalogue (1 – A lot, 3 – At all)

Problems mitigated by the use of an SRPs catalogue	Likert Scale Average ¹¹ (65 participants NOT using SRPs)	Likert Scale Average ¹¹ (6 participants using SRPs)	Likert Scale Average ¹¹ (all participants)
Incompleteness of requirements specification	1.59	1.66	1.60
Lack of requirements uniformity	1.64	1.66	1.64
Requirements inconsistency	1.76	1.66	1.75
Requirements ambiguity	1.80	1.66	1.79
Lack of requirements quantification	1.86	1.83	1.86
Stakeholders do not know their needs exactly	1.88	1.83	1.88
Too little time invested in requirements elicitation	1.89	2.17	1.91
Requirements non-verifiable	1.90	2.17	1.92
Too much time spent in requirements elicitation	1.93	1.83	1.92
Lack of requirements traceability	1.95	2.33	1.98
Stakeholders' needs change during the requirements elicitation process	2.00	2.33	2.03
Lack of requirements prioritization	2.08	2.33	2.10
Conflicts among needs stated by stakeholders	2.11	2.33	2.13
<i>Lack of requirements relationships (dependencies) ¹⁴</i>	<i>1.00 (3)</i>	---	<i>1.00 (3)</i>
<i>Efficiency of the requirements elicitation process ¹³</i>	<i>1.00 (5)</i>	---	<i>1.00 (5)</i>
<i>Accessibility of RE to small and medium sized enterprises ¹³</i>	<i>1.00 (3)</i>	---	<i>1.00 (3)</i>
<i>Change of stakeholders' needs during the requirements elicitation process ¹³</i>	---	<i>1.67(3)</i>	<i>1.67 (3)</i>

Problems mitigated by the use of an SRPs catalogue. The results in Table 10 show that the four problems that could be most mitigated are the following: *Incompleteness of requirements specification*, *Lack of requirements uniformity*, *Inconsistency of requirements*, and *Ambiguity of requirements*. The main differences among participants not using SRPs (65 participants) and the ones using SRPs (6 participants) is that the second group considered that SRPs could help them not *Spend too much time in requirements*

¹⁴ Further problems stated by participants to be mitigated by SRPs (in brackets, number of participants that stated them).

elicitation. The respondents added problems that were missing on the list, the most common ones being: *Lack of requirements relationships (dependencies)*, *Efficiency of the requirements elicitation process*, and *Accessibility of RE to small and medium sized enterprises*. For those individuals who had used SRPs, *The change of stakeholders' needs during the requirements elicitation process* was also categorized as a problem that is likely to be mitigated.

Critical factors for the successful adoption of an SRPs catalogue. *The existence of a well-defined method for using SRPs* as well as *The existence of tool support* were considered to be the most critical factors for the introduction of SRPs by all types of respondents (see results in Table 11). Remarkably, the respondents who had used SRPs gave more relevance to *The existence of a community of users supporting SRPs*, ranking it in third position. Finally, both groups agreed that *The existence of a help desk* was the least significant critical factor.

Other critical factors that were not included in the list but were considered as being very important by the participants were: *The existence of a ready-to-use SRPs catalogue*, *The existence of a person or department inside the organization expert on SRPs*, *The existence of successful cases using SRPs*, and *The possibility of having free trial periods*.

Table 11: Average response for critical factors influencing the adoption of an SRPs catalogue (1 – Totally agree, 5 – Totally disagree)

Critical factors influencing the adoption of an SRPs catalogue	Likert Scale Average ¹¹ (65 participants NOT using SRPs)	Likert Scale Average ¹¹ (6 participants using SRPs)	Likert Scale Average ¹¹ (all participants)
Well-defined reuse method	1.52	1.33	1.50
Tool support	1.65	1.66	1.65
Training courses	2.09	2.33	2.11
Existence of a community of users	2.22	2.00	2.20
Help desk	2.69	2.50	2.67
<i>Ready-to-use SRPs catalogue</i> ¹⁵	1.00 (3)	---	1.00 (3)
<i>Person or department inside the organization expert on SRPs</i> ¹⁴	1.50 (2)	1.00 (2)	1.25 (4)
<i>Successful cases using SRPs</i> ¹⁴	1.50 (3)	---	1.50 (3)
<i>Free trial periods</i> ¹⁴	2.00 (2)	---	2.00 (2)

Barriers for the successful adoption of an SRPs catalogue. For the respondents not using SRPs, only two items from the list of barriers to the adoption of SRPs (Table 12) were considered important: *The resistance of requirements engineers to change*, and *The integration of the catalogue with the existing requirements engineering process*. For the respondents that had used SRPs, the respondents reinforced the general conclusion that the most important barrier is: *The resistance of requirements engineers to change*. However, differences arose regarding the rest of barriers. *The risk of converting the requirements elicitation into a stiff process* was more important for the respondents that had used SRPs (in the second

¹⁵ Further critical factors stated by participants for adopting an SRPs catalogue (in brackets, number of participants that stated them)

position for these respondents, but on the fourth one for the respondents not using SRPs). For *The amount of reusable knowledge to create and maintain*, the participants that had used SRPs totally disagreed with this being a barrier, as opposed to the rest of respondents, which ranked it in the third position with an average of 2.41.

It is important to point out the statistical relationship that exists among the consideration of *The existence of tool support* as a critical factor for adopting SRPs and the consideration as barriers of *The integration of the catalogue with existent RE processes* and *The amount of reusable knowledge to create and maintain*. For the first barrier, the Chi-Square test gave a p-value of 0.002 and 41 participants (57.75%) agreed on both statements. For the second barrier, the p-value was 0.007 and 34 participants (47.89%) agreed on both aspects.

We found another statistical relationship among those responses believing that *The existence of a well-defined method* for using SRPs is a critical factor for introducing SRPs, and that also believed that there is a *Risk of converting requirements elicitation into a stiff process*. The p-value of the Chi-Square test was 0.047 and 40 participants (56.34%) agreed in both statements.

The participants added other barriers. The most common ones were: *The lack of management support* and *The difficulty of adapting SRPs output to the organization requirements specification format*.

Table 12: Average response for barriers influencing the adoption of an SRPs catalogue (1 – Totally agree, 5 – Totally disagree)

Barriers influencing the adoption of an SRPs catalogue	Likert Scale Average ¹¹ (65 participants NOT using SRPs)	Likert Scale Average ¹¹ (6 participants using SRPs)	Likert Scale Average ¹¹ (all participants)
Resistance to change of req. engineers	1.92	1.5	1.88
Integration of the catalogue with the existing req. engineering processes	2.03	2.00	2.03
Amount of reusable knowledge necessary to create and maintain	2.41	4.16	2.56
Risk of converting requirements elicitation into a stiff process	2.42	1.83	2.37
<i>Lack of management support</i> ¹⁶	1.00 (6)	1.00 (2)	1.00 (8)
<i>Difficulty of adapting SRPs output to the organization requirements specification format</i> ¹⁵	1.00 (2)	---	1.00 (2)

8 Discussion

In this section, we present the observations related to each RQ derived from the analysis of the survey results. Each observation is discussed and compared with related work.

8.1 On the Current State of the Practice of Requirements Reuse

With regard to the level of requirements reuse, requirements reuse techniques and types of requirements more prone to reuse, we derived the following observations:

¹⁶ Further barriers stated by participants for adopting an SRPs catalogue (in brackets, number of participants that stated them)

A significant percentage of the respondents practice requirements reuse. A high percentage of participants declared some level of reuse in their projects (79%). We think that this is a significant observation and it aligns well with previous surveys. In other surveys on RE practices, the average percentage of participants stating requirements reuse as a regular practice (i.e., always or widely used) ranges between 40% and 82% (see Table 3). In surveys focused on requirements reuse, in Chernak's (2012) survey 59% of the participants stated having used requirements reuse in the last projects they participated. This percentage is lower than in our survey (79%), and also in Bakar and Kasirum's (2014) survey, where the percentage is 72.2%. We cannot compare the results with Hoffman et al.'s (2013) interviews because their study does not include results on the state of the practice but on opinions of requirements engineers.

The level of requirements reuse is usually low. As a nuance to the previous observation, even if a majority of participants declared that they had adopted some reuse practices, they do not apply them in every single project or they do not apply them intensively. From the 79% of the participants who stated some level of requirements reuse, only 25% of them qualified such reuse as equal to or greater than high (see Figure 8). This could be explained by the lack of integration of reuse practices in the requirements engineering process, which makes it an ad hoc technique.

Chernack (2012) does not include questions that could illustrate this observation. However, Bakar and Kasirum's (2014) survey found that from the 72.2% participants that apply requirements reuse, 52.8% do not have a systematic reuse approach and they do reuse in an ad hoc manner. Even if the percentages are not exactly the same as those in our survey, they show some similarities that could hint at some trend in the way requirements engineers adopt reuse in their work.

Participants of larger organizations declare a higher level of reuse. Higher reuse levels in larger organizations (see Table 7) could be explained by a higher number of IT projects with similar characteristics in this kind of organization that would make the recurrence of requirements in subsequent projects more likely to appear. In particular, functional requirements can be expected to appear recurrently in projects that target the same domain, while non-functional and non-technical requirements may be in addition influenced by other characteristics, e.g., type of system (service-oriented architecture, ERP system, etc.) for non-functional requirements, and type of customer (e.g., public administration, large company, etc.) for non-technical requirements. Another explanation could be that larger organizations tend to have more established processes (see below).

It is not possible to compare these results with other surveys. The only paper that explores a similar relationship is Chernak's (2012), which observes that the level of adoption of reuse differs depending on the size of the project team, but it does not explore the relationship with the size of the organization.

Requirements reuse techniques most commonly used are those based on textual copy and subsequent modification of requirements from previous projects. The most common techniques (chosen by more than a half of the participants) were those based on textual copy and subsequent modification of requirements from previous projects (see Figure 9); in particular, participants using natural language to specify requirements adopted the duplication of specifications as the main reuse technique in more than half of the cases.

This observation exemplifies the gap between research and industry in this area, since as mentioned in Subsection 2.1, 47% of scientific proposals centered in patterns in the period 2010–2015, but this percentage does not match what happens in industry. This gap is also reflected by the fact that only 4% of proposals on requirement patterns have been validated in industry (Subsection 2.2).

The surveys on requirements reuse (Bakar and Kasirun 2014, Chernak 2012) did not ask about techniques or level of abstraction of the knowledge to reuse, but only about the artifacts or languages used to specify requirements. We think the reason is that they assume reuse to be a simple copy and modification of knowledge to reuse without considering having levels of abstraction in this knowledge.

There is a correlation between the level of requirements reuse and the requirements reuse techniques used. The study showed that the participants that used more elaborate reuse techniques were the ones that declared a higher reuse level in their projects (see Table 8). At the same time, it can be observed that a low level of reuse was significantly related to small companies (see Table 7). From these facts it may be inferred that the participants that stated a low level of reuse (32%), and less elaborate reuse techniques, were probably referring to ad hoc requirements reuse, i.e. not integrated in the requirements process of the company but as a practice followed by one or more employees or by small companies without consolidated development processes. Higher reuse levels lead to elaborate reuse techniques, since the level of reuse induces the definition of methods and processes of reuse.

The fact that larger organizations tend to have better-defined, well-known, and established methods and processes, which is a critical factor for applying reuse, is corroborated by Dyba (2003). As indicated above, other surveys do not ask about techniques with the same meaning as in our survey, and thus we were not able to check this observation in their results.

Organizations with more established software processes and methods are the ones that declare a higher level of requirements reuse. Participants declaring a low or medium reuse level tend to use different requirements reuse techniques in different projects (35.71% of participants). On the contrary, our results showed that reuse techniques seem to be more established in projects of participants that declare a high or very high reuse level in their projects. This point is also supported by answers to questions related to RQ2 (Section 6), where the participants who declared a nonexistent or very low level of reuse stated that the reason was that organizations do not know how to incorporate requirements reuse (i.e. it is supported by the comment of two participants about the lack of process maturity in their organizations).

This conclusion matches the assumption stated in Sommerville's requirements engineering maturity model (Sommerville and Sawyer 1997, Sawyer et al 1997, Niazi et al. 2008) that requirements reuse corresponds to an advanced requirements engineering elicitation technique. Other authors (Nikula et al. 2000, Chernak 2012, Hoffmann et al. 2013) reach the same conclusions about the importance of establishing and adopting well-defined requirements reuse processes in order to apply requirements reuse.

NFRs are more likely to be similar or recurrent among projects. With respect to the type of requirements that are more prone to be reused among projects, NFRs (rows 1 to 5, 8 and 9 in Table 9) were considered as more likely to be reused than FRs (row 10). For NTRs (rows 6, 7, 11, and 12), the results were not the ones we expected based on our own experience. For instance, requirements on *Supplier suitability*, which are defined in the questionnaire as those requirements that state conditions on the company that distributes or implements the software product, were considered to be less recurrent than

FRs. Our interpretation is that NTRs were not well understood by the participants since, according to the RE experts we collaborated with in the construction of the PABRE framework proposal (Franch et al. 2013a), this kind of requirement is in fact quite recurrent. This misunderstanding could be caused by the fact that NTRs are not always included in requirement specifications unless projects are call-for-tender projects.

Existing works (Withall 2007, Shahrokni and Feldt 2010, Supakkul et al. 2010, Hoffman et al. 2012) align with our findings since all of them are software requirements reuse proposals mainly involving requirements that fit the types that are identified as being more likely to be reused (i.e., NFRs). It is also important to remark that, as also supported by the previous works, a big percentage of NFRs and NTRs are domain-independent (i.e., they appear basically in the same way in different requirement specifications, even if they belong to projects from different domains). For FRs, it is quite the opposite: as stated by Lam (1998), it is necessary to identify and formalize the reusable requirements for each functional area. Since FRs are domain-dependent (Filipovikj et al. 2014 for the automotive domain, Li et al. 2012 for seismology applications, Jensen et al. 2009 for healthcare applications, Konrad and Cheng 2002 for embedded systems), it is not surprising that this type of requirement was ranked as less reusable than NFRs and NTRs. However, in the case of related work addressing reuse in companies working on product lines or on product releases, requirements reuse of FRs is high since they address development of software products in the same domain area (Cox et al. 2009, Chernak 2012). Finally, the interviewed requirement engineers in (Hoffman et al. 2013) agree that SRPs would be usable in general for NFRs and for recurrent FRs.

8.2 Reasons Behind the Lack of Adoption of Existing Requirements Reuse Proposals

Most of the participants agreed that the most common barrier for organizations was their ignorance on incorporating a reuse strategy into their current processes (93.3%). Other barriers considered relevant (Figure 10) for almost half of the respondents were the initial investment required, the lack of awareness about the benefits that reuse may bring, and the inherent complexity of implementing reuse. Regarding the drawbacks stated by participants on existing reuse proposals, the ones considered more important are the absence of a solid business case behind the approaches that may convince CIOs and the lack of process maturity in organizations.

Ignorance of reuse techniques and processes is the main reason for the lack of reuse adoption. As a summary, the results show that the main reason for companies not adopting requirements reuse is that organizations do not know how to do it, which implies being ignorant of the techniques and processes behind such reuse. This is corroborated by the main success factors related to the adoption of SRPs obtained as answers to RQ3 (see Subsection 8.3), which show the ignorance of methods to guide the requirements reuse process. Thus, more empirical research on requirements reuse should be carried out in order to transfer requirements reuse techniques and methods to companies and to demonstrate the benefits and return-on-investment that they provide (see RQ2 results in Section 6).

Bakar and Kasirun (2014) and Chernak (2012) also asked in their surveys the reasons for not reusing requirements. Both surveys differ in the answer, being in their cases more based on the knowledge and artifacts to reuse than the reuse process. The reason they identify is the lack of quality and incompleteness of requirements to reuse in requirement repositories, and the Bakar and Kasirun's survey also identified as

a critical factor the lack of convenient tools with suitable requirements classification and good facilities for accessing the requirements repository. We think the reason for the difference in the results of our survey is as follows: In our survey, this question was just answered by participants who declared a low level or no experience of requirements reuse, and in the surveys in the related work the questions were answered by all the participants (having 72.2% considerable experience in requirements reuse). Probably, the second group states the problems they have in requirements reuse application owing to the bad quality of reuse knowledge, and the first one just thinks about the doubts relating to introducing the practice, the cost of this introduction, and the process of applying reuse.

8.3 About Benefits and Drawbacks of Using a Catalogue of Software Requirement Patterns

With regard to the benefits and drawbacks of the use of a catalogue of SRPs, we derived the following observations from the opinions of the survey participants.

Problems mitigated by the use of SRPs are mainly related to the quality of resultant requirement specifications. The four problems that can be mitigated by the use of SRPs (as identified by the survey participants) were related to the quality of requirement specifications (see Table 10): *Incompleteness of requirements specifications*, *Lack of requirements uniformity*, *Requirements inconsistency*, and *Requirements ambiguity*. These enhancements are a logical consequence of working with a knowledge base of reusable artifacts that is supposed to contain artifacts of a certain quality. Considering specifically the answers of participants with experience in SRPs, an additional mitigated problem is to *Spend too much time in requirements elicitation*.

These problems are also identified as benefits in the interviews reported by Hoffmann et al. (2013). The interviewees in that study also point out the *Efficiency of the elicitation process* as a benefit of using requirement patterns. This is because they thought that less time would be spent on the elicitation process if patterns were used. In the industrial applications of pattern proposals reported in Subsection 2.2, three of the four applications effectively observed a decrease in the time dedicated to requirements engineering (Issa et al. 2010, Toval et al. 2002, Pacheco et al. 2015) and two of them (Pacheco et al. 2015, Karatas et al. 2014) observed an improvement in the requirements quality. The other benefit identified by these interviewees is the *Improvement on the requirements traceability*. In this case, the reason is that requirement dependencies would be incorporated in the patterns and propagated to requirement specifications. Regarding traceability, in our survey, the respondents did not see the relationship of patterns with the mitigation of the lack of traceability (their Likert scale average was 1.92 and 1.98, respectively, with 1-Agree a lot and 3-Do not agree at all).

Critical factors and barriers for the successful adoption of an SRPs catalogue are related to the existence of a well-defined reuse method and the people involved. The importance given to the existence of a *Reuse method* and *Tool support* (identified as critical success factors in Table 11) is probably caused by the absence of a well-defined, well-known, and established method to guide the reuse processes undertaken by the participants. It is not surprising that the barriers related to people involvement were considered the most important ones (Table 12) since, in organizational processes, the involvement of personnel is a key factor for the adoption of a reuse technique and its success (Dyba 2005).

The critical factors and barriers for the adoption of SRPs identified in our survey were also identified by Hoffman et al. (2013). Other aspects identified by Hoffman et al. (2013) that are related to the quality of the SRPs catalogue were not included our survey because we took them for granted (we do not consider that an SRP is reusable if it is not of good quality).

9 Threats to Validity

Internet surveys are powerful instruments that make it possible to know the current state of the practice. However, they usually have some weaknesses that threaten the validity of the conclusions drawn from them (Evans and Mathur 2005). In this section, we analyze the threats to the validity of our study based on some of the aspects defined in the literature (Evans and Mathur 2005, Trochim 2006, Wohlin et al. 2012, Dillman et al. 2014).

9.1 Internal Validity

Instrumentation. Instrumentation threats can appear if the experiment has an error in its design or changes are necessary in the middle of the experiment. The first fact that could be considered a threat is the choice of an Internet survey to determine the state of the practice (instead of contacting industrial organizations directly). From the several available empirical research instruments, we opted for online questionnaires as the way to target the widest possible population. In our view, for a state of the practice not based on a literature review, this is a valid instrument to gather knowledge from practitioners.

To avoid instrumentation threats, we also conducted pilots of the questionnaire to ensure its correct understanding and to find possible defects. We also hired a native English speaker to revise the questionnaire. Because of these pilots, we implemented some changes in the interface of a specific section of the survey where a high percentage of the non-completed attempts occurred (for more information see Survey Design in Section 3). Taking into account the results of the pilots, and to avoid the typical design errors in online questionnaire, some of the critical questions were accompanied by a glossary of terms and we added text fields for clarification whenever necessary. This glossary of terms included the description of the types of NFRs and NTRs used in the survey as well as the concepts SRP and SRPs catalogue.

Although not having used quantitative measures in the answers to some questions of the survey (e.g. for the requirements reuse level) could be matter of discussion, it was what we intended from the outset. Because of the length and complexity of the survey, we did not want to use quantitative measures in the answers. In Dillman et al.'s book (2014) it is stated that "Clearly, the time required to answer questions is a significant cost to the respondent, and many do not take the time required to respond. However, the realization by the respondent that he cannot provide accurate answers to questions increases the sense of burden further. The desire of surveyors to obtain answers to increasingly detailed questions needed for complex modeling of attitudes and behaviors appears often to be in conflict with the limitations and patience of respondents for providing answers." Asking for too many details would have made people bored and, as a consequence, they would have abandoned the survey. We firmly believe the completion rate would have been worse if we had used quantitative metrics too often.

Sampling validity. The population we were interested in was requirements engineers with industrial experience (either pure practitioners or researchers that worked or had worked in industry as requirements

engineers). Given their professional scope and skills, we may assume that they were an Internet-aware population with enough expertise to answer an online survey without technological impediments.

For the selection of a representative sample of this population, we focused on social networks and forums where these requirements engineers meet each other: RE conferences (Franch et al. 2013a, Franch 2013b, Palomares et al. 2014a), RE LinkedIn groups, and RE magazines (Palomares et al. 2014b). The sources of sample are suitable to find representative participants from the population. For instance, LinkedIn is a suitable forum to find a good representative sample of the population that we wanted to reach (de Mello et al. 2015), since it has become a common website for professionals of every type of position, and there are specific groups where requirements engineers meet. However, two threats jeopardize the results of our survey: the way the sample has been selected and the amount of answers. We cannot claim that the sample is truly random (because they were the participants that chose to answer the survey after seeing the announcement in a LinkedIn group or in another forum), but it is certainly less of a convenience sample than have been obtained from directly contacting industry representatives ourselves. With regard to the number of answers, the problem is that we do not know the number of requirements engineers that our survey announcement reached, and we do not know the percentage of people that answered the survey. However, we find it reasonable to state that the sources we used through the Internet allowed us to get many more responses than contacting individuals in industrial organizations, whilst also being more diverse and representative. Additionally, in order to screen out possible inappropriate individuals, we had a question in the survey to determine the grade of industrial experience in RE in order to be able to filter out responses from IT people with no industrial RE experience.

Another fact that may have affected the sampling is the title of the survey (*Requirement Reuse and Patterns*), therefore attracting more people that were interested in reuse and eventually skewing the answers towards positive observations. This is a difficult threat to avoid because the title of the survey needs to describe the content in order to attract expert respondents. We tried to mitigate this risk by including a balanced set of responses in multi-option questions, i.e. combining adequate positive and negative answers.

To compare our strategy to mitigate sample threats, we searched other similar studies in the corpus of publications between 2010 and 2015 in the Empirical Software Engineering Journal (which is the flagship journal in empirical studies in the software engineering area). Only two publications presented surveys conducted online (Chen et al. 2015, Dietrich et al. 2015) and advertised in a similar way to our survey (LinkedIn, mailing lists, social networks of some relevant association). Neither of them stated the potential respondents of their survey, just the percentage of answers from people that accessed the survey, and they recognized that it is not possible to be sure of the confidence of the sample because it was reached mainly through the Internet, which corroborates the threats that we have identified for our sample.

Looking at the distribution of the respondents according to their level of education, the fact that about 75% of the participants had an MSc or PhD might not seem to match the typical software company distribution of education in most countries. However, we think it is not a surprise. On the one hand, from the participants that knew about the survey through LinkedIn (32 people; 45%), most of them hold an MSc (18 people; 25%). Although we did not find statistics that classify LinkedIn users depending on their educational background, in the statistics presented in the Business Insider website¹⁷, the authors state that: “*Unsurprisingly for a job-*

¹⁷ <http://www.businessinsider.com/linkedin-as-a-marketing-and-brand-platform-2014-9>

oriented social network, LinkedIn's most notable demographic skew is based on education. Thirty-eight percent of college-educated adult web users are on LinkedIn, compared to only 16% of those with some college education.” On the other hand, from the respondents that knew about the survey because they attended some conference or because they were contacted by e-mail (31 people; 44%), most of them hold an MSc or PhD (28 people; 39%). Since most of the participants knew about the survey through LinkedIn, conferences, or personal e-mail contact, we think the level of education of our participants is not a surprise. Anyway, it is worth remembering that all of them had significant industrial experience (as there was a question in the survey to rule out those individuals without RE industrial experience).

We took several measures to avoid fake answers coming from people trying to sabotage or bias the study. First, in the platform used to implement the survey (Lime Survey), we activated detection of responses coming from the same IP address, making it impossible for the same person to submit the survey twice without changing the IP of their device. Second, we organized the online questionnaire into sections that were presented on different web pages. One or more questions on each page were impossible to skip so that the work involved in answering the survey would discourage people who were not really interested in the survey subject.

Finally, some readers may wonder if the RE communities where the survey was advertised had industrial people that we, or some of our colleagues, had previously collaborated with in the context of the requirements patterns work, thus affecting the validity of our sampling. Although it is not possible to completely mitigate the consequences of this fact (as some of the answers were anonymous), we firmly believe it does not impose big consequences on our sample. Firstly, there is only one organization that we have been closely collaborating with in our work about requirement patterns (members of the SSI department at TUDOR¹⁸). They were asked to review the questionnaire before conducting the survey and they did not participate in the study. Secondly, from the 62 participants that were not anonymous (i.e., the ones that entered their data at the end of the survey), there are only eight people that participated who belong to our research network, sometimes more closely (e.g., a former PhD student), sometimes only known by email (e.g., colleagues working in the topic with whom who have exchanged publications). These people are researchers with industrial experience, and we are sure they know the importance of answering a survey in a professional and scientific way, as they have conducted other empirical studies themselves too.

Participants' perceptions. Since we were using online questionnaires, if the survey proposal was perceived to be junk publicity, the outcome of the results could be affected. To avoid this, our survey invitation e-mails were sent to RE conferences attendees and RE communities from our academic e-mail addresses. We included a brief letter explaining the academic purpose of the survey and provided its link, and signed the e-mails with the authors' names. We also tried to provide a professional image by opening a specific webpage hosting the link (<http://www.upc.edu/gessi/PABRE/Survey.html>), which was a new tab in our web resource dedicated to the PABRE framework. In our announcements through the Internet (especially in the LinkedIn groups), we always presented the survey inside discussion topics and tried to keep them alive by participating in the discussions to show potential respondents that we were interested in them answering the survey for academic purposes.

¹⁸ <http://www.tudor.lu>

Low response rate. As Dillman et al. (2014) stated, one of the main problems of online surveys (besides finding a good sampling frame) is having very low participation rates. In our case, we estimated that we proposed the survey to approximately 30,000 members (through LinkedIn and community groups), and received 77 responses. However, we are not able to determine how many people actually read our adverts and decided not to participate. For instance, at the time of conducting the experiment, it was not possible to see how many people in a LinkedIn group had read or accessed the messages posted in that group. This fact implies that it is not possible to calculate the actual response rate.

In addition, in order to avoid low participation rates, in the case of LinkedIn we introduced a discussion topic with a question to engage people to participate in the discussion. Then we proposed the survey inside the discussion. We think that this resulted in a higher number of answers compared to just announcing the survey directly in the groups. For instance, a discussion that gave us additional answers was introduced as: *Are you using some requirements reuse practice during requirements engineering?* Even though these low participation rates (which are common in online questionnaires) cannot be used for a rigorous statistical analysis, they can be used to understand trends (Lethbridge et al. 2005).

Other surveys that addressed similar contexts and used similar channels had similar or lower participation rates (Milewski 2007, Umarji and Sim 2013, Solinski and Petersen 2014).

Low completion rate. From the 142 respondents that started to answer the survey, only 77 completed it, which represents a low completion rate. As a consequence of the survey pilots carried out, we realized that a high percentage of the non-completed attempts occurred in a specific section of the survey owing to a design error in the interface (for more information see Survey Design in Section 3). Before the change was implemented, the completion rate was about 45%; after the change, it grew to approximately 72%. Another aspect that might have influenced the completion rate was the length and complexity of the survey. Because of that, we decided not to use quantitative metrics when asking for aspects like requirements reuse, reducing the time needed to answer some questions.

9.2 External and Construct Validity

Results generalization. Although we tried to select participants in our questionnaire in a random way, the nature of their companies, research centers, and projects was very different. This diversity (added to the fact of having only 71 responses) does not allow for a generalization of the results; only observations about the current state of the practice can be made. Therefore, it is not possible to guarantee with a good level of confidence that the conclusions of our analysis correspond to results that would be obtained by conducting the same survey with the entire RE population.

In addition, the title of the survey, which was *Requirements reuse and patterns*, might have also influenced the respondents' answers to the questions related to requirements reuse (such as the reuse level or reuse techniques used), therefore affecting the generalization of the results. However, we do not believe the threat to be of high risk since, for instance, from the 56 participants answering the question about requirements reuse techniques, only six of them answered that they have used requirement patterns (around 10%), and it was the least used technique from all the ones proposed in the survey. We think that six people out of 56 is a reasonable number, considering that:

- In the survey we stated before the questions about patterns that “A Software Requirement Pattern consists of natural language templates for generating those requirements that are related to a specific objective (goal), as well as some information to identify its adequacy to a particular project and how it may be tailored to the project.” Thus, the existence of a set of reusable natural language requirement templates or the existence of a reusable document with natural language requirements could be considered as a patterns catalogue.
- In the literature, the term ‘pattern’ corresponds to templates of any kind. A former study (Cox et al. 2009) discovered that the use of templates is not so rare in requirements.

Also, it is worth pointing out that in the questions related to RQ3 (which is related to the benefits and drawbacks that could appear from the use of a catalogue of SRPs), the respondents were only giving their opinion as RE experts, so we cannot assume every benefit and drawback that was marked as relevant in the results actually would appear when using SRPs.

Finally, it can be argued that requirements reuse and requirement patterns may seem better suited to traditional RE practices, more related to heavyweight approaches to software development than to agile methods. However, it is a fact that many companies still work with traditional software development methods and some authors defend the position that, for some type of complex IT projects, agile methods are not the best suited (Schär et al 2015). For instance, a recent worldwide survey with 3.880 respondents reported that 53% of the population worked in companies in which non-agile methods continue to be adopted by more than one-half of the teams (VersionOne 2016). At a regional level, a SwissQ survey reported that 40% of more than 300 respondents in the Swiss IT community were developing with traditional, waterfall-oriented methods (SwissQ 2012b). It can be argued then that, at least for these type of companies, the results of this survey are of interest. On the other hand, to what extent agile and rapid software development methods, especially in the context of continuous deployment, may benefit from requirements reuse and requirements patterns, remains subject of further investigation. It may be argued that RE is equally important in all approaches to software development methods, and that the difference of RE activities among agile and more traditional methods is that in agile, these activities are executed continuously during the project (Schär et al 2015). Concerning reuse, the only available results we are aware of come from Chernak’s study (Chernak 2012) who reports that even for agile methods, the percentage of respondents in his study who declared practicing requirements reuse was 33%, compared to the 61% for non-agile methods. These numbers hint that even if the development methods influences requirements reuse, still this is a practice that may be important in the agile world.

Results accuracy. As quantitative measures were not used in the answers for some questions in the survey, the accuracy of the results of such questions might have been affected. The questions where this threat is applicable are the ones about the level of requirements reuse, the type of requirements likely to be reused, the problems mitigated by the use of an SRPs catalogue, and the critical factors and barriers for the successful adoption of an SRPs catalogue. Not having used a quantitative metric when asking for aspects like requirements reuse could end in people understanding differently, for instance, the *medium* category of this question, and therefore the results might have been affected. However, we believe that the extremes of these answers are commonly agreed, and for those participants answering *Nonexistent or very low* and those ones answering *Very high* the results are reliable. We decided to have simplified answers in the questions of the

survey as we were more focused on getting more responses, even if this implied having less accurate responses (see threat *Instrumentation* in the previous section for a discussion on how asking for too many details may affect the response rate).

Face validity. Face validity is the extent to which a measure addresses the desired concept, i.e. whether it measures what it is supposed to measure. In order to ensure face validity, we discussed with our collaborators in the PABRE project whether the proposed survey questions were a good representation to answer the research questions (RQs). The discussion indicated that the initial set of proposed questions seemed to be suited for answering the RQs.

Since by their very nature questionnaires are rigid and biased instruments owing to the predefined options that most questions offer, we tried to cover the most common answers in the questions that had a list of predefined values (single-answer, multiple-answer, or Likert scale questions). To do this, we collected the values from the principal books and publications on RE (e.g., Pohl 2010 and Hull et al. 2011). Furthermore, it was always possible to include alternative choices that were not explicitly offered by using open value fields.

Finally, for the analysis of questions using Likert Scales, even though we know that in the case of ordinal scales calculating the average is a controversial issue (Jamieson 2004), we decided to calculate it: we considered that the scales used in our case were not completely ordinal but more of the type interval (the average of which type may be calculated). This is corroborated by Boone and Boone (2012), where it is explained that when Likert scales (even if they are ordinal) are used in a series of items that when combined measure a particular trait, it is possible to use the average to present the results. In our case, the particular traits would be, for instance, the types of requirements more likely to be reused or the most common problems in RE that could be mitigated by the existence of an SRPs catalogue.

10 Conclusions and Further Work

In this paper, we have presented the results of a survey whose goal is to contribute to the investigation of the state of the practice in the reuse of requirements and to know the possible advantages, success factors, and barriers to the implementation of reuse in organizations. The study considered both reuse in general and reuse through requirement patterns in particular. In the analysis, we considered 71 valid and complete responses that came from IT professionals with some experience as requirements engineers in industry.

The answers to the RQs provide interesting insights for RE practitioners and researchers.

- *RQ1: What is the current state of the practice of requirements reuse?* The participants in our study declared that they reuse requirements often, but also in a limited way, since the most popular approaches are based on copy and paste of requirements from previous projects. Participants from larger organizations, and organizations with more established software development processes, are the ones who declare a higher reuse level, and less variety of reuse techniques used in their projects. Requirements more likely to be reused in requirement specifications are NFRs, since they are recurrent in participant projects.
- *RQ2: Why are existing requirements reuse proposals not being used in industrial practice?* The most common opinion expressed by the participants of the study was that organizations do not know how to fit requirements reuse into their RE processes. Concerns were also relative to return on investment due to

the initial effort required to set up the reuse infrastructure and the perceived complexity of reuse techniques.

- *RQ3: What benefits and drawbacks can emerge from the use of a catalogue of software requirement patterns?* Participants expressed their opinion that the problems that could be mitigated by the use of software requirement patterns are related to the quality of requirement specifications and with the efficiency of requirements elicitation. The critical factors and barriers are the existence of a well-defined reuse method, the tool support and the involvement of people in the software development team in the reuse process.

As future work, we foresee several research directions. First, we plan to investigate what existing approaches to requirements reuse have succeeded and the practical factors that cause such success by reviewing the literature on existing cases studies of requirements reuse.

We also want to validate whether or not the observations obtained from the survey results correspond to the real state of the practice of requirements reuse. To do so, we will use interviews or case studies involving requirements engineers that belong to industrial organizations of different sizes and with different levels of requirements reuse. With these new studies, we aim at gathering in-depth knowledge on the topic.

In addition, with the results of the survey and the interviews, we plan to work on a model for improving requirements reuse in organizations. In this model, practices, techniques, tools and actors that are more suitable for each reuse level will be defined and described in order to help organizations introduce or evolve their requirements reuse processes. Based on the observations of the survey we are currently envisioning five reuse levels: L1, for organizations that have not considered requirements reuse as an option; L2, for organizations that are already convinced of the benefits of reuse but that have not still implemented it; L3, for organizations that are already using the simple copy and paste technique and using natural language in their requirements specifications; L4, for organizations that already have a catalogue of requirement templates managed as a knowledge asset; and L5, for organizations that have a well-defined process to manage such a catalogue and customize it when applied to new projects. Once this initial theory is completed, our goal is to use semi-structured interviews as a preliminary validation.

Acknowledgments

We want to thank Claudia Ayala and Birgit Penzenstadler for their comments and suggestions. This work has been partially funded by the Spanish project TIN2013-44641-P.

References

- Arnuphaptrairong, T. (2011). Top ten lists of software project risks: Evidence from the literature survey. In proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS), Hong Kong, pp. 16-18.
- Bakar, N. H., & Kasirun, Z. M. (2014). Exploring software practitioners perceptions and experience in requirements reuse an empirical study in Malaysia. *International Journal of Software Engineering and Technology*, 1(2).

- Beecham, S., Hall, T., & Rainer, A. (2003). Software process improvement problems in twelve software companies: An empirical analysis. *Empirical Software Engineering*, DOI 10.1023/A:1021764731148.
- Boehm, B.W., & Basili, V. (2001). Software Defect Reduction Top 10 List. *Journal Computer*, 34 (1), DOI 10.1109/2.962984.
- Boone, H.N., & Boone, D.A. (2012). Analyzing Likert Data. *Journal of Extension*, 50(2), pp. 1-5.
- Bosch, J. (2000). *Design & Use of Software Architectures*, Addison-Wesley.
- Calleam Consulting (2015). Why do projects fail. 101 Common Causes. Last access on: February 2016. URL: http://calleam.com/WTPF/?page_id=2338.
- Carrillo-de-Gea, J. M., Nicolás, J., Alemán, J. L. F., Toval, A., Vizcaíno, A., & Ebert, C. (2013). Reusing requirements in global software engineering. *Managing requirements knowledge*, Springer Berlin Heidelberg, DOI 10.1007/978-3-642-34419-0_8
- Carvalho, J. P., Franch, X., & Quer, C. (2006). Managing non-technical requirements in COTS components selection. In proceedings of the 14th IEEE International Conference Requirements Engineering (RE), Minneapolis, pp. 323-326.
- Chen, J., Xiao, J., Wang, Q., & Osterweil, J.L. (2015). Perspectives on refactoring planning and practice: an empirical study. *Empirical Software Engineering*, DOI: 10.1007/s10664-015-9390-8.
- Chen, K., Zhang, W., Zhao, H., & Mei, H. (2005). An approach to constructing feature models based on requirements clustering. In proceedings of the 13th IEEE International Conference on Requirements Engineering (RE), Paris, pp. 31-40.
- Chernak, Y. (2012). Requirements reuse: the state of the practice. In proceedings of the IEEE Int. Conference on Software Science, Technology and Engineering (SWSTE), Herzelia, pp. 46-53.
- Chung, L., & Supakkul, S. (2006). Capturing and reusing functional and non-functional requirements knowledge: a goal-object pattern approach. In proceedings of the IEEE International Conference on Information Reuse and Integration (IRI), Waikoloa, pp. 539-544.
- Cox, K., Niazi, M., & Verner, J. (2009). Empirical study of Sommerville and Sawyer's requirements engineering practices. *IET software*, 3(5), pp. 339-355, DOI: 10.1049/iet-sen.2008.0076
- Daramola, O., Sindre, G., & Stalhane, T. (2012). Pattern-based security requirements specification using ontologies and boilerplates. In proceedings of the IEEE Second International Workshop on Requirements Patterns (RePa), Chicago, pp. 54-59.
- de Mello, R. M., Da Silva, P. C., & Travassos, G. H. (2015). Investigating probabilistic sampling approaches for large-scale surveys in software engineering. *Journal of Software Engineering Research and Development*, 3(1), DOI: 10.1186/s40411-015-0023-0
- Dietrich, J., Jezek, K., & Brada, P. (2015). What Java developers know about compatibility, and why this matters. *Empirical Software Engineering*, DOI: 10.1007/s10664-015-9389-1.
- Dillman, D. A., Smyth, J. D., & Christian, L. M. (2014). *Internet, phone, mail, and mixed-mode surveys: the tailored design method*. John Wiley & Sons.
- Duggan, M. (2015) *The Demographics of Social Media Users*. Last access on July 2016. <http://www.pewinternet.org/>

- Dyba, T. (2003). Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context. In ACM SIGSOFT Software Engineering Notes, 28(5), pp. 148-157.
- Dyba, T. (2005). An empirical investigation of the key factors for success in software process improvement. IEEE Transactions on Software Engineering, 31(5), DOI: 10.1109/TSE.2005.53
- Eriksson, M., Börstler, J., & Borg, K. (2009). Managing requirements specifications for product lines—An approach and industry case study. Journal of Systems and Software, DOI:10.1016/j.jss.2008.07.046
- Evans, J. R., & Mathur, A. (2005). The value of online surveys. Internet research, 15(2), DOI: 10.1108/10662240510590360.
- Filipovikj, P., Nyberg, M., & Rodriguez-Navas, G. (2014). Reassessing the pattern-based approach for formalizing requirements in the automotive domain. In proceedings of the 22th IEEE International Requirements Engineering Conference (RE), Karlskrona, pp. 444-450.
- Franch, X., Quer, C., Renault, S., Guerlain, C., & Palomares, C. (2013a). Constructing and using software requirement patterns. Managing requirements knowledge, Springer Berlin Heidelberg, DOI 10.1007/978-3-642-34419-0_5.
- Franch, X. (2013b). Software requirement patterns. Tutorial Summary in proceedings of the 35th International Conference on Software Engineering (ICSE), San Francisco, pp. 1499-1501.
- Galorath, D. (2012). Software Project Failure Costs Billions-Better Estimation & Planning Can Help. Galorath Incorporated. Last access on February 2016. <http://galorath.com/wp/software-project-failure-costs-billions-better-estimation-planning-can-help/>.
- Goldin, L., & Berry, D. M. (2013). Reuse of requirements reduced time to market at one industrial shop: a case study. Requirements Engineering, 20(1), DOI: 10.1007/s00766-013-0182-7
- Haeng-Kon, K. (2014). Effective Domain Modeling for Mobile Business AHMS (Adaptive Human Management Systems) requirements. International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD).
- Hall, T., Beecham, S., & Rainer, A. (2002). Requirements problems in twelve software companies: an empirical analysis. IEE Proceedings-Software, 149(5), DOI: 10.1049/ip-sen:20020694.
- Hauksdottir, D., Vermehren, A., & Savolainen, J. (2012). Requirements reuse at Danfoss. In proceedings of the 20th IEEE International Conference on Requirements Engineering (RE), Chicago, pp. 309-314.
- Hoffmann, A., Sollner, M., Hoffmann, H., & Leimeister, J. M. (2012). Towards trust-based software requirement patterns. In proceedings of the IEEE Second International Workshop on Requirements Patterns (RePa), Chicago, pp. 7-11.
- Hoffmann, A., Janzen, A., Hoffmann, H., & Leimeister, J. M. (2013). Success Factors for Requirement Patterns Approaches Exploring Requirements Analysts' Opinions and Whishes. Sozio-technisches Systemdesign im Zeitalter des Ubiquitous Computing (SUBICO) im Rahmen der Informatik.
- Hull, E., Jackson, K., & Dick, J. (2011). Requirements Engineering, 3rd edition. Springer Publishing Company, Incorporated.

- IEEE Computer Society. Software Engineering Standards Committee, & IEEE-SA Standards Board. (2009). IEEE Recommended Practice for Software Requirements Specifications. Institute of Electrical and Electronics Engineers.
- Iqbal, J., Ahmad, R., Nizam, M. H., Nasir, M., & Noor, M. A. (2013). Significant requirements engineering practices for software development outsourcing. In proceedings of the 22nd Australian Software Engineering Conference (ASWEC), Melbourne, pp. 137-144.
- Issa, A. A., & Al-Ali, A. (2010). Use Case Patterns Driven Requirements Engineering. In proceedings of the Second International Conference on Computer Research and Development (ICCRD), Kuala Lumpur, pp. 307-313.
- Jamieson, S. (2004). Likert scales: how to (ab) use them. *Medical education*, 38(12), DOI: 10.1111/j.1365-2929.2004.02012.x.
- Jensen, J., Tondel, I. A., Jaatun, M. G., Meland, P. H., & Andresen, H. (2009). Reusable Security Requirements for Healthcare Applications. In proceedings of the International Conference on Availability, Reliability and Security (ARES), Fukuoka, pp. 380-385.
- Jones, C. (2014). Scoring and Evaluating Software Methods, Practices, and Results. Version 12.0. Last access on February 2016. <http://Namcookanalytics.com>.
- Karatas, E. K., Iyidir, B., & Birturk, A. (2014). Ontology-Based Software Requirements Reuse: Case Study in Fire Control Software Product Line Domain. In proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW), Shenzhen, pp. 832-839.
- Khankaew, S., & Riddle, S. (2014). A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand. In proceedings of the 2014 IEEE Fourth International Workshop on Empirical Requirements Engineering (EmpiRE), Karlskrona, pp. 1-8.
- Kitchenham, B., & Pfleeger, S. L. (2003). Principles of survey research part 6: data analysis. *ACM SIGSOFT Software Engineering Notes*, 28(2), pp. 24-27.
- Konrad, S., & Cheng, B. H. (2002). Requirements patterns for embedded systems. In proceedings of the IEEE Joint International Conference on Requirements Engineering (RE), Essen, pp. 127-136.
- Konrad, S., & Cheng, B. H. (2005). Real-time specification patterns. In proceedings of the 27th International Conference on Software engineering (ICSE), St. Louis, pp. 372-381.
- Krippendorff, K. (2012). *Content analysis: An introduction to its methodology*. Third edition, Sage Publications, Thousand Oaks.
- Lam, W. (1998). A case-study of requirements reuse through product families. *Annals of Software Engineering*, 5(1), DOI: 10.1023/A:1018912105115.
- Lam, W., McDermid, J. A., & Vickers, A. J. (1997). Ten steps towards systematic requirements reuse. *Requirements Engineering*, 2(2), DOI: 10.1109/ISRE.1997.566834.
- Lethbridge, T. C., Sim, S. E., & Singer, J. (2005). Studying software engineers: Data collection techniques for software field studies. *Empirical software engineering*, 10(3), DOI: 10.1007/s10664-005-1290-x.

- Li, Y., Pelties, C., Kaser, M., & Nararan, N. (2012). Requirements patterns for seismology software applications. In proceedings of the IEEE Second International Workshop on Requirements Patterns (RePa), Chicago, pp. 12-16.
- Lo, D., Nagappan, N., Zimmermann, T. (2015). How Practitioners Perceive the Relevance of Software Engineering Research. In proceedings of the 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE), Bergamo, pp. 415-425.
- Mannion, M., & Kaindl, H. (2008). Using parameters and discriminants for product line requirements. *Systems engineering*, 11(1), DOI: 10.1002/sys.20086.
- Massonet, P., & Van Lamsweerde, A. (1997). Analogical reuse of requirements frameworks. Proceedings of the Third IEEE International Symposium on Requirements Engineering (RE), Annapolis, pp. 26-37.
- Matloff, N. (2009). From Algorithm to Z-Scores: Probabilistic and Statistical Modeling in Computer Science. Last access on: February 2016. URL: <http://heather.cs.ucdavis.edu/~matloff/probstatbook.html>.
- Matulevicius, R. (2005). Survey of Requirements Engineering Practice in Lithuanian Software Development Companies. *Information Systems Development*, Springer US, DOI: 10.1007/0-387-28809-0_29.
- Méndez, D., & Wagner, S. (2015). Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany. *Information and Software Technology*, 57(1), DOI:10.1016/j.infsof.2014.05.008.
- Mich, L., Franch, M., Novi, I.P. (2004). Market research for requirements analysis using linguistic tools. *Requirements Engineering*, 9(2), DOI: 10.1007/s00766-004-0195-3.
- Milewski, A. E. (2007). Global and task effects in information-seeking among software engineers. *Empirical Software Engineering*, 12(3), DOI: 10.1007/s10664-007-9036-6
- Naish, J., & Zhao, L. (2011). Towards a generalised framework for classifying and retrieving requirements patterns. In proceedings of the First International Workshop on Requirements Patterns (RePa), Trento, pp. 42-51.
- Neill, C. J., & Laplante, P. A. (2003). Requirements engineering: the state of the practice. *IEEE software*, 20(6), DOI: 10.1109/MS.2003.1241365.
- Niazi, M., Cox, K., & Verner, J. (2008). A measurement framework for assessing the maturity of requirements engineering process. *Software Quality Journal*, 16(2), DOI: 10.1007/s11219-007-9033-4
- Niazi, M., El-Attar, M., Usman, M., & Ikram, N. (2012). An empirical study identifying high perceived value requirements engineering practices in global software development projects. In proceedings of the 7th International Conference on Software Engineering Advances (ICSEA), Lisbon, pp. 283-288.
- Nikula, U., Sajaniemi, J., & Kälviäinen, H. (2000). A State-of-the-practice Survey on Requirements Engineering in Small-and Medium-sized Enterprises. Research Report. Lappeenranta University of Technology, Finland.
- Nuseibeh, B., & Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In proceedings of the Conference on The Future of Software Engineering (FOSE), International Conference on Software Engineering (ICSE), Limerick, pp 35-46.

- Pacheco, C. L., Garcia, I. A., Calvo-Manzano, J. A., & Arcilla, M. (2015). A proposed model for reuse of software requirements in requirements catalog. *Journal of Software: Evolution and Process*, DOI: 10.1002/smr.1698.
- Palomares, C., Franch, X., & Quer, C. (2014a). Requirements reuse and patterns: a survey. In *Requirements Engineering: Foundation for Software Quality*. In proceedings of the 20th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), Essen, pp. 301-308.
- Palomares C., Quer C., & Franch X. (2014b). Requirements Reuse with the PABRE Framework. *Requirements Engineering Magazine*. 2014(1), IREB.
- Palomares, C., Quer, C., Franch, X., Renault, S., & Guerlain, C. (2013). A catalogue of functional software requirement patterns for the domain of content management systems. In proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC), Coimbra, pp. 1260-1265.
- Panis, M. C. (2015). Reuse of Architecturally Derived Standards Requirements. *IEEE International Requirements Engineering Conference (RE)*, Ottawa, pp. 296-304.
- Pohl, K. (2010). *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated.
- Post, A., Menzel, I., & Podelski, A. (2011). Applying restricted English grammar on automotive requirements—does it work? A case study. In proceedings of the 17th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), Essen, pp. 166-180.
- Procaccino, J. D., Verner, J. M., Overmyer, S. P., & Darter, M. E. (2002). Case study: factors for early prediction of software development success. *Information and software technology*, 44(1), DOI: 10.1016/S0950-5849(01)00217-8.
- Renault, S., Méndez, O., Franch, X., & Quer, C. (2009). A Pattern-based Method for building Requirements Documents in Call-for-tender Processes. *International Journal of Computer Science and Applications*, 6(5), pp.175-202.
- Rine, D. C., & Nada, N. (2000). An empirical study of a software reuse reference model. *Information and Software Technology*, 42(1), DOI: 10.1016/S0950-5849(99)00055-5.
- Robertson, J., & Robertson, S. (2000). *Volere Requirements Specification Template*. Atlantic System Guild, Last access on February 2016. <http://www.volere.co.uk/template.htm>.
- Sadraei, E., Aurum, A., Beydoun, G., & Paech, B. (2007). A field study of the requirements engineering practice in Australian software industry. *Requirements Engineering*, 12(3), DOI: 10.1007/s00766-007-0042-4.
- Salini P, & Kanmani S. (2012). A knowledge-oriented approach to security requirements for an E-voting system. *International Journal of Computer Applications*, 49(11), DOI: 10.5120/7671-0953
- Sawyer, P., Sommerville, I., & Viller, S. (1997). Requirements process improvement through the phased introduction of good practice. *Software Process: Improvement and Practice*, 3(1), DOI: 10.1002/(SICI)1099-1670(199703)3:1<19::AID-SPIP66>3.0.CO;2-X.

- Schirpenbach, J. (2014). Re-Use of Requirements via Libraries: Opportunities & Approaches". Requirements Engineering Magazine. 2014(2), IREB.
- Shahrokni, A., & Feldt, R. (2010). Towards a framework for specifying software robustness requirements based on patterns. In proceedings of the 16th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), Göteborg, pp. 79-84.
- Solemon, B., Sahibuddin, S., & Ghani, A. A. A. (2009). Requirements engineering problems and practices in software companies: An industrial survey. *Advances in Software Engineering*, Springer Berlin Heidelberg, DOI: 10.1007/978-3-642-10619-4_9
- Solemon, B., Sahibuddin, S., & Ghani, A. A. A. (2010). Adoption of requirements engineering practices in Malaysian software development companies. *Advances in Software Engineering*, Springer Berlin Heidelberg, DOI: 10.1007/978-3-642-17578-7_15.
- Solinski, A., & Petersen, K. (2014). Prioritizing agile benefits and limitations in relation to practice usage. *Software Quality Journal*, DOI: 10.1007/s11219-014-9253-3.
- Sommerville, I., & Sawyer, P. (1997). *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc.
- Sommerville, I., Ransom, J. (2005). An empirical study of industrial requirements engineering process assessment and improvement. *ACM Transactions on Software Engineering and Methodology*, 14(1), DOI: 10.1145/1044834.1044837.
- Souag, A., Mazo, R., Salinesi, C., & Comyn-Wattiau, I. (2015). Reusable knowledge in security requirements engineering: a systematic mapping study. *Requirements Engineering*, DOI 10.1007/s00766-015-0220-8.
- Srivastava, S., (2013). A Repository of Software Requirement Patterns for Online Examination System. *International Journal of Computer Science*, 10(3).
- Supakkul, S., Hill, T., Chung, L., Tun, T. T., & Sampaio do Prado Leite, J. C. (2010). An NFR pattern approach to dealing with NFRs. In proceedings of the 18th IEEE International Requirements Engineering Conference (RE), Sydney, pp. 179-188.
- SwissQ (2012a), *SwissQ Requirements Trends & Benchmarks Switzerland 2012*. Last access on: October 2016. URL: <http://www.slideshare.net/swissq/agile-and-requirements-trends-benchmarks-2012-english>.
- SwissQ (2012b), *SwissQ Agile Trends & Benchmarks Switzerland 2012*. Last access on: October 2016. URL: http://www.mcpa.biz/wp-content/uploads/2015/09/SwissQ_Agile_Trends_2012_Web_EN-1.pdf.
- Schär, B., Jüngling, S., & Thönsen, B. Towards an Agile Requirements Engineering Process Combining HERMES 5 and SCRUM. In proceedings of the International Conference on Enterprise Systems (ES), Basel.
- Tahir, A., & Ahmad, R. (2010, December). Requirement engineering practices-an empirical study. In proceedings of the International Conference on Computational Intelligence and Software Engineering (CiSE), Wuhan, pp. 1-5.
- Tiwana, A. (2004). An empirical study of the effect of knowledge integration on software development performance. *Information and Software Technology*, 46(13), DOI:10.1016/j.infsof.2004.03.006

- Toval, A., Nicolás, J., Moros, B., & García, F. (2002). Requirements reuse for improving information systems security: a practitioner's approach. *Requirements Engineering*, 6(4), DOI: 10.1007/PL00010360.
- Trochim, W. M., & Donnelly, J. P. (2006). Research methods knowledge base. Last access on February 2016, URL: <http://www.socialresearchmethods.net/kb/>.
- Umarji, M., & Sim, S. E. (2013). *Archetypal Internet-Scale Source Code Searching. Finding Source Code on the Web for Remix and Reuse*, Springer New York, DOI: 10.1007/978-0-387-09684-1_21
- VersionOne (2016). The 10th Annual State of Agile™ Report. Last access on October 2016. URL: <http://stateofagile.versionone.com/> (by subscription).
- von Knethen, A., Paech, B., Kiedaisch, F., & Houdek, F. (2002). Systematic requirements recycling through abstraction and traceability. In proceedings of the IEEE Joint International Conference on Requirements Engineering (RE), Essen, pp. 273-281.
- Wieggers, K., & Beatty, J. (2013). *Software requirements*. Microsoft Press.
- Withall, S. (2007). *Software requirement patterns*. Microsoft Press.
- Wohlin, C., Runeson, P., Host, M., Ohlsson, M. C., Regnell, B., & Wesslen, A. (2012). *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers Norwell, MA, USA.
- Wright, K. B. (2005). Researching Internet-based populations: Advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services. *Journal of Computer-Mediated Communication*, 10(3), DOI: 10.1111/j.1083-6101.2005.tb00259.x.
- Young, R. (2015). The Main Thing is Keeping the Main Thing the Main Thing". *Requirements Engineering Magazine*, 2015(1), IREB.