

# Industrial Practices on Requirements Reuse: An Interview-based Study

Xavier Franch  <sup>[0000-0001-9733-8830]</sup>, Cristina Palomares<sup>[0000-0003-4722-5584]</sup>,  
Carne Quer<sup>[0000-0002-9000-6371]</sup>

Universitat Politècnica Catalunya (UPC-BarcelonaTech), Barcelona, Spain  
{franch|cpalomares|cquer}@essi.upc.edu

**Abstract.** **[Context and motivation]** Requirements reuse has been proposed as a key asset for requirements engineers to efficiently elicit, validate and document software requirements and, as a consequence, obtain requirements specifications of better quality through more effective engineering processes. **[Question/problem]** Regardless the impact requirements reuse could have in software projects' success and efficiency, the requirements engineering community has published very few studies reporting the way in which this activity is conducted in industry. **[Principal ideas/results]** In this paper, we present the results of an interview-based study involving 24 IT professionals on whether they reuse requirements or not and how. Some kind of requirements reuse is carried out by the majority of respondents, being organizational and project-related factors the main drivers. Quality requirements are the type most reused. The most common strategy is find-copy-paste-adapt. Respondents agreed that requirements reuse is beneficial, especially for project-related reasons. The most stated challenge to overcome in requirements reuse is related to the domain of the project and the development of a completely new system. **[Contribution]** With this study, we contribute to the state of the practice in the reuse of requirements by showing how real organizations carry out this process and the factors that influence it.

**Keywords:** Requirements reuse, requirements elicitation, requirements documentation, requirements engineering, survey, interview-based study.

## 1 Introduction

Requirements reuse is the practice of systematically eliciting and specifying requirements not starting from scratch, but from already available artefacts. These artefacts range from requirements appearing in previous requirement specification documents, to templates stored in some sort of catalogue, adapted to every new project.

As reported in Section 3, there is a good number of research works addressing software reuse in the scientific literature. Still, not many of them report on the state of the practice. Questions as: is requirements reuse an extended practice among requirements engineers in industry?, if so, how is it implemented?, what are the challenges to overcome and the perceived benefits?, require further field investigation through empirical studies with practitioners. In this paper, we report the results of a study in this direction.

The rest of the paper is organized as follows. Sections 2 and 3 describe the background and related work on requirements reuse that contextualizes our study. Section 4 describes the methodological aspects of the study, including the research questions, population and analysis procedures. Sections 5 and 6 present the results and their discussion organized according to the research questions. Finally, Section 7 concludes the paper and outlines some lines of future work.

## 2 Background

Two recent papers described in detail the background needed in this study. In a previous paper [1], we included a detailed description of background on requirements reuse until December 2015. In a paper published one year after [2], Irshad et al. presented a systematic literature review on requirements reuse based on publications previous to March 2016. The classification criteria that appear in both papers and that help to characterize better the reuse approaches are: the type of reusable artefacts and the means to retrieve the reusable artifacts from a repository. In this section, we classify the background according these criteria. The background consists on approaches identified in [1] [2] enriched with a search of approaches published after March 2016 in the three main RE-specific scientific venues: RE, REJ and REFSQ.

The type of reusable artifacts that are mainly used in the existing proposals are: requirements in natural language that can comply or not a certain form or template [3][4]; domain models [5][6]; and use cases [7][8]. Other artifacts that are not strictly requirements but that are also reused are: ontologies [9][10]; classifications of requirements in requirement specification or specification templates [11][12][13][14] and relationships or dependencies among requirements or models for reuse [11][15][16].

The means to retrieve the reusable artifacts that are mainly proposed in the existing proposals are: pattern based, using information embedded in the pattern (e.g., its goal) [11][17]; matching based, comparing and matching with existing requirements [8][18]; analogy based, searching for similar cases for applying requirements to be reused from past projects [19]; and ontology based, which in this case use ontologies not as reuse artifact but as a way to facilitate the retrieval of requirements to reuse suitable for a project [20][21].

Other criteria that can be mentioned are: the existence of a process prescribing the reuse strategy; the existence of tool support; the elaboration of the structure of the repository. For details, we refer again to the two papers mentioned above [1][2].

## 3 Related Work

In this section, we identify existing studies with similar aims than ours. We can consider two different types of studies.

**Type 1:** papers with requirements reuse proposals applied and validated in an industrial context. We already found out through a systematic search of publications (see [1] for more details) that there is a low percentage of papers on requirements reuse that

conduct an experimental validation. Complementing this information, Irshad et al. reported that from the 69 papers analysed in their review, only 22 were validated in industry [2]. Remarkably, after the validation, the papers only describe positive results with respect to requirements reuse, and do not mention the possible negative aspects. The main result reported is the decrease of effort.

**Type 2:** papers presenting the state of the practice of requirements reuse in industry through surveys or interviews. We identified in [1] two secondary studies that present surveys or interviews fully focused on requirements reuse [22] [23]. Applying the same systematic search used in [1], we have found one new paper published after 2016 in the context of ERP implementations [24]. The works reported in [22] [23] address four aspects of reuse in practice, the level of application, the benefits, obstacles and critical factors. As levels of reuse the results of both surveys give a level of reuse from 59% to 72%. Benefits reported from reuse are: the improved performance of requirements engineering; reduced projects costs; and requirements easier to understand. Regarding obstacles, they mention: the risk of low maintenance of the reuse repository and difficulty of identifying requirements to reuse. Furthermore, a critical factor that was observed is the existence of tool support for reuse. The work in [24] is not relevant for the work presented here, since it has a different goal, namely to identify reusable requirement artifacts used in the ERP implementation industry, and to define metrics to measure reusability of artifacts.

## 4 The Study

In this section we summarize the study protocol; full details are given in the document available at [25]. As a preamble, it is necessary to say that the results reported in this paper are part of a broader study that includes research questions not only on requirements reuse but also on requirements elicitation and documentation. For the sake of clarity, we present the part related to reuse as if it were an independent study.

**Research Questions.** This study aims at investigating the state of the practice on requirements reuse. To conduct this investigation, we identified the research questions (RQs) shown in **Table 1**. The overall question is to investigate how extended is requirements reuse in industry (**RQ1**). Since we do not expect this RQ to have an uncontextual answer, we inquiry next what factors may influence in reusing or not reusing (**RQ2**). Also, from former studies, we are aware that some types of requirements may be more prone to reuse than others, motivating **RQ3**. We are also interested in knowing the way in which requirements reused is implemented in practice (**RQ4**). Last, we want to understand the benefits (**RQ5**) and challenges (**RQ6**) brought by reuse as perceived by practitioners, since the final decision will be a trade-off of both.

**Table 1.** Research questions of the study.

<b>RQ1</b>	Is requirements reuse a usual practice in industry?
<b>RQ2</b>	What factors influence the level of adoption of requirements reuse?
<b>RQ3</b>	What types of requirements are subject of reuse?

<b>RQ4</b>	What is the process followed to implement requirements reuse?
<b>RQ5</b>	What are the benefits brought by requirements reuse?
<b>RQ6</b>	What are the challenges to overcome in requirements reuse?

**Population.** We interviewed 24 subjects coming from 12 companies working in different domains. Although our aim was having 2 subjects per company, we finally had one company with only 1 subject (company ID = C) and another one with 3 (company ID = J). Details are provided in **Table 2**. For brevity, we only include information that is referred to in Sections 5 and 6.

**Table 2.** Subjects, companies and projects of the study

ID	Exper.	Comp.	Years	Project domain	Methodology
S1	15	A	3	Messaging System	Waterfall
S2	15	A	10	Website	Waterfall
S3	20	B	≈9	Website	Agile
S4	13	B	13	Website	Agile
S5	25	C	4	Mobile OS	Waterfall
S6	20	D	20	Machine to Machine Internal System	Agile
S7	19	D	19	Carrier Business / Internal System	Agile
S8	15	E	15	Energy Measurement System	Waterfall
S9	20	E	6	Business Support System	Waterfall
S10	16	F	9	Carrier Business / Internal System	Agile
S11	17	F	0	Website	Agile
S12	12	G	≈9	Carrier Business / Internal System	Waterfall
S13	23	G	14	Carrier Business / Internal System	Waterfall
S14	10	H	5	Embedded System	Waterfall
S15	10	H	4.5	Embedded System	Waterfall
S16	25	I	19	Embedded System	Agile
S17	8	I	8	Embedded System	Waterfall
S18	9	J	2	Embedded System	Waterfall
S19	3	J	2	Embedded System	Waterfall
S20	23	J	16	Embedded System	Waterfall
S21	21	K	12	Mobile App	Waterfall
S22	9	K	9	Mobile App, Website	Agile
S23	15	L	4.5	Construction	Waterfall
S24	26	L	3.5	Construction	Waterfall

Exper.: Years in Industry; Comp.: Company ID; Years: Years in Organization

**Procedure and Instruments.** In order to gather data from the target population, we designed a semi-structured interview guide following the guidelines stated by Oates [26]. We asked the respondents to focus on one single project in order to gain as much insights as possible. The interview guide is available at [25]. The guide was piloted. The results were recorded but not transcribed according to the respondents' request.

**Data Analysis.** We applied coding techniques [27] with the support of the Atlas.ti<sup>1</sup> tool. We used multiple coding techniques in different steps, see details in [25]. We also applied some statistical techniques to look for associations among variables, remarkably

<sup>1</sup> <http://atlasti.com/>

Chi-square test of independence [28] (considering statistical significance if p-value < 0.05) and Cramer's V to estimate the strength of the association (strong association if V is  $\geq 0.5$ , moderate if  $V \geq 0.3$ ) [29]. Not all correlations neither all significant correlations are reported, but only those that we think are interesting and eventually explainable. Therefore, some characteristics of the participants of projects that one might thought might influence the results (such as the years of experience or the development methodology used) are only mentioned in the results when it showed interesting insights.

**Threats to Validity.** We outline threats to validity and outline strategies used to deal with them. Again we refer to the protocol document for complete explanations [25].

- *Construct Validity.* The study was supported by two main principles: rigorous planning and the establishment of protocols for data collection and data analysis as suggested in [30]. Additionally, the interview guide was designed and piloted. Finally, both in the interview guide and during the actual interview, the subjects were aware that the data and information they provided would be confidential, anonymised, and aggregated with the rest of interviews, so the subjects could freely share their real experiences and perceptions.
- *Conclusion Validity.* Throughout the coding, many concepts and their relationships were identified. Traceability from the raw data to the categories, and their relationships, was preserved. Different coding techniques (theory triangulation) were used to capture various aspects of the reuse phenomenon.

Given that some respondents didn't really apply reuse in the selected project or even in any other project (see RQ1 results), some of the responses given in RQ2-RQ6 are more based on educated opinions than in past experiences. We opted by not considering some of these responses if we thought they were too vague or too speculative.

- *Internal Validity.* We focused most of the questions on a single software development project. In this way, it was possible to further inquire and analyse specific contexts that generated a particular decision. To avoid subsequent threats: 1) the interview guide was sent in advance to the respondents so they rarely had difficulties remembering project details; 2) to minimize the risk of selecting only successful projects, we remarked that the study was not focused on analysing "wrong practices" but on knowing "how it is done in industrial practices".

To address single researcher bias in the coding process, we applied triangulation in different forms. Selected interviews were analysed independently by two researchers and the results were discussed to identify and eliminate any individual biases. Responses were triangulated too. In addition, the generated categories were analysed, discussed and reviewed by the team to ensure their accuracy, understanding and agreement.

- *External Validity.* Qualitative studies rarely attempt to make universal generalizations beyond the studied setting. Instead, as Robson explains [31], they are more concerned with characterizing, explaining and understanding the phenomena under the contexts of study. Still, we took some actions to strengthen external validity: 1) combination

of convenience sampling and maximum variation sampling to select the companies; 2) freedom for respondents to choose the project for the interview. In addition, also a third party involved in the study had contacts with the companies, she did not know about their way of working with respect to requirements engineering, so no pre-selection criteria was applied to choose the companies involved in the study. To support replication and validation by independent researchers, we are making available not just the protocol but all the coding results for the answers to research questions [32].

## 5 Results

In this section we report the results obtained for the 6 research questions. As general strategy, we applied content analysis to identify categories and subcategories. We usually present the results with bar graphics where the values are the number of respondents who answered in this category; the total number of respondents is explicitly stated given that usually it will be lower than the sum of respondents per category. To present the results, we have followed a narrative style integrating quasi-quotes from the respondents<sup>2</sup> in the general explanations. These quasi-quotes include the identifier of the subjects between curly braces.

### 5.1 RQ1. Is requirements reuse a usual practice in industry?

A slight majority of respondents (14) stated that there was some kind of reuse in the projects that they reported in the survey (see **Fig. 1**, left).

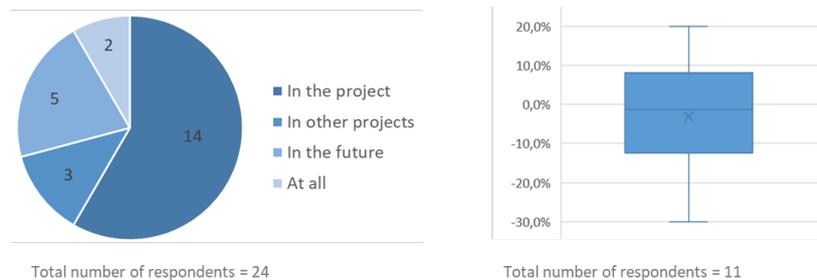
The level of reuse was quite balanced among them, with almost the same amount of respondents reporting high level of reuse (5 respondents informing up to 85%) and low level of reuse (6 respondents down to only 5%). The interpretation of “high” and “low” is quite consistent and the frontier seems to be in the interval [30%-50%], which was qualified as High percentage by one respondent, compared to [30%-40%] qualified as Low by another. Even the only respondent who qualified the level of reuse as Medium provides a consistent interval [60%-70%]. Effort reduction also oscillates, ranging from 5% to 80%. This effort reduction is focused on the requirements engineering stage: *less effort would have been put into the project, less time would have been needed for the elicitation of requirements, and less errors would have been made on the requirements* {S13}. As expected, there is a strong correlation among the level of the reuse and the effort reduction (see **Fig. 1**, right). It is interesting to quote the observation made by {S04} saying that in their case “more reuse would not have been more beneficial as they already reused as much as they could”.

As a follow-up question, to those respondents who reported lack of reuse in the selected projects, we raised some additional questions that allowed to see that 3 of them

---

<sup>2</sup> With “quasi-quotes” we mean syntactical adaptations of the sentences to make them fit to the story (e.g., including missing context in the sentence, aligning verb tenses, ...).

have reused requirements in other projects, while 5 others were expecting to start reuse in a near future. Overall, only 2 respondents reported complete lack of reuse.



**Fig. 1.** Requirements reuse in industry (left) and relation among level of reuse and effort reduction (right).

We explored the possible impact of demographic factors into software reuse adoption. We found an interesting statistically-relevant association with the project domain (p-value = 0.020; V value = 0.746), so that some domains seem more prone to reuse than others. As an extreme case, all 7 respondents who selected projects developing embedded systems did reuse requirements, while none of the 5 respondents who selected projects related to internal systems reused requirements.

## 5.2 RQ2. What factors influence the level of adoption of requirements reuse?

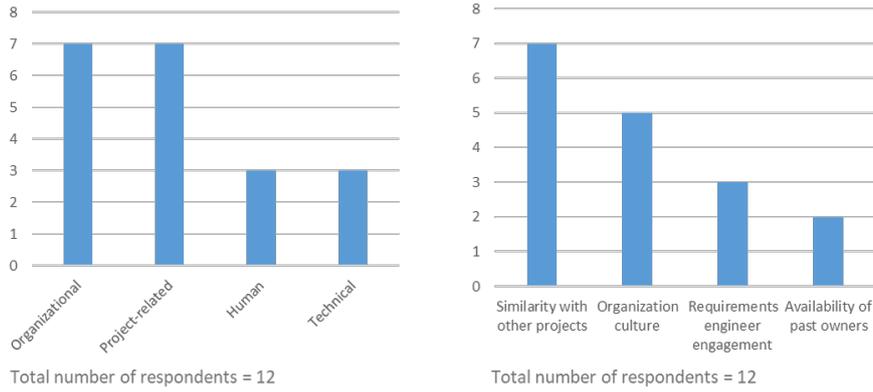
Given that 2 out of the 14 respondents that reported reuse in the selected projects in RQ1 were not able to identify factors influencing reuse, the answer to this research question is based on the responses given by 12 respondents only. From these interviews, we identified four big categories of factors

**Organizational.** Up to 7 respondents reported factors related to the organization. Among them, the most mentioned one (5 respondents) was organizational culture, e.g. *reuse seen as positive in the organization* {S19}. Also, a couple of respondents mentioned unavailability of previous specification documents for different reasons, like *not having any more intellectual rights of the previous documents (because the organization has been purchased by another)* {S18}. Finally, one respondent highlighted maturity of the organization, because *they have a quite mature requirements process that allows that the requirements of other projects are good enough to be reused* {S04}.

**Project-related.** The same number of respondents as above justified the adoption of requirements reuse upon the similarity of the selected project to similar previous projects. As obvious as it can seem, *if there is no previous system with some similar functionality or part that you can reuse, then it is impossible to reuse* {S03}.

**Human.** Only mentioned by 3 respondents, who justified the adoption of reuse upon the engagement and personal attitude of the requirements engineer who *put the extra effort to apply some requirements reuse* {S11}.

**Technical.** Also mentioned only by 3 respondents who provided a varied set of reasons, like *compliance to a new standard to be fulfilled by the platform* {S18} or impediments for tool support (*Electra was not used before, so not too many requirements were in Electra at the start of the project and they could not be reused* {S18}).

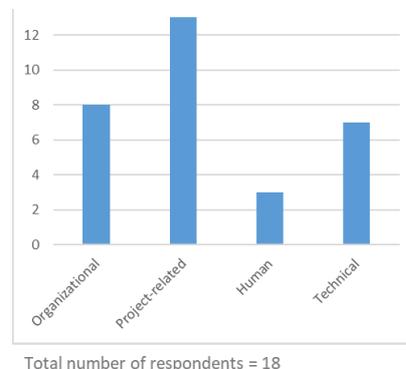
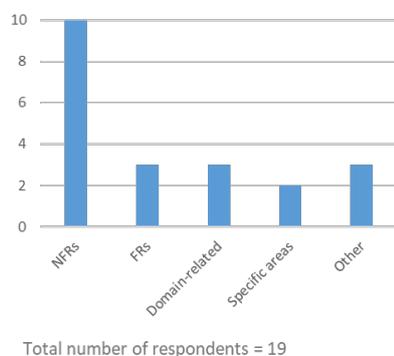


**Fig. 2.** Requirements reuse in industry: (a) categorization; (b) most frequent answers.

### 5.3 RQ3. What types of requirements are more prone to reuse?

The majority of respondents (up to 20) provided information related to this research question; only 4 of them didn't make it.

We got two categories of answers: types of requirements that are prone to be reused and types of requirements that are prone not to be reused. The majority of answers (from 19 respondents) come from the first category (see **Fig. 3**) and, in fact, one type (not surprisingly) prevails: non-functional/quality requirements (NFRs), reported by 10 respondents (e.g. *infrastructure requirements such as performance and network capacity* {S08}) *because these ones might be common to other projects* {S10}. Still 3 respondents mentioned functional requirements (FRs) when *they weren't the innovative part of the project* {S05} and therefore *they didn't change from previous projects* {S18}. The same number of respondents mentioned also reuse for domain-related requirements like *a search engine* {S03} or requirements related to a specific part of the system like *the client environment (operating systems, browsers, screen sizes, etc.)* {S11}. Remarkably, only one respondent mentioned reuse related to standards, namely for *health & safety issues* {S15}.



**Fig. 3.** Types of requirements prone to reuse. **Fig. 4.** Benefits brought by requirements reuse.

Types that prevent reuse were very different. For instance, {S23} identified as difficult to reuse the requirements that evolve too much technologically, whilst {S18} was more specific to a certain category, *the group that manages the requirements of the hardware used to activate it [a particular embedded subsystem] because from project to project what changes most is the hardware*.

#### 5.4 RQ4. What is the process followed to implement requirements reuse?

Again, we had 20 respondents providing details on the process followed, and this number increased until 22 considering some additional information related to the process itself, like techniques, artefacts and tools used during requirements reuse.

Similar to RQ3, a prevalent response emerged (see **Fig. 5, a**): the most popular process (10 respondents) was to search similar requirements in past requirements specification documents, copy them into the current specification and then adapt to the new project. This find-copy-paste-adapt strategy at the requirements level, was applied by 3 respondents at the software requirements specification document level by *looking at other projects, finding the most similar one, duplicating it and work in the parts as needed* {S05}.

Another approach to the reuse process followed by 4 respondents was the use of catalogues as central asset, i.e. *a kind of requirements repository, and user can check the repository by systems, subsystems, keywords, etc. and reuse something if necessary by linking to the source and copy and pasting to the new project* {S12}.

Some artefacts or concepts were mentioned when describing the process. Remarkably, three respondents talked about traceability to the source *having a link from where the requirements came from* {S18}. The use of tags or design rules (*set of requirements that all the projects/products have to comply with* {S07}) was also reported.

Last, we inquired the respondents about tool support (see **Fig. 5, b**). The majority of respondents (12) did not use tool support, but the rest did at some extent, in all cases using *the same tool as the one used for managing the requirements* {S24}. Half of them

(4) reported a tool-centric approach to the reuse process, e.g. *Jira allows reusing requirements by having a main ticket (requirement) to which you can link from a new system by creating a new ticket in another project that links to that main ticket* {S10}.

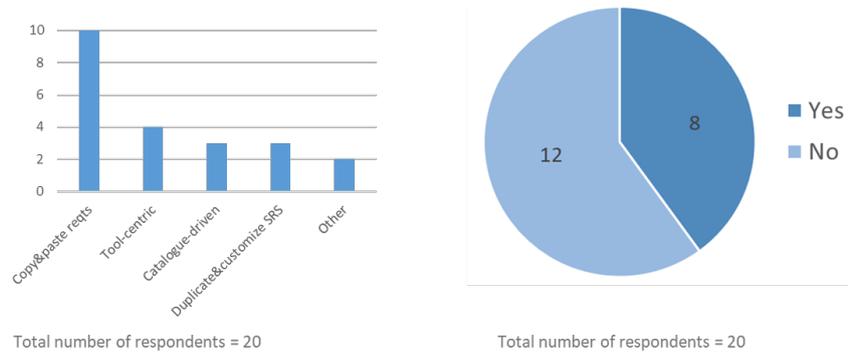


Fig. 5. (a) Processes followed to implement requirements reuse; (b) Existence of tool support.

### 5.5 RQ5. What are the benefits brought by requirements reuse?

Most of the respondents (22) considered reuse as a beneficial practice independently of whether they were reusing or not in the selected project (*it was beneficial* {S05} vs. *it would have been beneficial* {S01}). Only one respondent was absolutely negative because at the end *it was too much time to look for something, to reuse just a small part* {S03} while the other provided a contextual answer (*requirements reuse is not beneficial in my organization because everything we create is new* {S06}). Statistical analysis showed a moderate correlation ( $p$ -value = 0.037;  $V$  value = 0.426) between the perception of reuse being not beneficial and the type of project: the only 2 people not seeing reuse as beneficial work in projects following agile methodologies.

From those 22 respondents, 18 provided some justification to sustain their opinion. We classified the answers into the same categories as in RQ2 (see Fig. 3).

**Organizational.** 7 of the respondents reported causes related to the organization, in terms of *less effort needed for eliciting and specifying the requirements* {S02}. An additional reason given by {S11} only is that a reuse infrastructure makes it possible *not having to rely that much on people experience and knowledge*, making thus the organization less vulnerable. A moderate correlation ( $p$ -value = 0.020;  $V$  value = 0.473) informs that people perceiving organizational benefits also perceives project-related benefits (7 out of 8 respondents).

**Project-related.** Up to 12 respondents mentioned gain of efficiency in the RE process as the main benefit brought by reuse, mainly because *less time is needed for eliciting and specifying requirements* {S23}; more specifically, *if there is a high-level set of requirements already defined [...] to start the discussion, you could get faster to the key points* {S09}. The second reason given by only one respondent is that reuse can bring *a standard way of specifying requirements... including the level of detail and abstraction you should arrive* {S01}. We remark that we found a strong correlation ( $p$ -

value = 0.030; V value = 0.612) among this type of benefits and years in the organization: while 12 out of the 13 respondents who worked in the organization less than 15 years reported these benefits, only 1 out of 5 working more of 15 years reported them too. Another correlation was only moderate (p-value = 0.043; V value = 0.414) but also interesting: project-related benefits was mostly perceived by respondents working in projects that follow a waterfall methodology (11 out of 13 people).

**Human.** As happened with RQ2, this was the least influential category. Two respondents argued that reuse may ease communication inside the development team by *helping to put everyone together in the same page, from developers to testers* {S01}. An additional respondent highlighted that reuse was beneficial in front of customers *because it was easier to have something ready to introduce to customers* {S04} and get feedback from them earlier.

**Technical.** Some of the respondents argued that requirements reuse is beneficial from a technical perspective. One reason mentioned by 4 subjects was that with reuse *in general, you get more quality [in requirements], and because of that less errors* {S19}; only {S22} mentioned a more concrete quality criterion (*it is easier to write unambiguous requirements*). Other reasons were mentioned by one respondent each, e.g. *not only to reuse the requirements, but also the information about the effort and the historical data associated to the requirements (problems encountered, etc.)* {S16}, or also (and related) *the reuse of requirements implies for them also the reuse of code, tests, etc.* {S17}. It is worth to mention a statistically significant strong correlation here (p-value = 0.001; V value = 0.772): technical benefits are mostly perceived by people working in industry less than 10 years (and in fact, the 4 respondents in this situation reported technical benefits for reuse).

## 5.6 RQ6. What are the challenges to overcome in requirements reuse?

Remarkably enough, 8 respondents didn't identify any challenge to overcome related to requirements reuse. Answers given by the remaining 16 respondents showed two different strengths on the opinion. On the one hand, 11 respondents reported reasons that, either as an observation from the selected project or as an educated opinion, absolutely prevent the adoption of reuse practices, related to three of the categories mentioned in RQ2 and RQ5 (see Fig. 6, left):

**Organizational.** 2 respondents reported organizational challenges of different type:

- Immature organization: *If the organization is not mature, they do not put that much effort on requirements and they think that the effort is not worthwhile* {S10}.
- Siloed projects: *They don't work enough together between projects because they have too many projects at the same time* {S01}.
- Focus on short-term benefits only (and *reusing requirements produces benefits only in the long term* {S01}).
- Cost of licenses of the tools required to implement a reuse infrastructure: *without having Jira free of cost, we would have not tried* {S10}.

**Project-related.** 9 respondents informed about project-related challenges responding to two causes:

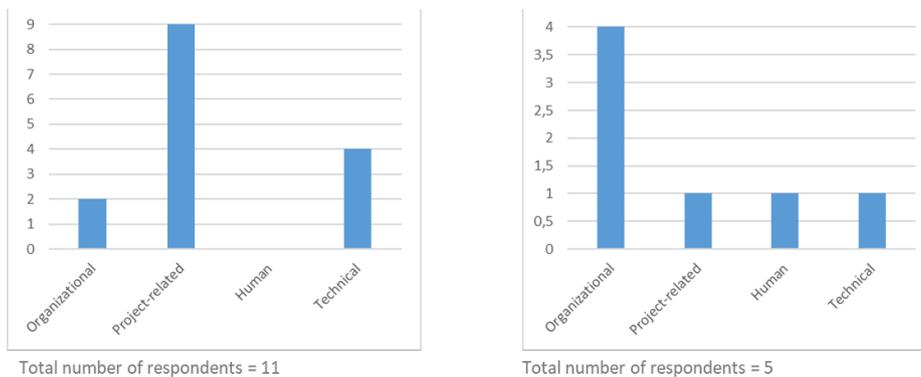
- Developing a new type of system (5 responses): reuse is not applicable in a *totally new platform; there was anything before, it was totally blank* {S01}.
- Developing a system in a new domain (5 responses): it can be related to the application domain but also *a new domain for the requirements engineer* {S24}, meaning that she has difficulties on reusing from a domain she does not master.

We found a strong correlation (p-value = 0.028; V value = 0.547) showing that most of the respondents having project-related reasons to not reuse requirements (8 out 9 people) are not reusing requirements now.

**Technical.** 4 respondents mentioned as technical challenges:

- Difficulty to access previous requirement specifications (3 respondents): in the extreme case, *no requirements were available of the old system to be replaced* {S13}.
- Lack of agreed standards: this causes that *everyone writes requirements in a different way [...] so it becomes really hard to take over somebody else requirements* {S01}.
- Low support from the requirements management tool: the tool their used was not used *for specifying the requirements of the system; it is used for other systems* {S02}.

On the other hand, 5 respondents communicated obstacles that made reuse challenging but still possible (see **Fig. 6**, right). The only category mentioned by more of one respondent was organizational, and among them, low consideration of RE in the company was cited by 3 respondents (*not too much attention is put into requirements* {S08}). Interestingly, {S03} stated as obstacle that *reusing is not fun (it will cut the creativity part of the projects) [...] people likes to do new things*.



**Fig. 6.** Challenges to overcome in requirements reuse: preventing (left) or interfering (right).

## 6 Discussion

**Observation 1.** *Requirement reuse is part of the requirements engineer toolbox.* RQ1 reports that a slight majority of respondents have reused requirements either in the project selected for this study or in the past. This observation aligns well with previous

studies in the field [1][22][23]. Putting this fact together with the factors reported as barriers or incentives to reuse, we may conclude that reuse shall be considered not as a universal principle that needs to be always pursued, but as yet-another-technique in the requirements engineer toolbox. For instance, the benefits of reuse in elicitation reported by several respondents point out that reuse can be viewed as an additional technique supporting requirements elicitation, as interviews, focus groups and others, with its own selection criteria [33]. Adding to this observation, the significantly statistical correlation informed in Section 5.1 (use in embedded systems) and Section 5.5 (use in agile projects) may hint that reuse may be more useful for some kind of projects than others.

**Observation 2.** *Requirement engineers have a positive perception about software reuse.* Even not being used in all projects, respondents clearly consider benefits outweighing challenges. Only one respondent was absolutely skeptical about requirements reuse. The difference among this positive consideration and the real extent and level of reuse reported in the study has to be attributed as said in Observation 1: reuse is not for every project but when contextual conditions apply, it will bring benefits.

**Observation 3.** *Organizational and project-related factors are determinant to reuse, over human and technical factors.* Connecting with Observation 1 again, it is clear that requirements reuse is hindered by new types of systems and new domains, although the fact that non-functional requirements have been identified as the most reusable artefact opens the door for their reuse at least at a high abstraction level even if this situation. About human factors, they are recognized to be capital for RE as a discipline, given the central role that stakeholders play. However, the study shows that this is not the case when it comes to requirements reuse. In fact, the only factor that is mentioned by some is the requirements engineer attitude towards adopting reuse and positive impact on human communication.

**Observation 4.** *Requirements reuse is still implemented in a very simple way.* As the answer to RQ4 shows, copy-and-paste based solutions are largely dominant as reuse technique, which fully aligns to our previous questionnaire-based study [2]. This dominance has a negative effect in the reuse level, as indicated by a statistically significant correlation (p-value = 0.008; V value = 0.807) that we found: most of the respondents using this approach to reuse report Low or Medium level of reuse. The main reason behind the dominance of copy-and-paste solutions can be the absence of tool support and the lack of well-established methods in the organization. The first candidate reason is supported by another significant correlation (p-value = 0.002; V value = 0.632): all respondents using catalogues for requirements reuse have tool support (i.e., the catalogue is not a separate, ad-hoc instrument but it is integrated somehow in the requirements management tool).

Comparing to the related work reported in Section 3, we can observe how some findings in our study are aligned with their observations (reuse level and benefits), but not all of them (neither obstacles nor adoption influencing factors):

- The level of reuse found in this study (14 respondents reusing in their selected projects, i.e. 58%) is in the lower range of the interval [59%-72%] reported in [22][23].

- Related work mentions three benefits. One of them, efficiency in the RE process, is the main benefit uncovered by our study. Another benefit, reduced project costs, is directly related to the decrease in cost identified as second main benefit in our study. Instead, the third benefit, better understanding of requirements, is only marginally mentioned by our respondents.
- The two obstacles identified in the related work are just marginally mentioned in our study. Instead, the main challenges and obstacles that we report in this paper are not identified in the previous work.
- Last, the existence of tool support was not a major influencing factor in our study.

To finalize this discussion, we compare the observations reported in our previous questionnaire-based study [1] with those uncovered in this current study. For each observation in [1] related to reuse, we assign as value: if the current study is fully aligned (FA), partially aligned (PA), misaligned (MA) or does not provide any evidence in this direction (NE). It is worth to mention that none of the 8 findings in [1] is contradicted in this study, but on the other side, only 2 of them are fully endorsed by ours, with 3 others partially aligned and 3 for which we did not get any evidence. This last statement, together with the slight misalignments with related work mentioned above, calls for more empirical studies to gather more evidence to build a theory on requirements reuse.

**Table 3.** Comparison of this study and the previous questionnaire-based study reported in [1]

Observations in the questionnaire-based study	This study
A Significant Percentage of the Respondents Practice Requirements Reuse	PA
The Level of Requirements Reuse is Usually Low	PA
Participants of Larger Organizations Declare a Higher Level of Reuse	NE
Requirements Reuse Techniques Most Commonly Used are those Based on Textual Copy and Subsequent Modification of Requirements from Previous Projects	FA
There is a Correlation Between the Level of Requirements Reuse and the Requirements Reuse Techniques Used	PA
Organizations with more Established Software Processes and Methods are the ones that Declare a Higher Level of Requirements Reuse	NE
NFRs are More Likely to be Similar or Recurrent Among Projects	FA
Ignorance of Reuse Techniques and Processes is the Main Reason for the Lack of Reuse Adoption	NE

## 7 Conclusions and Future Work

In this study, we have presented an empirical study based on interviews with practitioners on the adoption level of requirements reuse and the practices, benefits and challenges related. We responded six research questions, which results are:

- **RQ1:** Moderate adoption of reuse practices by practitioners
- **RQ2:** Prevalence of organizational and project-related factors influencing reuse adoption.
- **RQ3:** Non-functional requirements as the type of requirement more prone to reuse.
- **RQ4:** Find-copy-paste-adapt the most popular approach to implement reuse.

- **RQ5:** Prevalence of organizational and project-related benefits stemming from reuse adoption.
- **RQ6:** Prevalence of project-related challenges to overcome when reusing.

In addition to giving answer to the research question, we found some interesting insights. We found that requirement reuse is quite a common practice for requirements engineers and that they have a positive perception of reuse, since it usually leads to benefits. However, the determinant factors to reuse are associated to organizational and project-related issues, and requirements reuse is still implemented in a very simple way.

Finally, we compare our results with the related work and a previous questionnaire-based study. In both cases, we can observe how some results from our study are aligned with their observations, but not all of them (mainly because we did not get enough evidence to consider it a result as such).

Future work moves on the direction of replicating these studies in other settings in order to find further evidence that may help in establishing a theory on the topic.

## Acknowledgements

This work has been partially funded by the Horizon 2020 project OpenReq, which is supported by the European Union under the Grant Nr. 732463.

## References

1. Palomares, C., Quer, C., Franch, X.: Requirements Reuse and Requirement Patterns: A State of the Practice Survey. *Empirical Software Engineering* 22(6), 2017.
2. Irshad, M., Petersen, K., Poulding, S.: A Systematic Literature Review of Software Requirements Reuse Approaches. *Information and Software Technology* 93, 2018.
3. Carrillo-de-Gea, J. M. et al.: Reusing Requirements in Global Software Engineering. Chapter in *Managing requirements knowledge*, Springer, 2013.
4. Pacheco, C., Garcia, I., Calvo-Manzano, J. A., Arcilla, M.: Reusing Functional Software Requirements in Small-sized Software Enterprises: A Model Oriented to the Catalog of Requirements. *Requirements Engineering Journal* 22(2), 2017.
5. Haeng-Kon, K.: Effective Domain Modeling for Mobile Business AHMS (Adaptive Human Management Systems) Requirements. SNPD 2014.
6. Veeda, R., Cysneiros, L.M.: Towards a Tool to Help Exploring Existing Non-functional Requirements Solution Patterns. REW 2017.
7. Chung, L., Supakkul, S.: Capturing and Reusing Functional and Non-functional Requirements Knowledge: A Goal-object Pattern Approach. IRI 2006.
8. Kundi, M., Chitchyan, R.: Use Case Elicitation with FrameNet Frames. REW 2017.
9. Salini, P., Kanmani, S.: A Knowledge-oriented Approach to Security Requirements for an E-voting System. *International Journal of Computer Applications* 49(11), 2012.
10. de Brock, B.: Towards Pattern-Driven Requirements Engineering: Development Patterns for Functional Requirements. MoDRE 2018.
11. Franch, X., Quer, C., Guerlain, C., Renault, S., Palomares, C.: Constructing and Using Software Requirement Patterns. Chapter in *Managing Requirements Knowledge*, Springer, 2013.

12. Renault, S., Méndez-Bonilla, O., Franch, X., Quer, C.: PABRE: Pattern-Based Requirements Elicitation. RCIS 2009.
13. Panis, M. C.: Reuse of Architecturally Derived Standards Requirements. RE 2015.
14. Darimont, R., Zhao, W., Ponsard, C., Michot, A.: Deploying a Template and Pattern Library for Improved Reuse of Requirements Across Projects. RE 2017.
15. Srivastava, S.: A Repository of Software Requirement Patterns for Online Examination System. *International Journal of Computer Science* 10(3), 2013.
16. Chen, X., Han, L., Liu, J., Sun, H.: Using Safety Requirement Patterns to Elicit Requirements for Railway Interlocking Systems. REW 2016.
17. Knote, R., Söllner, M., Leimeister, J.M.: Towards Requirement Patterns for Smart Physical Work Assistants. REW 2017.
18. Niu, N., Savolainen, J., Niu, Z., Jin, M., Cheng, J.R.C.: A Systems Approach to Product Line Requirements Reuse. *IEEE Systems Journal* 8(3), 2014.
19. Chiang, C.C., Neubart, D.: Constructing Reusable Specifications through Analogy. SAC 1999.
20. Bonilla, B., Crespo, S., Clunie, C.: Reuse of Use Cases Diagrams: An Approach based on Ontologies and Semantic Web Technologies. *Int. Journal in Computer Science*, 9(1), 2012.
21. Carvalho, R. M., Andrade, R. M. C., Oliveira, K. M., Kolski,.: Catalog of Invisibility Requirements for UbiComp and IoT Applications. RE 2018.
22. Chernak, Y.: Requirements Reuse: The State of the Practice. SWSTE 2012.
23. Bakar, N.H., Kasirun, Z.M.: Exploring Software Practitioners Perceptions and Experience in Requirements Reuse: An Empirical Study in Malaysia. *International Journal on Software Engineering Technologies* 1(2), 2014.
24. Baig, J. J. A., Al Fadel, M. A.: Measuring Reusability during Requirement Engineering of an ERP Implementation. ICICIS 2017.
25. Palomares, C., Franch, X., Quer, C.: Industrial Practices on Requirements Reuse: An Interview-based Study – Research Protocol. Available at <http://tiny.cc/reuse-protocol>.
26. Oates, B.J.: *Researching Information Systems and Computing*. SAGE Publications, 2006.
27. Saldana, J.: *The Coding Manual for Qualitative Research*. SAGE Publications, 2009.
28. Field, A.: *Discovering Statistics using SPSS*. SAGE Publications, 2009.
29. Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum Associates, 1988.
30. Runeson, P., Höst, M.: Guidelines for Conducting and Reporting Case Study Research in Software Engineering. *Empirical Software Engineering* 14(2), 2009.
31. Robson, C.: *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*, Blackwell Publishers Inc., 2002.
32. Franch, X., Palomares, C., Quer, C.: Industrial Practices on Requirements Reuse: An Interview-based Study – Coding Results. Available at <http://tiny.cc/reuse-replication-package>.
33. Carrizo, D., Dieste, O., Juristo, N.: Systematizing Requirements Elicitation Technique Selection. *Information & Software Technology* 56(6), 2014.