



The state-of-practice in requirements specification: an extended interview study at 12 companies

Xavier Franch¹ · Cristina Palomares² · Carme Quer³ · Panagiota Chatzipetrou⁴ · Tony Gorschek^{5,6}

Received: 8 April 2022 / Accepted: 15 February 2023
© The Author(s) 2023

Abstract

Requirements specification is a core activity in the requirements engineering phase of a software development project. Researchers have contributed extensively to the field of requirements specification, but the extent to which their proposals have been adopted in practice remains unclear. We gathered evidence about the state of practice in requirements specification by focussing on the artefacts used in this activity, the application of templates or guidelines, how requirements are structured in the specification document, what tools practitioners use to specify requirements, and what challenges they face. We conducted an interview-based survey study involving 24 practitioners from 12 different Swedish IT companies. We recorded the interviews and analysed these recordings, primarily by using qualitative methods. Natural language constitutes the main specification artefact but is usually accompanied by some other type of instrument. Most requirements specifications use templates or guidelines, although they seldom follow any fixed standard. Requirements are always structured in the document according to the main functionalities of the system or to project areas or system parts. Different types of tools, including MS Office tools, are used, either individually or combined, in the compilation of requirements specifications. We also note that challenges related to the use of natural language (dealing with ambiguity, inconsistency, and incompleteness) are the most frequent challenges that practitioners face in the compilation of requirements specifications. These findings are contextualized in terms of demographic factors related to the individual interviewees, the organization they are affiliated with, and the project they selected to discuss during our interviews. A number of our findings have been previously reported in related studies. These findings show that, in spite of the large number of notations, models and tools proposed from academia for improving requirements specification, practitioners still mainly rely on plain natural language and general-purpose tool support. We expect more empirical studies in this area in order to better understand the reason of this low adoption of research results.

Keywords Requirements engineering · Requirements specification · Requirements documentation · Natural language requirements · Requirements management tools · Empirical studies · Interviews

✉ Xavier Franch
franch@essi.upc.edu

Cristina Palomares
cpalomares@essi.upc.edu

Carme Quer
cquer@essi.upc.edu

Panagiota Chatzipetrou
panagiota.chatzipetrou@oru.se

Tony Gorschek
tony.gorschek@bth.se

Num.1–3, Omega Building, Room S206, Barcelona, Catalonia, Spain

³ Software and Service Engineering Group (GESSI), Universitat Politècnica de Catalunya (UPC), Jordi Girona Num.1–3, Omega Building, Room 124, Barcelona, Catalonia, Spain

⁴ Department of Informatics, School of Business, Örebro University, Örebro, Sweden

⁵ Software Research Engineering Lab (SERL), Blekinge Institute of Technology, Karlskrona, Sweden

⁶ Fortiss GmbH, Munich, Germany

¹ Software and Service Engineering Group (GESSI), Universitat Politècnica de Catalunya (UPC), Jordi Girona Num.1–3, Omega Building, Room 122, Barcelona, Catalonia, Spain

² Software and Service Engineering Group (GESSI), Universitat Politècnica de Catalunya (UPC), Jordi Girona

1 Introduction

Requirements specification, also known as *requirements documentation* [30, 40], refers to the description of the system requirements that are elicited and negotiated during the early stages of the requirements engineering (RE) process.

The primary goal of requirements specification is to document the elicited system requirements. To this purpose, requirements are often compiled into a Software Requirements Specification (SRS) document [19]. However, requirements specifications are sometimes merely stored in a requirements management tool (or in another type of tool adapted to play this role, for example, a spreadsheet). SRSs are also known by alternative names, including *business requirements documents*, *product specifications*, *system specifications*, or simply *requirements documents*. Throughout the existence of an SRS, many stakeholders, including customers, marketing and sales departments, project managers, product owners, software developers, testers, and maintenance and support staff, rely on the content of this document to conduct their work.

The requirements that are included in an SRS can be specified in different ways [16, 30]. Evidence shows that natural language (NL) is the most dominant means to express requirements [9], either in the form of free text or by following templates, for example, the Volere Requirements Specification Template [34] or the Easy Approach to Requirements Specification (EARS) [25]. While it is the easiest option for practitioners to choose, requirements that are written in NL suffer from a range of different problems, including ambiguity, inconsistency, and incompleteness [5, 15]. Therefore, several other notational systems have been proposed. Visual models, including class diagrams, statechart diagrams, and process models, have also become popular complements to NL specifications (for example, class diagrams that describe data requirements more precisely). Other visual notation systems are popular in the research community but have not been widely adopted by industry, for example, the goal models known as i^* [42] and KAOS [7]. Formal specifications based on a form of logical specification language have also been used, but only in very specific contexts, such as safety-critical systems [24]. Given this variety of approaches to requirements specification, a natural follow-up question is whether they have been adopted in industry, under what circumstances and facing which challenges. This question links to a number of recent papers in this direction both in the specific area of RE [13] or the broader discipline of software engineering [14].

In this article, we report on the results of an interview-based, empirical study involving 24 experienced senior practitioners working in 12 IT companies. Our study thus

presents how requirements specification practices are performed in industry. We directly address five research questions regarding the specification artefacts (and associated languages) that are used by practitioners, the templates and guidelines that they follow, the classification schemas that they employ to organize requirements, the tools that are used in the specification process, and the challenges that practitioners face in the documentation of requirements specifications. We also present an analysis of these results and compare them with the findings of previous research into the documentation of requirements specifications.

This paper is structured as follows: In Sect. 2, we present a summary of previous research related to the present study. Section 3 describes the research methodology that is used in this study. In Sect. 4, we inform the reader of the results of our interview study, while Sect. 5 presents our analysis of these results. Finally, in Sect. 6, we discuss our conclusions and provide recommendations for future work.

2 Related work

We identified 15 relevant empirical studies published after 2010 that report findings on the state of the practice regarding requirements specification (see Table 4, Appendix 1). We summarize the characteristics of these studies as follows:

- They are based on three types of empirical study: questionnaires, interviews, and systematic literature reviews. Several of the studies can be said to be multi-instrument studies.
- Almost half of the studies that we identified (seven studies) were conducted in one country, while five of the studies included participants from more than four countries. Remarkably, six of the seven single-country studies were the oldest on our list.
- Most of the studies that we examined are not focussed on requirements specification only but also address a broader RE scope.
- Some of the studies are focussed on particular types of projects or systems, namely Agile projects, safety-related systems, embedded systems, or outsourced development environments.
- Only two of the studies focus on companies within a specific domain, namely the automotive domain and the domain of nuclear energy systems.
- A number of the studies address specific types of requirements, including security requirements or, more broadly, quality requirements (QR).
- Two of the studies are longitudinal studies. One study (the NAPIRE project) was reported on in a series of papers, of which, we selected three.

- Two of the papers report on different studies that were conducted with the same participants.

In Table 5 (Appendix 1), we summarize the results of each of the papers that we deemed relevant to our study. The results of these papers are related to the five research questions that we present in Sect. 3.1.

Eleven of the papers included in our study address different aspects of the representation of requirements. Six of the studies state that the majority of the participants who were included in these studies use NL representation [1, 20, 28, 32, 38, 39]. It was reported in five studies that UML models were used, with one third to 57% of the participants reporting that they employed such models [8, 20, 22, 28, 39]. Elahi et al. [8] and Sikora et al. [38] report on the wide use of models in the context of security requirements and embedded systems, respectively. Hotomski et al. [17] find that free textual representations are most frequently used by companies that follow a Waterfall approach, while companies that follow an Agile approach are more likely to employ user stories. Finally, three studies report on results that are specific to QRs: Berntsson-Svensson et al. [3] and Wagner et al. [39] state that QRs were specified in a quantifiable manner by more than half of their subjects, and Alsaqaf et al. [2] observe that QRs were specified as related to user stories and that the precision of the specification of QRs is project-dependant.

Four of the papers that we examined discuss templates and guidelines in their investigation of standards. Elahi et al. [8], who distributed a questionnaire to practitioners working on security requirements, concluded that 73% of their participants used standards or templates. Wagner et al. [39], on the other hand, reported that almost all of their respondents employ an RE internal company standard. They note that, in many organizations, the standard is tailored at the beginning of a project, and they claim that external normative standards are often considered to be too complex and elaborate to apply. Palomares et al. [28] reported on the use of templates in the specification of requirements in the context of reuse, which was not as popular as the more basic technique of copying and pasting requirements. Franch et al. [12] observe that standards in RE, in general, are infrequently used. However, the specification of standards in RE did take place with a slightly higher frequency.

Only two of the papers included in our study report on requirements structure. Raatikainen et al. [32] note that although requirements are logically layered, hierarchical structures are often not made explicit in the SRS. More recently, Ågren et al. [1] have made the opposite observation, perhaps justified by the domain of their study. These

scholars note that because of the complexity of automotive systems, requirements are generally organized in a number of abstraction levels that are dealt with by different roles within the organization. These circumstances increase the complexity of their management.

Four papers addressed the use of tools in requirements specification. Three of the papers report on the use of general-purpose, Hotomski et al. [17], Kassab et al. [20] and Liu et al. [22] discuss the use of document management tools and project management tools that are used in the context of requirements specification, respectively. On a similar theme, Liu et al. [22] note that tools relevant to the specification of requirements are more accessible and available in multi-national corporations, even though more than a half of the participants in their study reported that they did not use any specific tools for the specification of requirements.

Finally, 13 of the papers included in our study address a number of challenges related to requirements specification. The challenges that these studies identified are very diverse and can be related to the following issues: the under-specification of requirements or related knowledge [6, 8, 17, 18, 21, 22, 39] the over-specification of requirements [1, 6], limited use of models because they are challenging to use [2, 38] the inclusion of different layers of abstraction in specifications [21, 38] problems in specifications depending on the individual who wrote them [1, 17] management and use of tools [32], insufficient resources for the proper management of requirement specifications [21], and a lack of traceability [4, 32]. In terms of specific concepts, several studies reported challenges with regard to the specification of user stories [2, 4, 18] and non-functional (or quality) requirements [2, 4, 22].

We refer to the results of the above-mentioned studies when we contextualize our own findings in the present study.

3 Research methodology

We conducted in-depth semi-structured interviews in our study, therefore adopting a qualitative research perspective [35]. Qualitative research is appropriate when the purpose of an investigation is to explore a particular area of interest or to improve our understanding of a phenomenon [35, 36], as it is the case of our study. In order to contextualize this paper, it is necessary to mention that our study

Table 1 The research goal and the research questions that are raised in the present study

Goal	To identify how requirements are specified
RQ1	What artefacts are used for specifying requirements?
RQ2	What templates and guidelines, if any, are followed to specify requirements?
RQ3	What classification schemas, if any, are used for organizing requirement specifications?
RQ4	What tools, if any, are used in the specification process?
RQ5	What challenges, if any, are faced in the specification process?

focuses on several aspects of RE, each one presented in a different paper: understanding of the role of requirements engineer elicitation [11], requirements elicitation [29], requirements reuse [10] with emphasis on requirement patterns (following the approach by Renault et al. [33]), and requirements specification, which is the focus of the current paper. We provide the protocol that we used during this study in a separate document that is available as supplementary material.¹ Consequently, we report on just the most relevant facts regarding the protocol in this paper. We refer the interested reader to this separate document for additional details.

3.1 Summary of the protocol

3.1.1 Research questions

The Research Questions (RQs) that informed our study are presented in Table 1. The first RQ is *What artefacts are used for specifying requirements?* We use this question to interrogate the instruments and notation systems that are used in industry to specify requirements. In particular, we want to examine the dominance of natural language (NL) in industrial practices as reported in previous studies and identify what other options are employed by practitioners. The second RQ is *What templates and guidelines, if any, are followed to specify requirements?* This question is aimed at investigating the adherence to specification practices that are widespread in the RE field. This RQ reveals whether practitioners possess established knowledge that they may apply to their projects and whether they find this established knowledge useful. Our next question, RQ3, is *What classification schemas, if any, are used for organizing requirement specifications?* This question identifies the concepts that are used by practitioners when they organize their specifications, for example, ‘types of requirements’ and ‘domain concepts’.

¹ <https://www.upc.edu/gessi/RE/Protocol-Study-RE-state-of-the-practice.pdf>. We cannot disclose the data gathered from the interviews due to confidentiality reasons.

Given that the specification activity needs to produce a tangible outcome, namely the SRS, our fourth RQ is *What tools, if any, are used in the specification process?*. Finally, RQ5 is *What challenges, if any, are faced in the specification process?* This question is aimed at (i) identifying the challenges that practitioners are confronted with when specifying requirements and (ii) assessing their perceived severity.

3.1.2 Sampling

The target population of our study included practitioners in charge of RE activities in several software development projects at different Swedish companies. We took care to ensure variation in our sample population in terms of size, domain, and business area. We asked the companies that we selected to nominate two employees. However, one company nominated only one employee, and another company nominated three employees. In summary, we conducted 24 interviews at 12 companies (see Sect. 4.1).

3.1.3 Procedure and instruments

We designed and piloted a semi-structured interview guide by following the guidelines provided by Oates [27]. We asked each participant to focus on a single project, while they answered our interview questions so as to facilitate our interpretation and assessment of contextual information. We recorded the face-to-face interviews for later reference. The average duration of each interview was approximately 2 h in length, of which approximately 10 min were used to describe the company and/or project and 40 min were used to discuss requirements specification.² We did not provide consent forms to the interviewees, nor they requested them. Subjects were informed about confidentiality of their responses only informally. All participants were known to the Swedish team of authors and this made consent forms unnecessary in this study.

To mitigate possible bias in the responses that could be introduced by the specific projects that our interviewees were involved in, we followed the recommendations by Lutters and Seaman [23] and Patton [31] and thus included questions such as *Is this typically how you do this? If not, how do you usually do it?* in our interview guide. This approach enabled us to (i) compile a thick description of the requirements specification processes that was performed by the interviewees and (ii) allow their opinions to emerge during the interviews. The interview guide is available in Appendix 1 of the protocol document.

² The rest of the time spent during the interviews addressed other parts of our study that are not reported in this paper.

3.1.4 Data analysis procedure

The interviews were recorded for subsequent analysis, and notes were taken by the interviewer both during and after the interviews, with the main answers provided by the interviewees, using a template designed for that purpose. The result of this triangulation procedure was the basis for coding, according to Saldana [37].

We used multiple coding techniques with the support of the Atlas.ti³ tool, as detailed in our protocol document. Most remarkably, we configured an initial list of codes from the results of a previous survey, which was further elaborated with the information gathered in the interviews. Two of the authors led the process working in pairs, and then discussed with the whole research team the final consolidation of codes and categories through dedicated meetings. Consensus in the team was usual, although one particular research question demanded special attention, namely RQ4. The reason was that grouping tools into categories was challenging (except for the case of Office tools), because categories had at the end some overlapping functionalities. We report the frequencies of these codes as a way to quickly visualize the most followed practices in relation to every research question.

One of the main objectives of our study is to identify both demographic factors that may influence the answers to the research questions and dependences among these answers. As a strategy to facilitate the identification of such associations, we analysed the data using statistic tests. More precisely, we employed the Chi-Squared test to suggest non-causal associations and Cramer's V test to measure the strength of the suggested association. We remark that the purpose of such statistical analysis was purely explorative: it was useful to help identifying which associations among demographic factors and interviewees' answers have higher chances to represent a real relationship, but we did not assume that statistical significance implied real existence of an association. Furthermore, in every facet of each question, we have inquired those relationships that we thought, building upon our expert knowledge on RE, could eventually be observed in the data. In other words, we have not limited our analysis to those identified by the Chi-Squared test.

3.2 Validity

This section outlines the threats to the validity of our study. To this aim, we follow the categories suggested by Wohlin et al. [41]. We also include a discussion of complementary strategies that we used to mitigate these threats. We refer the reader to the protocol document for a more complete discussion of these points.

- **Construct validity** Threats to the construct validity were associated with differences in the terminology that was used across the different interviews. These threats were addressed by (1) piloting the interview guide before the research interviews were conducted, (2) asking clarification questions regarding terminology during the interviews when needed, and (3) applying multiple codes to the same statement so as to capture multiple interpretations. Secondly, both in the interview guide and during the actual interviews, the participants were made aware that the information that they provided would be confidential, made anonymous, and aggregated with the other interviews. Under such conditions, the participants shared their experiences and perceptions freely. Last, the use of a statistical test to suggest candidate associations among demographic factors and interviewees' responses entails two threats: (1) statistical-based exploration may have highlighted spurious associations while hiding actual ones; (2) the data are not well-suited to statistical analysis mainly due to small and convenient sample. To mitigate this threat, as reported in Sect. 3.1, we used statistical analysis subordinated to qualitative analysis and discarded every possible association that had not an evident rationale behind.
- **Conclusion validity** Threats to the validity of our conclusions were based on the possible bias that we introduced during our coding steps. These threats were mitigated by (1) preserving traceability from the raw data to the coding categories and (2) using different types of triangulation methods (i.e. theory triangulation and researcher triangulation).
- **Internal validity** A threat to the internal validity of our study was related to the fact that the interviews that we conducted were each based on a single software development project. To mitigate the consequences of this limitation, we sent out the interview guide in advance to the study's participants. They were also informed that the study was not focussed on analysing 'incorrect' RE practices, thereby minimizing the risk that they would choose to discuss successful projects only. Another threat to the internal validity of this study is posed by the fact that the interviews were recorded but not transcribed. This threat was mitigated by importing the audio recording of the interviews into a qualitative data analysis tool (i.e. Atlas.ti).
- **External validity** To strengthen the external validity of our study, we took two measures: (1) we employed a combination of convenience sampling and maximum variation sampling in our selection of the companies [35], and (2) we granted the interviewees the freedom to choose the project that they wished to talk about during the interview. Notwithstanding this, as in any other qualitative study, generalization is not the main goal of

³ <http://atlasti.com/>.

our study. Instead, we are concerned with characterizing, explaining, and understanding a phenomenon as it exists in a particular context [35].

4 Results

In Sect. 4.1, we describe the individuals who participated in the interviews, their companies, and the projects that they chose to talk about. In Sect. 4.1–4.1, we present each of our research questions. We first summarize our primary findings for each research question and then justify them by referring to the codes that emerged in the content analysis, quantitative analysis, and the interviewees' quotes. A complete list of the coding obtained from the interviews' answers can be found in Appendix 3.

4.1 Demographics

In this section, we discuss the most relevant features of (1) the subjects who participated in the interviews, (2) their companies, and (3) the projects they chose to talk about during the interviews (see Appendix 2).

4.1.1 The interview subjects

All 24 interview subjects played the role of requirements engineer at least in the chosen projects, even if only one of them has exactly this job title; we may find other job titles such as “requirements analyst” or “business analyst”, which are often used with the same meaning [11, 40]. A summary of other characteristics follows:

- Educational background. Mostly related to computer science, information systems, and software engineering, through BSc or MSc degrees. Still, a number of subjects hold degrees in other areas such as chemical engineering, civil engineering, telecommunications or robotics.
- Working experience. Ranging between 3 and 25 years in industry (16.2 years on average) and between 0 and 15 years working at a university or in a research laboratory (3.2 years on average).
- Position. Ranging from a subject new to his position and even to the company, to 15 years in the current position (and 20 to the company).
- Relation to RE. In spite of the concrete position they had in their companies, they all were involved in RE activities in the project they chose for the interview.

4.1.2 Companies

The twelve companies participating in our study belong to one of the three following categories, depending upon the final customer:

- Software consultancy company (SCC). Performing a variety of software development tasks for different clients.
- IT department (ITD). Performing or outsourcing a variety of software development tasks in the context of an organization
- Software house (SH). Developing and commercializing software systems and services.

Appendix 2 also shows business domain of the company, as well as their public or private nature (being the latter the dominant case).

4.1.3 Projects

The projects chosen by the interviewees were certainly diverse in terms of domain, duration, team size and development approach:

- Domain. Mainly embedded systems, websites, mobile applications, and customer business support operations.
- Duration. From four months to around ten years to complete.
- Team size. Varying from two individuals up to thousands of people (although tree subjects did not know this particular characteristic).
- Development approach. About two thirds of the projects were developed following a Waterfall-like approach, with the rest being Agile.

4.2 RQ1: what artefacts are used for specifying requirements?

The use of unrestricted NL is the option preferred by practitioners to specify requirements. NL sentences are often annotated with graphical elements or models and can be used as a resource to produce more elaborate textual artefacts, including use cases and user stories. Textual artefacts, graphical elements, and models are used to a similar extent. Mockups and diagrams are the most frequently used graphical elements. No particular type of model (not even class diagrams) was widely used, but UML is dominant as a language.

Figure 1 shows the frequency with which the interview subjects used *NL sentences*, *Textual artefacts*, *Graphical*

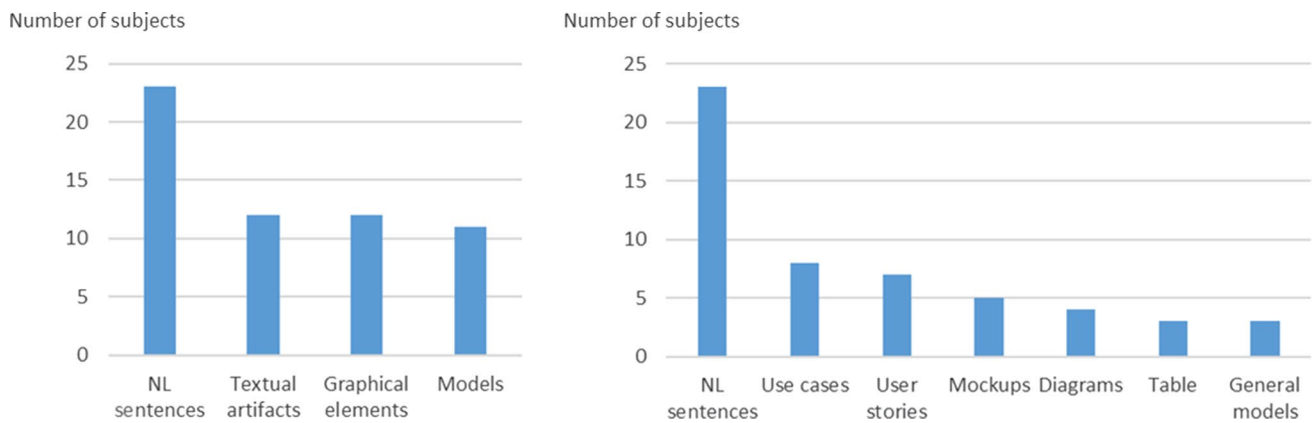


Fig. 1 An overview of the requirements specification artefacts that were used by the interviewees. The graph on the left depicts the main artefact categories that were used, and the graph on the right indicates the frequency with which these artefacts were used

elements, and *Models*. We may observe that, in spite of the well-known problems associated with plain NL (namely, ambiguity, incompleteness, and inaccuracy), the low (virtually non-existent) communication effort that is implied by unrestricted *NL sentences* causes them to be the preferred option that was used in the projects that were reported on. In fact, NL sentences were used by every interviewee except one. With respect to NL sentences, we observed the following:

- Four interview subjects (17%) stated that they used NL sentences only in their requirements specifications. These participants all mentioned the use of attributes “like name, priority, description, owner, rational, target, verification methods and attachments” (S14(H)⁴) as complementing their work, primarily for the purpose of traceability.
- Nine interviewees (38%) reported that they primarily used NL sentences in conjunction with some other language(s) as a form of annotation. Four interviewees used graphical elements only, for annotation purposes. For example, S23(L) used “sentences in NL and also some tables of requirements, as well as diagrams for the shape of the tunnel [a domain object]”. However, the other five interviewees mentioned that they used some kind of models (state diagrams or UML metamodels) as annotations. The most complex scenario was described by S2(A), who reported that they used business process models, mockups, and use cases, each with a particular

annotation purpose (for example, process models were used to describe a navigational flow).

- Five other interviewees (21%) reported that they used NL sentences as a starting point and developed them into textual artefacts “for communicating the needs with the client” S3(B). In several cases, this development was either more complex, for example, S10(F) “converted these [NL] requirements into user stories, and then into use cases and sequence diagrams”. However, this type of elaborate development did not always take place. In some cases, a contrasting approach was adopted, for example, by S22(K), who kept the more technical requirements in NL “more like a state of compliance”.
- Three interviewees (13%) reported that they used NL sentences for specifying quality (non-functional⁵) requirements. S1(A) and S13(G) used these requirements to complement use cases, while S4(B) used several other languages for functional requirements (which were derived from NL, so they are included in the category above).
- Finally, four interviewees (17%) did not draw a clear line between the use of NL sentences and other types of languages (mainly textual artefacts). However, S18(J) reported on using a UML diagram.

The only interview participant who did not report on using NL sentences as a specification language was S21(K). This practitioner stated that they had a highly elaborated specification structure in place with “user stories, even for non-functional requirements; wireframes or mockups

⁴ The notation used in this article to refer to the interview subjects corresponds to the scheme: Sx(y), where Sx is a unique identifying number for the interviewee (see Table 6 in Appendix 2) and y is the identifying number assigned to the interviewee’s company (see Table 7 in Appendix 2).

⁵ Interviewees used both terms synonymously. We will use the term *quality requirements* (QR) throughout this paper, except when we quote our interviewees verbatim.

showing how users interact with the system; and some diagrams for the [programming] interfaces too”.

The other three categories shown in the graph on the left of Fig. 1 are more evenly distributed since they were used by approximately half of the interviewees. Two *Textual artefacts* were dominant, namely *Use cases* and *User stories*, with a similar proportion of interviewees using them, i.e. 33 and 29%, respectively. One interviewee also mentioned *Test cases* as a specification artefact. Thus, a total of 12 subjects (50%) reported on using textual artefacts. Note too that some interviewees used more than one type of textual artefact in their projects for different purposes. In fact, after NL sentences, use cases and user stories were the most frequently used artefact by our participants (see Fig. 1, right), both belonging to the *Textual artefact* category. The interviewees remarked on the well-known limitations of these two types of artefact when the practitioner moves beyond the specification of functional requirements. Remarkably, as already mentioned above, “requirements that could not be specified with use cases were stated as natural language requirements as annotations of the use cases, in the ‘other requirements’ section” S12(G). This was particularly the case with respect to QRs and compliance conditions. However, use cases and user stories were also annotated with (1) mockups, to show the user interaction graphically; (2) diagrams, for indicating the system’s interfaces: “in the sense of communication between systems” S21(K); or (3) domain models, such as “power or capacity diagrams for the battery, temperature histograms (different diagrams at the hardware level) and state flow diagrams (at the software level)” S19(J).

The use of *Graphical elements* was widespread among our interview subjects. As many as 12 participants, i.e. 50%, reported that they used this type of artefact. The main purpose of using graphical elements was stated to provide additional information in the form of annotations or to “help to visualize the requirements; they are a more visual form of seeing the requirements” S20(J). However, there was a great deal of dispersion in the motivation as to why these artefacts were used. Specifically, it was reported that *Mockups* were the most frequently used graphical element (21%, see Fig. 1, right), followed by *Diagrams* (17%), *Tables* (13%), *Pictures* (8%), and *General elements* (4%). We were informed that, in some cases, graphical elements acted as supporting figures that clarify the NL sentences “like overview diagrams and so on” S9(E). In other cases, graphical elements took on the form of much more complex diagrams. For instance, S19(J) reported that “[t]he requirements specifications had natural language requirements, figures and tables to clarify or summarize some requirements”.

Similarly, there was a great dispersion in the choices made by the 11 subjects (46%) who declared that they had used *Models* in the requirements specification work. We note that no single type of model was used by more than

two interview subjects. Several of our interviewees (three out of the 11 who reported on using models) used more than one type of model. For example, S12(G) stated that “[we used] mainly use cases, but also sequence diagrams and some UML metamodels too”. *Domain models* were not extensively used, and only S19(J) reported on using “power or capacity diagrams for the battery, temperature histograms [different diagrams at the hardware level], and state flow diagrams [at the software level]”. The only modelling language that the interviewees mentioned was UML. Almost half of the practitioners who stated that they used models (five interviewees) reported that they specified their models with UML. For the other six interviewees, we cannot say whether they used alternative forms of notation or simply the notation was not considered important enough for them to mention it.

Finally, it is noteworthy that 20 of the 24 interviewees (83%) used more than one artefact. Two of the interviewees reported that they used up to five different artefacts and another four participants declared that they used a combined number of four artefacts.

4.3 RQ2: what templates and guidelines, if any, are followed to specify requirements?

Templates and guidelines are widely used for requirements specification, although RE standards are rarely taken into consideration since templates and guidelines are often prescribed by the organization. They are mainly used to (1) define the SRS layout, (2) declare document attributes, and (3) establish writing guidelines.

As many as 18 participants in our study (75%) reported that they follow templates or guidelines when they are specifying requirements. Five of these 18 interviewees (28%) stated to “have different templates for the different requirement documents we produce” S4(B), for example, in the form of basic requirements and use cases (for example, used by S6(D)). Some of the six subjects who did not use templates reported that they knew about the use of templates in other phases of software development (for example, during coding, testing, and API documentation) but not during RE. However, one participant, S1(A), was aware of “a lot of needs related to how to write requirements, but nothing still implemented in my company”.

The graph on the left in Fig. 2 indicates the *Provenance* of the templates or guidelines that are used in the context of requirements specification. The dominant provenance (for our interviewees) is *Organization*, with 11 subjects (i.e. 61% out of the 18 subjects who reported on using templates or guidelines) with respect to the origin of such templates or guidelines. A number of interviewees provided details

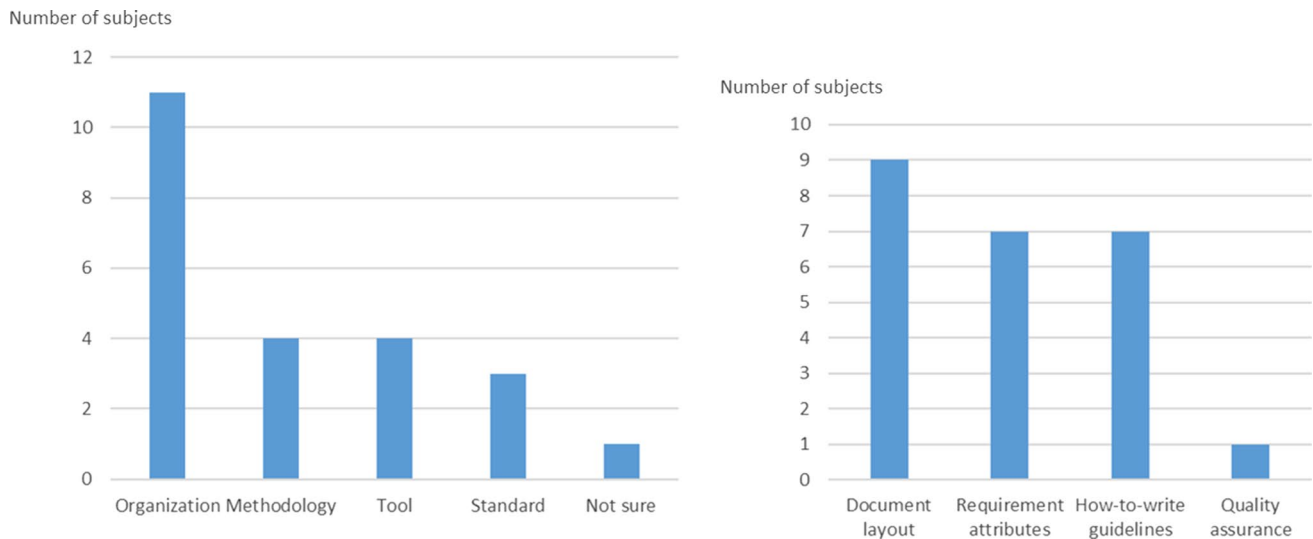


Fig. 2 An overview of the use of templates and guidelines for requirements specification: provenance (left) and purpose (right)

related to either the content of the templates or guidelines: “they usually write requirements in a certain way, by using Simplified English [which is a set of words that you are allowed to use when writing the requirements]” (S16(I)), or to the structure of the guidelines: “the template defines the attributes/parts a use case should have when it is specified” (S12(G)). The rest of provenances are somewhat dispersed, as indicated in the graph to the left in Fig. 2 shows. We note that *Methodologies* and *Tools* (4 interviewees each, 22% of 18 who mentioned the provenance of the templates and guidelines) are slightly more popular than real *Standards* (3 interviewees, 17%). In fact, standards were mentioned only vaguely, for example: “some ISO standards for documentation” (S19(J)) or in terms of “writing guidelines based on some IEEE standard” (S15(H)). In contrast, some of our participants were aware that “the company templates are not compliant to the ISO standards and so on” (S2(A)). The requirements specification process may be informed by a methodology that follows some specific proposal, e.g.: “a modified SCRUM project Excel template from Roman Pichler, an expert in SCRUM methodology” (S10(F)), but may also include general guidelines from RUP, Scrum itself, or even just SMART principles. As for tools, a practitioner referred to Mingle as source for user story templates (S6(D)). Finally, one participant (S11(F)) claimed that they were *Not sure* about the provenance of the templates or guidelines that they used.

The graph on the right in Fig. 2 shows the *Purpose* behind the use of templates and guidelines. Four of our interviewees who stated that they used templates or guidelines did not provide any information concerning the purpose of their use. Consequently, only 14 participants provided information concerning this aspect. Templates and

guidelines were used primarily to fix the *Document layout* (9 subjects, 64% of the 14 subjects), at least the headings or section titles. However, on occasion, the use of templates and guidelines was more elaborated, as explained by S6(D): “Some fields are input [first high-level requirement], clarifications, requirement breakdown, implementation sketch, impacts on verification, etc.”. The participants used the other two main purposes of templates and guidelines equally (seven subjects, 50%), including the declaration of *Requirement attributes* in the form of a title or author and the setting out of writing guidelines (*How-to-write guidelines*), e.g.: “a set of over 100 rules on how to write requirements” S23(L). One interviewee reported on using ISO-based templates “as part of the software *Quality assurance* process” S19(J).

4.4 RQ3: what classification schemas, if any, are used for organizing requirements?

Requirements are always classified, in terms of main functionalities (the most popular option), aspects of the project, or parts of the system. Classification is reflected in the headers and sections that are contained in the SRS, and, in some cases, by means of tags. The classification schema may change from project to project, although there are good reasons to follow the same schema in every project.

All of our interviewees employed a classification schema to classify requirements. As Fig. 3 shows, the most frequently recurring *Structuring concept* was the system’s *Main functionality* (10 subjects, 42%), which subsumed related concepts, including ‘main functionalities’, ‘group of

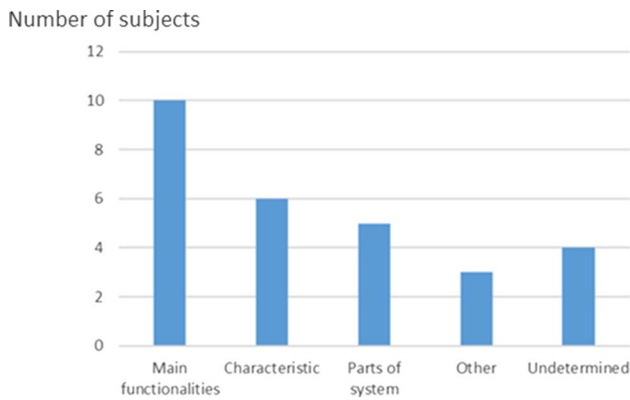


Fig. 3 An overview of the structuring concept used for classifying requirements in SRSs

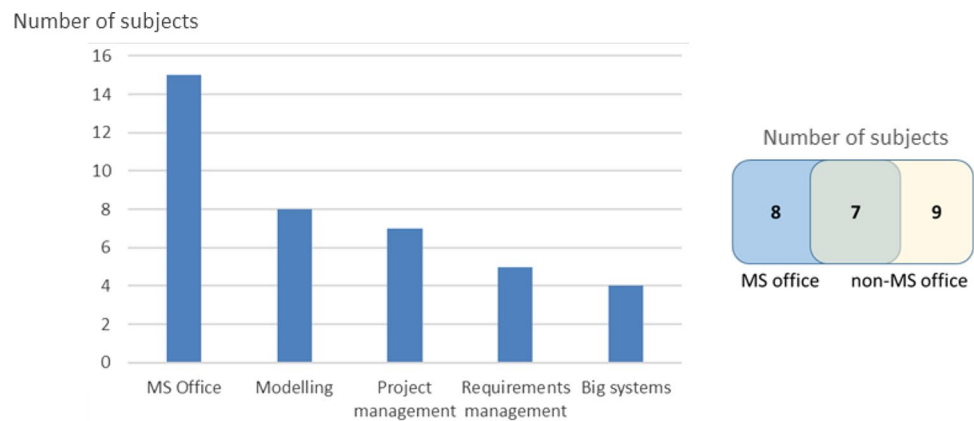
services’, and ‘business areas’. Six of the participants (25%) compiled requirements that are similar to one another in terms of a *Characteristic*, for example: “the requirements are organized in different sections that have headers, that are based on different aspects of the system, like ‘signed requirements’, ‘functional requirements’, ‘capacity requirements’, and ‘cost requirements’” S16(I). Alternatively, five subjects (21%) followed a solution-oriented approach and grouped requirements in terms of *Parts of the system*, for example: “‘user side’, ‘back office’, and ‘administration’” S11(F), and, in certain cases, including machine functions and product structure, for example, in terms of “car system areas” S20(J). Three concepts were mentioned only by one interviewee (shown together as “Other” in the graph). These were (i) *Artefact* (“order of the use cases, activity diagrams, ...” S1(A)); (ii) *Workpackage* (“The headers are not the same from project to project and are usually based on the workpackages [...]. In that specific project, they were User Interface, Platform Support and Platform Customization” S6(D)); and (iii) *Model elements* (“These headers are not usually the same from project to project, because they are usually based on Business Process Models” S13(G)). Remarkably, four interviewees (17%) could not identify a pre-defined structuring concept or the description that they provided was not sufficiently detailed, i.e. *Undetermined*.

Concerning the *Structuring element*, 92% of our interviewees (22 respondents) use *Headers and Sections* for this purpose. This classification process was usually described in somewhat vague terms. For example, S8(E) remarked: “We just put some headers for the different parts of the document, but the requirements were not organized inside.” However, seven interviewees (29%) reported on using a more complex, multi-level classification schema, either in a general way: “it was organized with sections and subsections” S22(K), or by

providing specific details: “In the first level of sections, the headers correspond to the different machine functions. At the second level, inside each machine function, they have requirements to start the function, to stop it, the requirements during the function, etc.” S17(I). In one case, the classification schema mimicked the system structure: “On the first level, the headers of the sections were the different webpages. On the second level, the headers corresponded to the different sections of the webpage” S2(A). In addition to headers and functions, two participants stated that they use *Tags*. In one case, tags are used only as a means to structure the requirements specification by using them as fields in an Excel spreadsheet. Only one subject reported on relying exclusively on software engineering concepts to structure requirements specification by grouping user stories into *Epics* (S22(K)).

Finally, we observe that 22 of the interviewees discussed the degree of *Flexibility* of the classification schema. Twice the subjects stated that the classification schema might *Change from project to project* compared to those that have a fixed schema (14 vs. 7, i.e. 58 vs. 29%). One participant reported that they used a mixed approach with respect to the classification schema: “The groups [first level] are the same from project to project [...], lower levels vary from project to project” S19(J). Even in the case where a fixed schema was used, some small degree of flexibility was present, for example, as described by S3(B): “remove or add headers if necessary”. Most of the interviewees who reported that they used a flexible classification schema mentioned that they also used project-related aspects as structuring concepts (for example, ‘functionality’ or ‘system parts’). Consequently, as reported by S7(D): “usually they [the headers] are not the same from project to project”. However, several other practitioners stated that they still used a fixed structure, that is the same classification schema in every project that they worked on. Given that, S15(H) reported that “not every project has requirements in every section”. A good reason for always using the same headers in a requirements specification document is that it will prevent the practitioner from forgetting a specific type of requirement. This claim was confirmed by S23(L): “We use similar organizations of the requirements between the different projects, so we are sure we don’t forget any aspect of any part”. Another reason why the same set of headers is re-used is that “we aim to create a database in the future, so it would be easy to reuse requirements” S5(C).

Fig. 4 Requirements specification tools: Frequency of categories (left) and usage of MS Office tools (right)



4.5 RQ4: what tools, if any, are used in the specification process?

The specification of requirements is always supported by the use of tools. Three options enjoy similar degrees of popularity: (1) the use of MS Office only (mainly Excel, but also Word), (2) the use of other types of tools only, or (3) the use of MS Office in conjunction with other types of tools. In the latter case, MS Office tools are primarily used to create a first version of the requirements specification document, which is then incorporated into another, more sophisticated tool. Non-MS Office tools include requirement management tools (for example, Doors), modelling tools (for example, Rational Rose), and project management tools (for example, JIRA). Some practitioners use more than one tool. These tools can be used for requirements specification but also for requirements management.

All of our interviewees reported that they use tools for specifying requirements. However, as Fig. 4 indicates, there is a great deal of diversity in terms of which tools they use. A similar number of practitioners use *MS Office* tools (15 participants, 63%) and non-MS-Office tools (16 participants, 67%). The three possible combinations with respect to the use of tools were represented as follows: (1) MS Office only (8 participants, 33%), (2) non-MS-Office only (9 participants, 37%), and (3) both types of tool (7 participants, 29%).

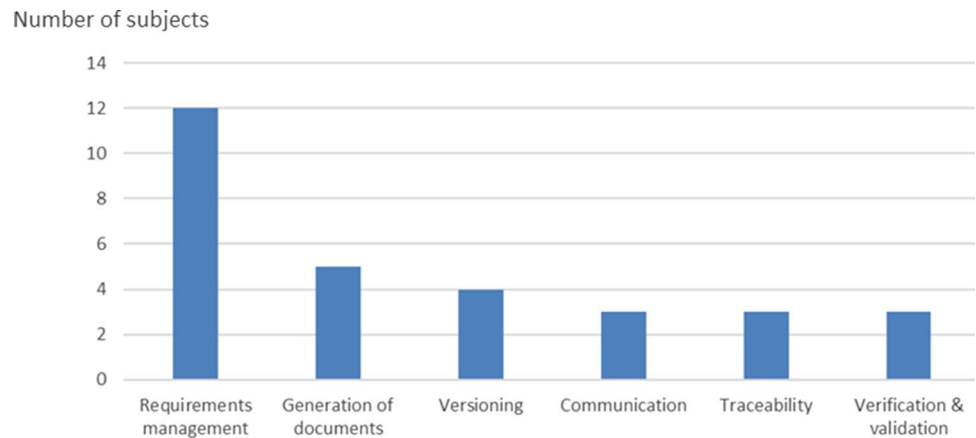
The interviewees who stated that they used MS Office primarily used *Excel* (nine interviewees, 60% of the users who use MS Office), followed by *Word* (five interviewees, 33%). The interviewees' preference of Excel over Word follows their preference for structure over appearance, as evidenced by the structuring facilities that Excel offers. *Visio* (two interviewees, 4%) and *PowerPoint* (one interviewee, 2%) were used less frequently. S10(F) provided clear reasons why MS Office tools enjoy such widespread use: "The use of a dedicated tool for RE was proposed, but was discarded because it was too complicated and the learning curve was big, and it was not available to all the people working on

the project, since it was a commercial tool that needed a license".

Even though the differences between requirements specification tools are sometimes blurred and constantly changing, we grouped the non-MS-Office tools into four categories. Surprisingly, the category of *Requirements Management Tools* (for example, Doors) was not the most frequently used tool, with only five interviewees who reported on using such tools (31% of the interviewees who did not use MS Office tools). Instead, *Modelling Tools* (namely, *SAP PowerDesigner*, *CGS Electra*, *IBM Rational Rose*, and *Sparx Enterprise Architect*) were more frequently used, with eight interviewees (50%) reporting that they used them. Furthermore, we note that *Project Manager Tools* were fairly popular, with seven interviewees reporting that they used these tools (44%). In this category (and all other non-MS-Office options), *Jira* was the tool that was used most frequently, with four interviewees stating that they used this tool. *Big Systems*, including *Siemens TeamCenter* and *IBM Rational ClearCase/ClearQuest*, comprise the final and least populated category (four interviewees).

Practitioners who use these more RE-oriented tools enjoy a clear advantage over MS Office-only users, given the richer features that they have access to. For example, S1(A) used *SAP PowerDesigner* for "drawing use cases, defining activity diagrams, specifying requirements in NL as an attachment to use cases, writing all information related to the use case (including non-functional requirements in the "other" section), adding pictures (photographs of mockups) for specifying the [user] interfaces, and generating the software requirement specification". Note the disparity between this and the sparsity of functionality in: "writing a huge list of requirements in an Excel file" (S5(C)). For this reason, nine out of 16 interviewees (56%) managed to perform their requirements specification work with just these tools, without any support from the MS Office suite. While four practitioners (44%) managed to perform their work with only one tool, the other five (56%) used a second non-MS Office tool. For example, S4(B) used "Clear Case for managing requirements

Fig. 5 Requirements specification tools: Their use in other activities



and bugs, Rational Rose for creating different types of diagrams, and Jira for user stories and a little bit of traceability”. This diversity in tools may cause some problems for some practitioners. However, a number of mitigation actions can be consequently put in place, for example: “trying to move the bug/error management from ClearQuest to Jira” (S3(B)).

Of the seven practitioners who reported on using MS Office tools as support for non-MS-Office tools, five of them stated that their primary reason for doing so was to create a first version of the requirements. For example, S14(H) mentioned that “Excel is used at first to quickly assemble the stakeholder requirements. Once a stable version of the requirements is ready, they are incorporated into Cockpit, and from that point, all of the tasks related to requirements are done there”. Other reasons given by some interviewees included recording traceability matrixes, communicating with stakeholders, and building a single-page document as a summary of the project (using PowerPoint). Tool workflows may ultimately become quite complex, as was the case reported by S16(I): “SafetyCase is used until a first stable version of the requirements is achieved (to gather the requirements, specify them, export to other development tools, etc.). Once they have the first version, the requirements are exported to Word so they could be verified/validated by customers, marketing and so on. Once the documents have been validated by all parties, the requirements are exported to Enterprise Architect, to relate the requirements into the system modules. And after that, they are exported to TFS (Team Foundation System), where they do the actual coding”.

Figure 5 presents the additional activities, beyond specification (which includes *Diagramming* and *Annotating* specification elements with additional information), that were mentioned by our interviewees in connection to using tools. Half of the interviewees (12 interviewees, 50%) mentioned requirements *Management* but without proving any specific details on the topic (“managing requirements in general” S13(G)). However, some of our interviewees referred to particular management activities, for example,

progress monitoring and prioritization. Since it is usually a complex task, requirements management requires quite powerful tools. Unsurprisingly, practitioners who claimed that they exclusively use only MS Office also reported that they engage in some type of requirements management. For example, S11(F) stated that they “used Excel to specify requirements and follow their progress across the whole project (development, testing, etc.)”. The remaining activities included:

- *Generation of documents* for system requirements specification or other types of reports (five interviewees, 21%). The reports are generated from tools, for example, Power Design or Mingle, that allow “exporting requirements to different formats like.doc,.xls, etc.” S13(G).
- *Versioning*⁶ (four interviewees, 17%). For instance, S4(B) used “Clear case for managing versions of requirements documentation and bugs”.
- *Communication* within the organization (three interviewees, 13%). Having the requirements centralized in Jira, for example, “helped in the communication of all the teams working on the project (requirements engineers, developers, and testers)” S4(B).
- Keeping track of *Traceability* information (three interviewees, 13%). Tool support is useful in maintaining (i) intra-requirements links using traceability matrixes (even with Excel, as reported by S18(J)) and (ii) links to other elements, for example “into the system modules [with Enterprise Architect]” S16(I).
- *Verification and validation* of requirements. The three practitioners (13%) who worked at company J reported that they used the Electra tool for verification and validation purposes.

⁶ Although version management could be considered part of requirements management, since it was explicitly mentioned by four interviewees, we decided that it should constitute a category of its own.

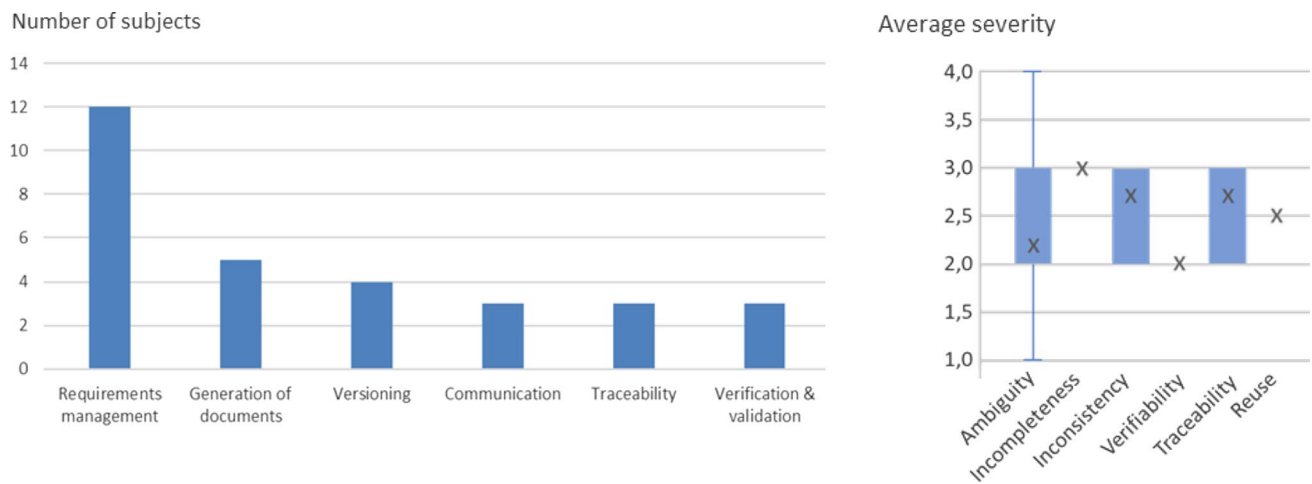


Fig. 6 Challenges to requirements specification (left) and the severity of these challenges (right)

4.6 RQ5: What challenges, if any, are faced in the specification process?

It is not uncommon for practitioners to face a number of challenges related to requirements specification. A single project may give rise to multiple challenges. Due to the widespread use of NL, challenges such as ambiguity, incompleteness, and inconsistency are frequently mentioned. Other challenges include a lack of traceability and a lack of verifiability. The relevance of these challenges is characterized as being 'low' to 'medium'. Incompleteness is classified as the most severe challenge. The combined effect of several challenges may increase their perceived severity.

Figure 6 shows the different challenges that the practitioners reported to us during the interviews. Only four interviewees (17%) did report *No challenge* in the project that they chose to talk about. Different reasons were provided for this. For example, S3(B) claimed that this was the case “because it was really focussed on a goal understood by everyone working in the project”. S16(I) stated that “the tool they use is quite good in helping them with that [with specification]”. This sentiment was echoed by S6(D), who observed that “it is not the usual case, but this project was like that”.

Ambiguity was reported to be the most mentioned challenge to setting out specification requirements (16 interviewees, 67%). This was followed by *Incompleteness* (nine interviewees, 67%) and *Inconsistency* (six interviewees, 25%). The combined effect of these challenges and their impact are astutely summarized by S22(K): “Some requirements were missing at the end, and some requirements were ambiguous and inconsistent, and they noticed that after a month or so of being on the development stage. [...] extra work was needed, especially if the changes / missing features had a big impact on other things on the system, and some extra

functionalities were moved to the maintenance stage to be able to finish on time”.

The interviewees also observed that

- The impact of the challenge to requirements specification may depend on different factors, for example, time. With respect to ambiguity, S19(J) remarked that “it [ambiguity] could have a low impact or a high impact. It depends on when the problem is detected; usually, it is only necessary to make small clarifications, but sometimes, ambiguous requirements get passed on to the development stage, and then, the impact is high”.
- Unsurprisingly, problems with ambiguity were most frequently related to human communication, as explicitly mentioned by five interviewees (22%). The most dramatic effect of ambiguity was found in the testing phase because “the testers didn't understand the requirements properly [...] time was lost when testing the requirements because of that” S1(A). A number of participants also reported on misunderstandings between developers and requirement engineers/analysts. Early detection of ambiguity is crucial, but it can be solved: “just by clarifying and improving the writing at the different iterations” S14(H). S1(A) mentioned that term glossaries could be used as a mitigation asset so that “everyone refers to the same concept using the same word”.
- The type of artefact that is used to express the requirements may also have an influence on requirements specifications. For instance, S7(D) describes their experience of a lack of verifiability as “from the fact of having user stories: how do you know you are doing it 100% right? You are never sure”.
- S23(L) considered incompleteness to be a wicked problem in the context of requirements specification because “it is difficult to achieve completeness without putting

more requirements than necessary”. In other words, by addressing incompleteness, one may make the error of ‘over-committing’ specifications. However, being oblivious to specific requirements during the later phases of a software development project may have severe consequences on the project. Note the remarks made by S11(F) concerning his project: “Some requirements were missing at the end, and a couple of missing requirements were found in the middle of the development, and it implied some extra cost and the solution would have been better if these requirements were known beforehand.”

Our participants highlighted two other challenges. Four interviewees (17%) reported on issues concerning *Lack of traceability*, for example: “some requirements were lost, they did not know where they came from, how they were at development or testing, etc.”. S12(G) experienced a lack of traceability in the context of the evolution of a system because, in his project, “there was a necessity to investigate requirements to know how the system worked before creating a new evolution / function / modification in the system”. A particular class of traceability, namely a statement of the relationships between the requirements themselves, was highlighted by S10(F) in the following: “if you do not have that clear, you will end up with having requirements that are inconsistent in the specification”. S15(H) also referred to this class of problem with respect to *traceability*, adding that there is a further challenge of relating requirements that are at different levels.

Three interviewees (13%) mentioned that verifying the requirements (*Lack of verifiability*) may also pose a challenge. For S7(D), this problem could not be solved in the context of user stories because “they just tell what the system has to do, but with not much detail, so sometimes developers do not know how to implement a part of the system [...] how do you know you are doing it 100% right? You are never sure.” The same problem was reported by S1(A) but in relation to testers instead of developers.

Our interviewees mentioned four other challenges:

- *Requirements reuse* with respect to system or modules that were specified before the adoption of the current requirement management tool, thus making it impossible to access the history of the requirement. (S11(F))
- The appropriate *Level of detail* needed to specify all stakeholders. S1(A) mentioned that “The level of specification was good enough for developers and requesters because they participated in the discussions on requirements, but not for testers [...] They did not give the requirements to the testers at the early stages of the project to check whether the requirements were specified in sufficient detail.”

- *Short-term vision*. This problem was highlighted by S1(A), who stated that: “The requirements weren’t good enough for later evolutions of the system (so someone was able to modify the system without having been involved in the first release) because they do requirements for this project, but not for the long-term life cycle. They only look project-wise, because long-term maintenance is not usually needed.”
- S7(D) mentioned problems related to communication due to the fact that the organization is based across several countries and the challenge that *Team Spread* poses for their work.

Finally, we refer to the reported severity of the impact of these challenges in a scale from 1 (‘very low’) to 5 (‘very high’), as summarized in Fig. 6. The average severity of the reported challenges was 2.6, i.e. a figure between ‘low’ (a score of ‘2’ on our scale) and ‘medium’ (a score of ‘3’ on our scale).⁷ On the level of individual analysis, the four challenges discussed above related to the quality of the requirements (ambiguity, incompleteness, inconsistency, and lack of verifiability). These challenges were scored as showing ‘average severity’; between a score of 2.0 (for lack of verifiability) and a score of 3.0 (for incompleteness). Furthermore, we note that lack of traceability and requirements reuse were scored between intervals, each scoring 2.7 and 2.5, respectively. Examining the details of some of these challenges, we note:

- In relation to ambiguity, most of the interviewees considered it to have a low impact on their work for different reasons. For example, S12(G) stated that “it only implied asking for clarifications during system development”. S9(E) claimed, however, that “it only was necessary to have different iterations to specify them properly”. According to S13(G), the impact of ambiguity was limited in the sense that “[ambiguity] only happened at the start of the project, when the first versions of requirements are released”. S7(D) dealt with ambiguity in the following way: “it is about being in touch with developers and answering questions and clarifying parts of the project as the project develops”. An exception to this approach was voiced by S1(A), who reported that the impact of ambiguity “was pretty high, as time was lost when testing the requirements because of that [ambiguity]”. In this context, S19(J) provided a somewhat more nuanced response by observing that “[impact] depends on when the problem is detected. Usually, it is only necessary to make small clarifications, but sometimes

⁷ Not every interview participant rated the severity of each of the challenges.

ambiguous requirements get passed to the development stage, and then the impact is high.”

- S11(F) reported on the severity of a lack of traceability and incompleteness together: “Some requirements were missing at the end, and a couple of missing requirements were found in the middle of the development process. The impact was high since it implied some extra cost [...]. Second, a lack of traceability was an issue too. Some requirements were lost. They did not know where they came from, whether they were at the development stage or the testing stage. The root cause of these problems was a lack of consistent documentation, accompanied by an excessive use of Post-It Notes and such like”.

5 Analysis

The presentation of our analysis is structured in terms of our five research questions. We summarize our main observations and report on the relationships between the concepts addressed by our research questions. We also align the results of our analysis with previous research. Section 5.6 summarizes the relationships that we identified in terms of demographic characteristics. Further information concerning our analysis can be found in the protocol.

5.1 RQ1: what artefacts are used for specifying requirements?

The use of unrestricted NL is dominant in this context, in contrast to the complete absence of formal methods. We found that textual artefacts enjoyed more widespread use by our interviewees who (1) have spent more than 10 years in their position or organization; (2) work in organizations in the telecommunications or IT solution domain; (3) reported on projects that did not deliver embedded systems (such projects also represented the largest share of projects that exclusively used unrestricted NL, by far). We also found that (4) the use of models for requirements specification is more frequent by practitioners with experience of working at a university or research laboratory. Finally, several practitioners reported that they use several artefacts in conjunction with other artefacts and, remarkably, (5) website development projects use the highest number of different artefacts. We also note that (6) projects that are developed according to Agile methodologies use more artefacts than those that follow a Waterfall approach.

Our study shows that NL sentences are the most popular instrument for specifying requirements. It is clear that, for our interviewees, the well-known drawbacks of NL (ambiguity, incompleteness, etc.) are overlooked because of its ease of use. The dominance of NL supports the suggestion that

practitioners make to researchers on devoting more research efforts to NL requirements-based RE [13].

A more striking observation is that not one of the interviewees stated that they used a type of formal language, even in the domain of embedded systems, where some kind of formal verification is sometimes required. It is worth remarking that formal languages were not used, even though the majority of the interviewees hold either a master’s or a bachelor’s degree. It could be argued that the practitioners’ educational background did not pose an obstacle to their use of a formal language in this context. However, it could also be the case that they received training in the formal languages but have forgotten how to use it, or they find it too challenging to use. Or that, in spite of their degree, they never received training in the relevant formal language.

We analysed several demographic factors of the interviewees and found the following relations:

Relation to practitioner characteristics

- The percentage of interviewees with less than 5 years in their current position or organization who used textual artefacts did not exceed 25%. However, this percentage increases to 78% (position) and 100% (organization) when the number of years is greater than 10 years. This increase in the use of textual artefacts as a practitioner’s level of experience increases suggests that the interviewees had learnt to appreciate textual artefacts as they became more experienced and thus were better positioned to deal with the limitations of unstructured NL sentences as the sole means of requirements specification documentation.
- Ten of the 11 interviewees (91%) who reported that they used models had worked at a university or research laboratory in the past. Only five of the 13 (38%) who reported that they used models did not have this type of experience. This difference may be attributed to the practitioners’ knowledge of modelling languages from their previous research-oriented positions. Nevertheless, even if this is not the case, these practitioners may be more educated engineers in the sense that they know the advantages and limitations of such modelling languages, and they may possess the skills that allow them to use these languages. We could not find any further positive influence on model adoption from other demographic data (for example, educational background or project domain).

Relation to organization characteristics

- All of the interviewees who worked for organizations where telecommunication was their principal activity (six interviewees) and the majority of the interviewees who worked in producing IT solutions (six out of nine

interviewees) reported on using textual artefacts. None of the nine remaining interviewees who worked in other types of organization reported on using textual artefacts. We can thus speculate that the interviewees who work with telecommunication and IT solutions find themselves in a more ‘technologically rich’ ecosystem than those who work, for example, with software consultancy or public transport administration. We conclude, therefore, that practitioners who work in telecommunication and IT prefer textual artefacts over unrestricted NL.

- While most of the interviewees who worked in organizations with less than 10,000 employees reported that they used textual artefacts (nine out of 11), the situation was the opposite with those who work in organizations with more than 10,000 employees, where eight out of 11⁸ did not use textual artefacts.

Relation to project characteristics

- The majority of interviewees who worked in projects related to websites, mobile phone technology, or carrier business (12 out of 15) reported that they used textual artefacts. However, no interviewee who worked in projects that produced embedded systems used textual artefacts (zero out of six). In fact, the percentage of interviewees who only used NL in their requirements specification work was significantly greater for interviewees who worked in embedded system projects (three out of seven, i.e. 43%) than for the rest of interviewees (out of 17, i.e. 6%). In summary, textual artefacts are not well-suited for the documentation of requirements specification for embedded systems, and consequently, the interviewees continued to use unrestricted NL sentences.
- Remarkably, three out of the four interviewees who worked on website development projects were the only practitioners who used the four different types of language. The fourth practitioner used three types of language. This indicates that the requirements for website development projects are very diverse and require a combination of different documentation approaches.
- On the other hand, the development methodology also influences diversity. While the percentage of interviewees who worked in Agile projects and used only one specification artefact was as low as 25% (one out of the four interviewees who reported that they used only one artefact), this figure increased to 37,5% (three out of eight) of practitioners who used three artefacts and to

67% (two out of three) of practitioners who used four artefacts. These facts seem to be inherently related to the lightweight documentation style promoted by Agile methodologies, which promotes the use of ‘convenient’ artefacts without strict rules.

Relation to previous research

Our results align well with most of the previous research in this area, as discussed in Sect. 2. In fact, the frequency of the notation systems that our interviewees used matches quite well with the report compiled by Wagner et al. [39]. Hotomski et al. [17] analysed the difference in the use of NL between (1) companies that use a Waterfall approach and (2) Agile companies. In their study, Waterfall companies were observed to primarily use basic requirements, while Agile companies used user stories to a greater extent (as a textual artefact). This finding is also corroborated by our study, where we note that six out of eight practitioners who used Agile methodologies also used user stories, and 15 out of 16 practitioners who used Waterfall methodologies also used unrestricted NL. The relationship between Agile methodologies and user stories is exemplified by the following statement made by S7(D): “[W]e changed the way we were working quite a lot and used more Agile principles, so we started to writing user stories [...]”.

Some of the studies included in this paper reported on the use of models for specifying requirements. Our study aligns with the frequent use of models reported by Liu et al. [22] and Palomares et al. [28] and stands in contrast with the scarce use of models reported by Sikora et al. [38], who reported that two out of ten practitioners made frequent use of models.⁹ Remarkably, Sikora et al.’s study was conducted in the domain of embedded systems. We have already reported that this domain has some particularities that may explain this discrepancy. The present study also corroborates the following findings: (1) UML is the most frequently used modelling language [8, 20, 28]; (2) formal methods or goal models are very rarely used (if ever) [20, 38]; and (3) multiple specification artefacts can be used in a single project [22].

Unfortunately, we did not gather evidence regarding the specific topic of QR documentation. Thus, we cannot align our study with previous research that has reported on this particular aspect (for example, [2, 3]).

⁸ Two interviewees did not inform us about the size of their organization.

⁹ However, one must take into consideration the fact that their definition of what a model is, is different to our category of graphical languages since they did not include mockups or figures in their analyses of how requirements can be clarified.

5.2 RQ2: what templates and guidelines, if any, are followed to specify requirements?

Templates and guidelines were more frequently used by interviewees who (1) work in technical roles; (2) have a significant amount of work experience; (3) reported on large and/or long-term projects. We also observe that (4) standards are used only in projects that deliver embedded systems; (5) templates and guidelines that are defined by organizations are dominant in the case of very large organizations and/or long-term projects; and (6) Agile projects use templates and guidelines more frequently than Waterfall projects.

In Sect. 4.3, we discussed how practitioners frequently use templates and guidelines when specifying requirements (75% of interviewees reported that they used templates and guidelines). Below, we provide several observations regarding the demographics of the group of practitioners whom we interviewed:

Relation to practitioner characteristics

- Managers used templates or guidelines less frequently (67% of managers only) compared to analysts/designers and developers/testers, all of them using templates or guidelines. This observation highlights the technical nature of templates and guidelines. Managers may find it challenging to use templates or guidelines since they have different priorities. In a similar vein, from the seven interviewees who reported that they use templates or guidelines for requirements attributes, the majority (five) were analysts/designers (which leaves only one analyst/designer not using templates or guidelines for this purpose). Again, we note that no single manager made use of templates or guidelines.
- Experience matters: The percentage of usage of templates and guidelines increases from 60% for interviewees with less than 5 years in their position to 86% for the rest of the practitioners whom we interviewed. This difference may indicate that our more experienced interviewees had become aware of (i) the role of templates and guidelines or (ii) the necessity of using templates and guidelines, as they acquired additional professional knowledge.

Relation to organization characteristics

- Interviewees working in very large organizations (more than 100 000 employees) use templates or guidelines that are defined by their organization. This observation supports the hypothesis that very large organizations possess more accumulated knowledge, which allows them to define the best solution for a diversity of projects and employees.

Relation to project characteristics

- The duration of a project has a similar impact on the use of templates and guidelines, as does the practitioner's professional experience. Only 60% of the interviewees who reported on short projects (less than one year) stated that they used templates or guidelines, compared to 86% of practitioners who worked on long-term projects. We also observe a dominance of the use of organization-wide templates or guidelines in large projects. 50% of the interviewees in large projects that the provenance of respect to templates and guidelines that they use was from the organization for which they worked.
- In general, the use of standards was scarce, with only three interviewees reporting their use (12.5%). However, all of those practitioners who said that they used standards had worked on embedded systems projects (where the percentage consequently grows to 43% of interviewees reporting on embedded systems projects).
- We observed a somewhat contradictory state of affairs regarding the software development approach that the practitioners used. While the percentage of interviewees working on Agile projects who declared that they used templates or guidelines was higher than those practitioners who were engaged in Waterfall projects (87 vs. 69%), all of the interviewees who worked on Waterfall projects were able to explicitly identify the purpose of such templates or guidelines, compared to the 43% of practitioners who used an Agile approach to software development. We cannot provide any reason for this difference. A second observation is that none of the interviewees who worked on Agile projects used standards as a source of templates, and only one practitioner who worked on Agile projects (12.5% over 8 interviewees working on agile projects) used templates or guidelines for more than one purpose compared to the six interviewees (38%) who used standards as a source of templates and guidelines in the context of the Waterfall approach to software development. Finally, we observed that all of the interviewees who reported on using requirement attributes for structuring worked in Waterfall projects.

In conclusion, only seven practitioners used templates or guidelines for more than one purpose, and in particular, only four subjects (17% over the total) use templates and guidelines together. It may be thus argued that holistic solutions should be used to provide more guidance to requirement engineers with regard to writing specification documents.

Other observations

- The interviewees used tools to define document layouts. In particular, this was the case for the four interviewees who reported on the use of tools. Practitioners can

quickly begin their work on the requirements specification phase and become acquainted with the details of the project that they are working on.

Relation to previous research

The infrequent use of standards that we observed in our study aligns with the results reported by Wagner et al. [39] and Franch et al. [12], especially regarding the low level of knowledge of standards among requirements engineering practitioners. Wagner et al. [39] report on the frequent use of templates/standards, but these templates/standards were internal to the relevant companies. These scholars also observed a low use of actual, or, as they call them, ‘external’ standards. On the same topic, Franch et al. [12] report that templates are used independently of standards, as observed in the present study. This low level of knowledge regarding standards may be considered a problem, even in Agile projects, since standards are a good source of established knowledge. Even if standards are not strictly followed, they can contribute to the definition of customized templates or methodologies.

In contrast, the results of the present study do not align with Elahi et al. [8], who concluded that up to 73% of the participants in their study used templates or standards. Similarly, Palomares et al. [28] reported that half of the participants in their study used templates or patterns in the context of reuse. When we look more closely at the details of these studies, we realize that Elahi et al. focussed on one type of requirement, namely security requirements, which may have been a determining factor with regards to their results. We note that for specific types of requirements, especially in the case of security requirements where a plethora of standards and regulations exist, it is more likely that templates will be considered a valuable asset for requirement engineers.

5.3 RQ3: what classification schemas, if any, are used for organizing requirement specifications?

In addition to being the most popular specification structuring criterion, main functionality is especially dominant: (1) in Agile projects and (2) by less experienced practitioners. On the other hand, classification schemas are more often preserved between projects that are either (3) long-term projects; (4) large projects; (5) deliver embedded systems; or (6) are produced by practitioners who do not possess a great deal of experience. In fact, we observe that the classification schema is prone to change from one project to another in (7) small organizations; (8) telecommunications or IT solution companies.

The use of main functionalities or groups of services is the most used structuring criterion. In fact, it is the only one

used with some other criterion (principally, parts of the system). Besides, we find the following relations to demographic characteristics.

Relation to practitioner characteristics

It is worth remarking that the four participants with less than 10 years in industry had dominance in all factors: main functionalities as structuring concept (3 interviewees), all of them using a multi-level structure with sections or headers as structuring element, and the same fixed classification schema (3 interviewees). This suggests that novice practitioners are likely to adopt a conservative approach to specification structuring (multi-level and fixed structure), which may change as they gain professional experience.

Relation to organization characteristics

- The smaller organizations that were included in our study (< 1000 employees) employ different classification schemas across projects, as reported by six out of the seven subjects who worked in this type of organization. When we consider RQ2, it follows that the size of an organization may be an indicator of flexibility and the ability to customize the requirements engineering process.
- Similarly, we note that the telecommunication companies included in our study also opt to use flexible schemas, with all participating practitioners who work in this type of company reporting that they employ flexible schemas. This may indicate that the projects that this type of organization works on are fundamentally different to the projects undertaken by the other organizations.

Relation to project characteristics

- Six out of the ten interviewees who used *main functionality* as a structuring criterion worked in Agile projects. In other words, 75% of the interviewees who worked in Agile projects structured their specifications using the functionalities or services provided by the system. This observation aligns well with the functional orientation of Agile backlog, where themes, epics, and features are used to structure user stories.
- The seven interviewees who worked on projects that delivered embedded systems used the same classification schema across different projects of this type. In fact, only one interviewee who had not worked on embedded systems projects followed the fixed approach. We also note, however, that all the other practitioners declared that the structure could change across projects. If we connect this

observation with RQ1 and RQ2, it can be argued that a fixed classification schema is a natural consequence of the use of standards. The use of standards also counteracts the freedom associated with the use of unrestricted NL in embedded systems.

- Practitioners who reported on working in long-term projects (more than a year in duration) and large-scale projects (more than ten members) were more likely to use fixed schemas than variable schemas. The percentage of practitioners who used fixed schemas and who were engaged in long-term or large-scale projects was 71 and 38%, respectively. When we consider all of the practitioners included in this study, the figures are 58 and 29%, respectively. This difference may indicate that practitioners who work on long-term or large-scale projects need to be more proficient with respect to requirements specification in order to manage the complexity associated with such projects.

Relation to previous research

Our results in this domain are in accordance with those of [26] study, which states that most practitioners apply some form or another of classification system in their requirements specification work. However, Méndez et al.'s study does not provide details on what aspects of the requirements such classification systems are based on or whether they are used across different projects. The present study addresses these questions. We also note that the other research studies included in the present paper do not provide similar insight into requirements classification systems.

5.4 RQ4: what tools, if any, are used in the specification process?

The balance between the use of MS Office and more sophisticated tools could change in the future since several organizations indicate that they wish to adopt more sophisticated tools for the requirements specification process. MS Office tools are prominently used by two classes of practitioner: (1) practitioners who write guidelines and (2) practitioners who do not have experience working at a university or research laboratory. With respect to non-MS-Office tools, we make the following observations: (3) practitioners who use requirement management tools shared several properties in relation to the use of templates and the structuring of requirements; (4) modelling tools are used for a broader range of purposes than other types of non-MS-Office tools; and (5) project management tools are prominently used in Agile projects.

The balance between the use of MS Office tools and other, more sophisticated tools is indicative of the ongoing debate with respect to the use of such tools, not only in the specific

context of requirements specification or RE but in the context of software engineering in general. While specialized tools provide many advantages once the practitioners have mastered them, their potentially steep learning curve, associated licensing costs, and the challenge of fitting them into a project or company toolchain may pose an obstacle to their adoption. The present study illustrates that this debate is far from over and that extreme positions regarding the use of sophisticated tools should be avoided.

However, when we conducted our study, five interviewees reported that their organizations were considering adopting requirements management tools in the near future, thus indicating that MS Office tools cannot fulfil all requirements specification purposes. Some interviewees mentioned even more ambitious plans regarding the same topic, for example, regarding the creation of create repositories (“The tool is really powerful, but they are still not using the entire powerfulness of the tool. They [the company] are looking now at the possibility of creating some kind of repository with this tool to have all the knowledge about all the projects centralized, so they know what each system is about (but not with reusing purposes)” S1(A). On the theme of managing their company’s portfolio of products, S12(G) reported that “right now the tool is only used for managing the requirements at a project level, although it is expected to be used in the future to provide an overview of all the existing solutions inside the organization”.

We analysed the types of tools that were used by the practitioners and identified the following relationships:

- All of the seven interviewees who worked on how-to-write guidelines used MS Office tools. This is a good indicator that the other used tools focus more on the product (the requirements document) than on the process, and opens the space to add this perspective to the tools in order to gain competitive advantage. On the other hand, almost all the practitioners who reported on using MS Office tools (14 out of 15) did not use these tools to generate SRS or reports (apart from the generation of pdfs, of course). This is not surprising, given the limited RE-related capabilities that MS Office tools offer.
- The five practitioners who reported that they use requirements management tools share several characteristics:
 - o They adopt organizational-defined templates defining the layout of the document.
 - p They do not organize the requirements specification according to the main functionalities of the product.
 - q In most cases, they apply the same classification schema to each project that they work on.

These three characteristics seem to facilitate the management of requirements, which is ultimately the motivation why one would adopt this type of tool. As stated by S15(H),

the central expectations that a practitioner has regarding a requirements management tool are “[referring to the Cockpit tool] you can add different levels of requirements, i.e. stakeholder requirements, system requirements, and subsystem requirements, specify them and manage them”. On the other hand, the same five practitioners also reported that they use MS Office tools in their work, thus suggesting that requirements management tools are not powerful enough to fulfil every task associated with requirements specification and requirements management.

- Modelling tools outperform other tools in terms of versatility. Of the eight practitioners who reported that they use such tools, all of them identified three task areas in which they used them. We note that every practitioner who reported on using tools for verification and maintaining traceability used modelling tools for these tasks. In addition, we also note that modelling tools are very rarely used in conjunction with MS Office tools (only two out of those eight practitioners used them).

Generally, practitioners who use non-RE specific tools for RE activities, for example, Focus Point, Safety Case, Clear Quest, or Team Foundation System, do not face the challenge of having ambiguous requirements. This could be the case because they used these tools as consumers of requirements and not as producers. Therefore, it could be argued that the requirements that are managed by these tools have already been reviewed several times, thereby explaining why the requirements are not ambiguous.

Relation to practitioner characteristics. Eight out of the nine interviewees who have no experience of working at a university or research laboratory used MS Office tools (89%, compared to the 60% of use of MS Office by the 24 interviewees). Employees at a university or research laboratory may be more inclined to experiment with advanced tools and also may be subject to fewer licensing problems than commercial companies. This claim is supported by the fact that the other two demographic characteristics of our study’s participants, namely years in industry and the participants’ educational background, show no correlation with the participants’ adoption of MS Office-based tools.

Relation to project characteristics. Project management tools are most frequently used by the practitioners who work on projects that employ Agile methodologies (six out of the seven practitioners who use this kind of tools, worked in Agile projects). Or in other words, only two practitioners who reported on using Agile methodologies did not use project management tools. The main reason for this is that it is likely that our interviewees already used these tools for managing their Agile projects, and since these tools offer enough functionalities for the practitioner to specify requirements (even if they are not as sophisticated as dedicated

requirements management tools), they prefer to use them instead.

Relation to previous research

Previous research on requirements specification and management reflects the dichotomy between requirement management tools and other types of tools. The use of tools that are not specifically designed for RE is reported on by Liu et al. [22], Raatikainen et al. [32], and [20]. Hotomski et al. [17] show that Jira and Confluence are most frequently used by Agile companies to specify requirements, while MS Word and/or Excel are used by Waterfall companies to a greater extent. In our study, we make a similar observation. Of the five practitioners who reported using Jira, four of them used Agile methodologies. With respect to the eight interviewees who reported on using Agile methodologies, six of them used either Jira or Mingle. However, this observation is not surprising since Jira, Confluence, and Mingle are tools that are focussed on Agile practices. On the other hand, of the 18 practitioners who reported on using MS Office tools, 14 of them followed Waterfall methodologies.

Raatikainen et al. [32] remark on why some of the participants in their study decided to stop using MS Office tools and use requirements management tools instead. They state that the practitioners decided to make this change to overcome the challenges associated with MS Office, especially with respect to supporting the management of interdependencies, hierarchies, and traceability. This finding aligns with our own, where 67% of the practitioners whom we interviewed reported that they use more than one tool in their work on requirements specification and that 29% of them have tried to change from one tool to another. This indicates that they are not satisfied with their current requirement specification tools.

5.5 RQ5: what challenges, if any, are faced in the requirements specification process?

The most frequently mentioned challenges (ambiguity, inconsistency, incompleteness) are less likely to occur: (1) when practitioners use textual artefacts in their requirements specification work instead of unrestricted NL only; (2) in Agile projects; or (3) when practitioners played the manager role. Ambiguity is more dominant in the case of (4) analysts, designers, and practitioners who hold a position related to requirements specification; and (5) projects that deliver embedded systems (the challenge of inconsistency was also reported in these contexts). Practitioners who failed to perceive any challenges in the requirements specification process (6) did not hold an MSc/PhD degree; (7) had worked in the same position for several years; or (8) reported on a small project during their interview.

The dominance of ambiguity, incompleteness, and inconsistency as challenges is an unsurprising finding (of the 24 interviewees, 79%, i.e. 19 interviewees, mentioned at least one of these challenges) given the intensive use of NL as a specification language (see Sect. 4.2). These challenges were seen as an obstacle “to putting everyone together on the same page” S13(G). It is worth mentioning that although ambiguity was mentioned more frequently by the interviewees, incompleteness was perceived as more damaging to their work, with nine interviewees rating it as ‘important’. This result may be of interest to the research community because a large number of research papers are focussed on the study of ambiguity but not so many on incompleteness. From an academic perspective, the use of templates could seem a valid strategy to cope with these challenges, but we observe that the majority of our interviewees (16 out of the 19 who mentioned at least one challenge) reported on using templates but still face one or more of these challenges. This observation suggests that templates should be accompanied by more prescriptive writing guidelines if these challenges are to be satisfactorily overcome in practice. We should also mention that using models or graphical elements together with textual artefacts did not improve the interviewees’ perception of these challenges.

We report the following relationships between the characteristics of the practitioners who participated in our study and the projects that they reported on (we did not find any relation with organization characteristics):

Relation to practitioner characteristics

- Of the 14 interviewees who worked as an analyst, designer, or held a requirements-related position, 13 reported that ambiguity was a challenge in their work. This challenge was also mentioned in combination with either incompleteness or inconsistency (eight out of 14). Possible explanations for this figure may be related to the fact that these practitioners are more educated than the other practitioners included in our study in the requirements engineering field *or* that they are more inclined to examine the details associated with requirements specification work, given their responsibilities within the company they work for. This number drops significantly if we consider managers: only three out of nine managers mention ambiguity as a challenge, and only one of these three together with another challenge, namely inconsistency.
- While only one interviewee out of 13 with MSc or PhD degrees does not perceive any challenge related to requirements specification, the number increases up to 3 out of 11 (27%) when it comes to practitioners without these degrees. This suggests that practitioners who have more educational background have a better understanding of the problems they face when specifying requirements.
- We observe that the 75% of practitioners who reported that they faced no challenges in their requirements specification work (i.e. three interviewees) have more than 10 years of professional experience. Although we could use the same argument as above in our discussion of the role of a practitioner’s professional experience, another possible explanation for this reported absence of challenges is that the more experience a practitioner has, the easier it is for this person to specify the requirements in a project.

Relation to project characteristics

- The three dominant challenges that we report on above were less severe in Agile projects. Concretely, 62.5% of the practitioners who reported on working in Agile projects (five out of eight practitioners) said that they had experienced at least one of these challenges. The proportion of practitioners increases to 87.5% in Waterfall projects (14 out of 16). Of the five Agile practitioners who reported that they had experienced challenges, only one mentioned more than one challenge. This magnitude represents 12.5% of the total number of practitioners who worked on Agile projects, compared to the 50% of practitioners who worked on Waterfall projects (eight interviewees) who reported more than one challenge. We hypothesize that the conciseness of user stories and their grouping into epics/features/themes are key activities that explain this difference.
- We find again that projects delivering embedded systems were seen as particularly challenging since the seven interviewees who reported working on this type of project (29% over the overall number of practitioners) represent the 50% of interviewees reporting inconsistency problems and the 38% reporting ambiguity problems. In other words, six out of the seven practitioners who worked on embedded systems mentioned that ambiguity was a challenge that they had experienced. This observation, together with the observations that we have made in conjunction with the previous RQs, emphasizes the distinctiveness of embedded systems from an RE perspective.
- On the other hand, interviewees who reported on website projects were more likely to not report on any challenge, representing 50% of the total share (2 out of 4 practitioners not reporting any challenge), which may indicate that these systems are less complex than the rest, at least from an RE perspective.
- Three out of the four interviewees who reported that they had experienced no challenges were involved in projects with less than ten employees. The fourth one referred to

Table 2 Challenges identified in our study and in other empirical studies

		Empirical studies dealing with specification challenges												Our study
		[22]	[8]	[32]	[6]	[38]	[18]	[17]	[4]	[21]	[1]	[2]	[39]	
Specification challenges	Ambiguity							X			X		X	X
	Incompleteness	X	X		X		X	X		X			X	X
	Inconsistency				X					X			X	X
	Lack of verifiability											X		X
	Lack of traceability			X					X	X				X
	Level of detail													X
	Requirements reuse													X
	Short-term vision													X
	Team spreading													X

a project that had a budget of less than one million Euros, thus indicating that it was a small project. This observation suggests that common RE-related challenges may not be relevant to small projects.

Relation to previous research

Other empirical studies have also dealt with specification challenges (see Table 2). All of the challenges that were identified by more than one participant in our study are discussed by other studies. The challenges that were identified in more than three previous studies are *incompleteness*, *inconsistency*, *ambiguity*, and *lack of traceability*. Incompleteness is the challenge that has been identified by the largest number of studies. This fact could be related to the result in our study that shows that incompleteness is perceived as the most severe challenge to RE specification. In the case of *verifiability* (identified in only one previous study), it is not perceived as a challenge in the other studies because it may be perceived as a consequence of the presence of ambiguity and/or incompleteness in RE specifications. This type of reasoning may also be applied with respect to the challenge identified as a *lack of detail*: it may be the case that this challenge has also been perceived in some research studies as *incompleteness* of RE specifications.

None of the studies that we examined for the purpose of the present study mentioned *requirements reuse*, *short-term vision* or *team spreading*. Regarding *requirements reuse*, we acknowledge that it is a very specific subject that one would not necessarily expect to emerge in an empirical study if one did not plan for it in advance. Although they aim to deliver a comprehensive RE survey, Wagner et al. [39] explicitly state that *requirements reuse* is not addressed in their survey.

5.6 Relevance of demographics: summary

From our analysis, it is evident that demographic characteristics may influence the challenges that we have identified. Since

our discussion of these characteristics spans across several sections above, we have summarized them in Table 3. For each characteristic, we highlight the following:

- *Personal characteristics*. The practitioner's role is relevant to identifying challenges and using templates and guidelines within a project, while the several facets of time (experience, etc.) are more diverse and, in particular, affect the adoption of textual specification artefacts.
- *Organization characteristics*. The size and domain of the organization influence several aspects of the work done in requirements specification, but these characteristics are not relevant to the practitioners' perception of challenges.
- *Project characteristics*. In this context, we find the two most relevant influencing factors, namely (1) the type of system that is delivered (particularly in the case of embedded systems); and (2) the development approach (particularly in projects that are developed using Agile methodologies). The duration and size of the project correlated with the challenges that were experienced, but not to the same extent as the type of system or the development approach.

6 Conclusions

This paper has presented the results of an empirical study based on interviewing 24 practitioners working at 12 Swedish companies. The paper answered five research questions whose answer can be summarized shortly as follows:

- Unrestricted natural language (NL) is the preferred type of artefact for specifying requirements, often with some other complementary notation system(s).

Table 3 Influence of demographic characteristics with regard to the observations made in this study

Demographic characteristics	Observations	Section
<i>Practitioner characteristics</i>		
Managers	Use fewer templates and guidelines than other roles	5.2
	Perceive ambiguity as a challenge less often than other roles	5.5
Analysts/designers	All of them use templates or guidelines	5.2
Analysts/designers, together with other requirements-related roles	Almost all of them perceive ambiguity as a challenge, often combined with inconsistency and incompleteness	5.5
Developers/testers	All of them use templates or guidelines	5.2
Without MSc/PhD	Are less likely to perceive challenges	5.5
With experience in university or research labs	Almost all of them use models for specifying requirements	5.1
Without experience in university or research labs	Are more likely to use MS Office tools for specifying requirements	5.4
Not novice (≥ 5 years)	Use templates and guidelines more often than other practitioners	5.2
Less experienced (< 10 years)	Primarily use main functionalities as requirements structuring criteria	5.3
	All of them structure the classification criteria into multiple levels	
	Preserve the classification schema from project to project	
Short time in the same organization (≤ 5 years)	Infrequently use textual artefacts to specify requirements	5.1
Long time in the same organization (> 10 years)	All of them use textual artefacts to specify requirements	5.1
Short time in the same position (≤ 5 years)	Infrequently use textual artefacts to specify requirements	5.1
Long time in the same position (> 10 years)	Most of them use textual artefacts to specify requirements	5.1
	Are less likely to perceive challenges	5.5
<i>Organization characteristics</i>		
Very large organizations ($> 100\,000$ employees)	Use templates or guidelines that are defined by their organizations	5.2
Small organizations ($< 1\,000$ employees)	Mostly use flexible classification schemas	5.3
Telecommunications	All of them use textual artefacts to specify requirements	5.1
	All of them use flexible classification schemas	5.3
IT solutions	Mostly use textual artefacts to specify requirements	5.1
	Mostly use flexible classification schemas	5.3
Do not work with on telecommunications or IT solutions	Textual artefacts to specify requirements are not used	5.1
<i>Project characteristics</i>		
Embedded systems	Textual artefacts for specifying requirements are not used	5.1
	Unrestricted NL sentences are used much more frequently than in other projects	5.1
	The only project type that used templates or guidelines that are based on standards	5.2
	Mostly preserve the classification schema from project to project	5.3
	Are more likely to perceive challenges associated with ambiguity and inconsistency	5.5
Website development	Use a large number of specification artefacts	5.1
Long projects (> 1 year)	Frequently use templates and guidelines	5.2
	A dominance of templates and guidelines that are defined by the organization	
	Mostly preserve the classification schema from project to project	5.3
Large projects (> 10 members)	Mostly preserve the classification schema from project to project	5.3
Small projects (≤ 10 members or < 1 M€ budget)	Are less likely to perceive challenges	5.5

Table 3 (continued)

Demographic characteristics	Observations	Section
Agile development	Use a larger number of different artefacts than Waterfall projects	5.1
	Do not use standards to define templates or guidelines	5.2
	Use more templates and guidelines than Waterfall projects	
	Use templates and guidelines for a single purpose	
	Mostly use main functionalities as requirements structuring criteria	5.3
	<i>Ambiguity, inconsistency, and incompleteness</i> are less frequent (individually and combined) and less severe than in Waterfall projects	5.5
Waterfall development	Mostly use project management tools to specify requirements	5.4
	The purpose of using templates and guidelines is well-known in the Waterfall methodology	5.2

- Templates and guidelines are widely adopted by practitioners. However, they rarely take RE standards into account.
- Requirements are often classified in terms of the system's main functionalities, although project aspects and parts of the system may also inform the classification of requirements.
- MS Office tools and other types of tools are similarly used to specify requirements, either individually or in combination with each other.
- Challenges related to requirements quality (*ambiguity, inconsistency, incompleteness*) are the most frequently mentioned challenges, being incompleteness perceived as the most severe of them.

Finally, we have also shown how demographic factors influence these findings and we have identified several other relations relevant to our results, thereby demonstrating how context is crucial to a proper understanding of the utility of our findings for interested practitioners.

Our study may benefit both practitioners and academics. Practitioners may know current practices in requirements specification adopted by a number of fellow requirement engineers, which could eventually adopt. The pertinence of this adoption can be assessed on the basis of both (1) the characteristics of

the companies, projects and individuals, reported in detail in the paper through the codes, and (2) the quotes which provide rationale for decisions made. Academics may learn two different things. On the one hand, the low adoption of research results should trigger reflection to academics in general. Aspects such as scalability, contextualization and learning curve are often not considered in detail when proposing new methods and models for requirements specification. On the other hand, some of the reported results and subsequent discussion may trigger further empirical studies to be made by academics. For instance, replicating this study with the same research questions over a different population would contribute to strengthen the findings and better contextualize them.

Appendix

Appendix 1 Details of previous research

This appendix provides an overview of the characteristics of the previous research papers that we included in our study (see Table 4). We also provide a summary of the findings of each of these papers that are of interest to the present study (Table 5).

Table 4 Empirical studies on industrial practices in the field of requirements specification

	Study	Topics with results relevant to requirements specification
[22]	<p><i>Goal:</i> To examine the state of the practice of RE in Chinese companies</p> <p><i>Type of study:</i> questionnaire</p> <p><i>Population:</i> 377 subjects from 237 companies or research organizations of different domains and sizes</p> <p><i>Country:</i> China</p>	<p><i>Specification languages</i></p> <p><i>Specification practices</i></p>
[8]	<p><i>Goal:</i> to identify RE practices in relation to security</p> <p><i>Type of study:</i> questionnaire</p> <p><i>Population:</i> 377 subjects from 237 companies of different domains and sizes (same population as in [22])</p> <p><i>Country:</i> China</p>	<i>RE specification practices in relation to security</i>
[32]	<p><i>Goal:</i> to study the state of the practice in RE</p> <p><i>Type of study:</i> interviews</p> <p><i>Population:</i> 7 subjects from 3 companies and a public authority dealing with safety systems in the nuclear energy industry</p> <p><i>Country:</i> Finland</p>	<p><i>Specification languages</i></p> <p><i>Specification challenges</i></p>
[38]	<p><i>Goal:</i> to identify key industry needs with respect to RE</p> <p><i>Type of study:</i> Interviews and Pre- and post-interview questionnaires</p> <p><i>Population:</i> 17 subjects with a clear view of RE needs from 7 companies in the domain of embedded systems</p> <p><i>Country:</i> Germany</p>	<p><i>Specification languages</i></p> <p><i>Specification challenges</i></p>
[3]	<p><i>Goal:</i> to investigate the elicitation, analysis, negotiation, management, and handling of QRs in industry</p> <p><i>Type of study:</i> Interviews</p> <p><i>Population:</i> 22 subjects (11 product managers, 11 project leaders) from 11 software companies</p> <p><i>Country:</i> Sweden</p>	<i>Quantification of quality requirements</i>
[20]	<p><i>Goal:</i> to investigate the evolution of requirements engineering practices during a ten-year period</p> <p><i>Type of study:</i> three online questionnaires (2003, 2008, 2013)</p> <p><i>Population of the three iterations:</i> 194, 93, and 247 subjects</p> <p><i>Country:</i> USA</p>	<i>Evolution of specification Techniques</i>
[18]	<p><i>Goal:</i> To examine the RE practices in Agile methods and the challenges that teams face when working with RE in such a context</p> <p><i>Type of study:</i> systematic review of empirical research publications</p> <p><i>Population:</i> 21 identified papers from 2002 to 2013</p> <p><i>Country:</i> Worldwide</p>	<i>Specification challenges in Agile methods</i>
[17]	<p><i>Goal:</i> to investigate the practices for writing, maintaining, and linking requirements- and acceptance test documentation</p> <p><i>Type of study:</i> interviews</p> <p><i>Population:</i> 20 interviews with practitioners from 17 business units in 15 companies working in different domains</p> <p><i>Countries:</i> 7 countries (specific countries not stated)</p>	<p><i>Specification languages</i></p> <p><i>RE tools for specifications</i></p> <p><i>Relation of specification with testing</i></p>
[4]	<p><i>Goal:</i> to identify challenges and practices of non-functional requirements specification in Agile projects</p> <p><i>Type of study:</i> interviews</p> <p><i>Population:</i> 12 subjects from 4 companies of different domains and sizes</p> <p><i>Country:</i> 4 countries</p>	<p><i>Specification practices and challenges in Agile projects</i></p> <p><i>Guidelines for documenting non-functional requirements</i></p>
[28]	<p><i>Goal:</i> to investigate the state of the practice on requirements reuse and requirement patterns</p> <p><i>Type of study:</i> questionnaire</p> <p><i>Population:</i> 71 participants</p> <p><i>Country:</i> Worldwide</p>	<i>Use of templates in RE</i>
[21]	<p><i>Goal:</i> to identify specific problems/challenges in automotive RE with respect to communication and organizational structure</p> <p><i>Type of study first run:</i> interviews</p> <p><i>Population first run:</i> 14 interviews (2 companies)</p> <p><i>Type of study second run:</i> questionnaires</p> <p><i>Population second run:</i> 31 valid questionnaire responses</p> <p><i>Country:</i> 4 countries</p>	<i>Automotive RE specification challenges</i>

Table 4 (continued)

	Study	Topics with results relevant to requirements specification
[1]	<p><i>Goal:</i> to identify aspects of requirements engineering that impact system development speed and identify practices that increase system development speed</p> <p><i>Type of study:</i> interviews</p> <p><i>Population first step:</i> 20 participants (2 companies)</p> <p><i>Population second step:</i> 12 participants (3 from the previous step)</p> <p><i>Country:</i> not stated</p>	<i>Specification aspects</i>
[2]	<p><i>Goal:</i> to identify challenging situations in the engineering of QRs in an Agile context and industrial practices so as to be able to mitigate the effects of such situations</p> <p><i>Type of study:</i> semi-structured open-ended interviews</p> <p><i>Population:</i> 17 interviews</p> <p><i>Country:</i> The Netherlands</p>	<p><i>Specification challenges</i></p> <p><i>Specification practices</i></p>
[39]	<p><i>Goal:</i> to study the state of the practice in RE</p> <p><i>Type of study:</i> questionnaires</p> <p><i>Population first iteration:</i> 58 companies (one participant per company)</p> <p><i>Country first iteration:</i> Germany</p> <p><i>Population second iteration:</i> 228 companies (one participant per company)</p> <p><i>Countries second iteration:</i> 10 countries worldwide</p>	<p><i>RE standards for specification</i></p> <p><i>Requirements classification</i></p> <p><i>Specification templates</i></p> <p><i>RE tools for specification</i></p> <p><i>Specification challenges</i></p>
[11]	<p><i>Goal:</i> to investigate which standards and templates are known and used in RE</p> <p><i>Type of study:</i> questionnaire</p> <p><i>Population:</i> 90 valid responses</p> <p><i>Country:</i> Worldwide</p>	<i>Use of standards and templates in specification</i>

Table 5 Summary of relevant results obtained in the previous research included in the present study

Specification	References	Results	
<i>Representation</i>	[22]	NL is not the dominant notation; notwithstanding this, graphical elements are widely used UML models are used by more than half of the participants Multiple specification techniques are used in combination in individual projects RE status is different depending on the type of organization (for example, multi-national company vs government-owned enterprises)	
	[8]	Modelling notations are more frequently used than NL sentences and tables More than half of the respondents use UML to express requirements A majority of the respondents use non-UML models for security requirements	
	[32]	Requirements are primarily represented as NL sentences, usually at a relatively detailed level	
	[38]	NL is dominant in the specification of requirements Wide use of models in the specific context of embedded systems Goal models are rarely used	
	[3]	QRs are specified in a quantifiable manner by most of the subjects Interdependencies are not often represented	
	[20]	The majority of participants use NL, with a small increase as the practitioner's professional experience increases Formal specification is not commonly used (slight trend decrease over the years) A third of the study's participants use diagrammatic languages as UML	
	[17]	Traditional representation (structured sentences) are primarily used by companies following a Waterfall approach User stories are used by every company following an Agile approach, sometimes complemented with sticky notes, index cards, or detailed explanations	
	[28]	NL and textual artefacts are highly dominant, with a similar frequency of use UML is used by more than half of the interviewees	
	[1]	Requirements are typically expressed in prose There is the potential to use model-based RE, but replacing textual requirements with models is not always an option	
	[2]	QRs are specified by some participants as a checklist of conditions to be met QRs are specified by other participants in response to user stories QRs are expressed with varying degrees of precision, depending on the type of project	
	[39]	Used by more than a third of participants (in descending frequency below): Free-form textual structured requirement lists Semi-formal (UML) use case models Textual structured requirement lists with constraints Free-form use case models and semi-formal (UML) data models About half of the study's participants use quantified textual requirements to document non-functional requirements	
	<i>Templates/standards</i>	[8]	The majority of interviewees reported that they use standard or tailored modelling notation systems in their RE practices
		[28]	Half of the interviewees who implemented the reuse of requirements across projects used templates or patterns to specify the requirements
[39]		Almost all of the interviewees follow an RE internal company standard In many organizations, the standard is tailored at the beginning of the project by the project lead based on previous experiences of the team members	
[11]		General knowledge about RE standards is low Documenting requirements is the most frequently mentioned reason for using standards Templates are used independently of standards	
<i>Structuring</i>	[32]	Hierarchical structures are not necessarily explicit	
	[1]	Because of the complexity of automotive systems, requirements are generally organized in a number of abstraction levels handled by different roles	

Table 5 (continued)

Specification	References	Results
<i>Tools</i>	[22]	Tools are more accessible and available in multi-national corporations that use project management and RE specific tools General-purpose tools are used in domestic private companies More than half of the participants do not use a tool
	[32]	Participants use document management systems instead of RE tools Identified a need for dedicated tools for requirements management
	[20]	Requirements management tools are not used, but other general-purpose tools, which cover different project activities, are used Emergence of JIRA
	[17]	Agile companies primarily use Jira and/or Confluence Waterfall companies prefer to use MS Word and/or Excel tools
<i>Challenges</i>	[22]	More than a third of the study's participants do not usually include non-functional requirements in specifications
	[8]	Security requirements are often left undocumented
	[32]	From NL to multi-dimensional representation of requirements, including multi-levels of hierarchies, dependencies, and cross-cutting concerns Do not only adopt RE-related tools but also integrate them into existing systems The ageing of tools becomes an issue
	[6]	Existence of communication gaps between stakeholders and requirements engineers that may result in incomplete, over-scoped, or inconsistent requirements specifications
	[38]	More intensive use of models in RE, particularly for safety-critical systems
	[18]	In Agile global software projects, when developers are distributed across several sites, working only with user stories may be insufficient
	[17]	Problems, for example, misunderstood requirements or underestimated cost, caused by the specification of requirements made by managers User stories are not sufficient to support tasks dependent on the specification
	[4]	Non-functional requirements documented in diverse ways in Agile projects It is challenging to establish the traceability of non-functional requirements
	[21]	Including contextual knowledge in specifications so as to provide context to developers and tester Dealing with specifications at different abstraction levels to avoid challenges related to traceability and possible mismatches between levels Having insufficient resources for maintaining requirements
	[1]	Current requirements-centric culture (a culture of over-specification or an over-emphasis on requirements specification) does not always pay off and constrains development speed
	[2]	Difficult to model QRs in Agile projects associated with a lack of formal modelling of detailed requirements makes the process of verifying the QRs more challenging The specification of QRs associated with user stories hinders the establishment of a shared understanding of QR specifications in Agile contexts
	[39]	Under-specification of requirements that remain too abstract and allow for various interpretations is the most frequently mentioned challenge

Appendix 2 Description of the interviewees

See Tables 6, 7, and 8.

Table 6 An overview of the study's participants

ID	Highest level of educational attainment	Years in industry	Years of experience working at a university or research lab	Job position	Years in position	Years in organization
S1	BSc in computer science	15	3	Business analyst	3	3
S2	MSc in computer science	15	3	Project manager	≈5	10
S3	BSc in information systems	20	≈4	System analyst	6	≈9
S4	BSc in computer science	13	3	Requirement analyst	13	13
S5	MSc in computer science	25	5	Requirement analyst	2.5	4
S6	BSc in information systems	20	0	System manager	15	20
S7	MSc in computer science	19	5	System manager	6	19
S8	BSc in computer science	15	0	Senior project manager	15	15
S9	BSc in energy systems	20	0	Senior business consultant	6	6
S10	MSc in computer science	16	0	Senior developer	9	9
S11	MSc in software engineering	17	5	Unit manager	0	0
S12	MSc in business	12	≈5	Solution designer	≈8	≈9
S13	BSc in computer science	23	0	Business analyst	14	14
S14	PhD in food engineering	10	15	System engineer	2	5
S15	MSc in chemical engineering	10	0	System engineer	0.25	4.5
S16	BSc in telecommunication	25	0	Product manager	5	19
S17	MSc in industrial engineering	8	0	System engineer	8	8
S18	MSc in vision and robotics	9	5	Project leader	2	2
S19	MSc in electrochemistry and electronic sensors	3	3	Lead engineer	0.5	2
S20	PhD in civil engineering	23	10	Software, manufacturing, & electrical engineer	1.5	16
S21	MSc in computer science	21	0	Senior consultant	5	12
S22	BSc in interaction design	9	3	Senior consultant	3	9
S23	BSc in quality engineering	15	4	Assignment manager	5	4.5
S24	BSc in mathematics, physics, and computer science	26	4	Requirements engineer	3.5	3.5

Table 7 Companies included in the study

Organization	Practitioner	Number of employees	Main business area
A	S1, S2	≈2,000 worldwide	ITD of a telecommunication operator
B	S3, S4	≈900	SCC in the public sector
C	S5	≈350	SH (UI platforms for Symbian-based smartphones)
D	S6, S7	≈115,000 worldwide	SH (telecommunications products)
E	S8, S9	≈68,000 worldwide	SCC
F	S10, S11	50	SCC
G	S12, S13	800	SCC (telecommunication products)
H	S14, S15	≈23,000 worldwide	ITD of a Tetra Bricks manufacturer
I	S16, S17	≈150,000 worldwide	SH (power and automatization systems)
J	S18, S19, S20	≈20,000	ITD of a car manufacturer
K	S21, S22	1,200	SCC
L	S23, S24	Not sure	public transport administration

Table 8 Projects included in the empirical study

ID project	ID Subject	Project Main functionality	Project Domain	Project duration (in years)	Project number employees	Project costs (in €)	Project methodology
<i>P1</i>	S1	Getting customer feedback	Messaging system	1	≈10	Not sure	Waterfall
<i>P2</i>	S2	Webshop for acquiring phones and contracts with a carrier	Website	1	≈10	Not sure	Waterfall
<i>P3</i>	S3	Translating a website into english	Website	1	10–12	≈0.5 Million	Agile
<i>P4</i>	S4	Management of the social security rights of children	Website	1.5	≈35	≈6.0 Million	Agile
<i>P5</i>	S5	OS for a specific smart- phone taking into account the carrier's restrictions	Mobile OS	0.5	≈100	Not sure	Waterfall
<i>P6</i>	S6	Carrier system to track the users' consumption	Machine to machine system	0.25	7	Not sure	Agile
<i>P7</i>	S7	Providing services to customers (charging, changing plan, consumption, etc.)	Carrier business support system	2.5	Not sure	Not sure	Agile
<i>P8</i>	S8	Managing consumption energy levels measured by energy companies	Energy measurement system	1.5	≈2	Not sure	Waterfall
<i>P9</i>	S9	System for an energy company involving the contract and offering module	Business support system	2	Not sure	≈1.3 Million	Waterfall
<i>P10</i>	S10	System for a carrier involving big data, call data management, etc	Carrier internal system	1	100	Not sure	Agile
<i>P11</i>	S11	Webshop for acquiring public transport system tickets	Website	0.33	5	Not sure	Agile
<i>P12</i>	S12	Offering roaming services to customers	Carrier business support system	≈1.5	≈20	≈1.5 Million	Waterfall
<i>P13</i>	S13	Managing customer calls in a customer service centre	Carrier internal system	1.5	25	Not sure	Waterfall
<i>P14</i>	S14	Modifying an existing machine (and its system) to make it more productive	Embedded system	4	35	≈9.0 Million	Waterfall
<i>P15</i>	S15	A new machine (and system) for a new package	Embedded system	0.75	10	Not sure	Waterfall
<i>P16</i>	S16	Managing control and safety processes	Embedded system	≈1.5	≈200	≈6.0 Million	Agile
<i>P17</i>	S17	Controlling the machines of a sugar factory	Embedded system	1.5	6	Not sure	Waterfall

Table 8 (continued)

ID project	ID Subject	Project Main functionality	Project Domain	Project duration (in years)	Project number employees	Project costs (in €)	Project methodology
<i>P18</i>	S18	Managing the different functionalities of a car	Embedded system	≈3	≈60	Order of Billions	Waterfall
<i>P19</i>	S19	Controlling the battery charge in electric cars	Embedded system	2	20	Not sure	Waterfall
<i>P20</i>	S20	Controlling the machines for producing a car	Embedded system	6–7	×000	Order of Billions	Waterfall
<i>P21</i>	S21	Checking films, booking tickets, etc. for a cinema company	Mobile app	1	18	≈1.5 Million	Waterfall
<i>P22</i>	S22	Integrating payment services	Mobile app, website	0.25	12	Not sure	Agile
<i>P23</i>	S23	Specifying a tunnel's construction safety system	Construction	10	Not sure	≈2.5 Billion	Waterfall
<i>P24</i>	S24	Specifying a tunnel's construction safety system	Construction	10	Not sure	≈2.5 Billion	Waterfall

Appendix 3 Coding of interview answers

Categories	Codes	Categories	Codes
			Headers and Sections, Tags, Epics
		Flexibility	
RQ1			Change from project to project, Fixed structure
NL sentences	NL sentences	RQ4	
Textual artefacts	User stories, Use cases, Test cases	Which tool	
Graphical elements	General, Tables, Pictures, Diagrams, Mockups	MS Office	Word, Excel, Power Point, Visio
Models	General, Domain models, Class diagrams, Block diagrams, Activity diagrams, State diagrams, Sequence diagrams, Stateflows, Business process models	Requirement management tools	Cognition Cockpit, Safety Case, FocusPoint, IBM Doors
		Modelling Tools	SAP PowerDesigner, IBM Rational Rose, Sparx Enterprise Architect, CGS Electra
RQ2		Big Systems	SIEMENS Teamcenter, IBM Rational ClearCase /ClearQuest
Provenance		Project management tools	JIRA, Mingle, MS Team Foundation Center
	Organization, Standard, Methodology, Tool, Not sure	Purpose	
Purpose			Management, Generation of documents, Versioning, Communication, Traceability, Verification and validation, Writing, Diagramming, Annotating, Export to other tools
	Document layout, Requirement attributes, How-to-write guidelines, Quality assurance		
RQ3		RQ5	
Structuring concept			Ambiguity, Incompleteness, Inconsistency, Lack of traceability, Lack of verifiability, Requirements reuse, Level of detail, Short-term vision, Team spread, No challenge
	Main functionalities, Characteristic, Parts of the system, Artefact, Workpackages, Model elements, Undetermined		
Structuring element			

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00766-023-00399-7>.

Acknowledgements This paper has been funded by the Spanish Ministerio de Ciencia e Innovación under project/funding scheme PID2020-117191RB-I00/AEI.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ågren SM, Knauss E, Heldal R, Pelliccione P, Malmqvist G, Bodén J (2019) The impact of requirements on systems development speed: a multiple-case study in automotive. *Requirements Eng* 24(3):315–340
- Alsaqaf W, Daneva M, Wieringa R (2019) Quality requirements challenges in the context of large-scale distributed agile: an empirical study. *Inf Softw Technol* 110:39–55
- Berntsson-Svensson R, Gorschek T, Regnell B, Torkar R, Shahrokni A, Feldt R (2012) Quality requirements in industrial practice: an extended interview study at eleven companies. *Trans Softw Eng* 38(4):923–935
- Behutiye W, Karhapää P, Costal D, Oivo M, Franch X (2017) Non-functional requirements documentation in agile software development: challenges and solution proposal. In: *Proceedings of the 18th International Conference on Product-Focused Software Process Improvement (PROFES 2017)*, pp 515–522
- Berry DM, Kamsties E (2004) Ambiguity in requirements specification. In: *Perspectives on Software Requirements*. The Springer International Series in Engineering and Computer Science, Springer, 753
- Bjarnason E, Wnuk K, Regnell B (2011) Requirements are slipping through the gaps—a case study on causes & effects of communication gaps in large-scale software development. In: *Proceedings of the 19th IEEE International Requirements Engineering Conference (RE 2011)*, pp 37–46
- Dardenne A, van Lamsweerde A, Fickas S (1993) Goal-directed requirements acquisition. *Sci Comput Program* 20(1–2):3–50
- Elahi G, Yu E, Li T, Liu L (2011) Security requirements engineering in the wild: a survey of common practices. In: *Proceedings of the 35th IEEE Annual Computer Software and Applications Conference (COMPSAC 2011)*, pp 314–319
- Ferrari A, Dell’Orletta F, Esuli A, Gervasi V, Gnesi S (2017) Natural language requirements processing: a 4D vision. *IEEE Softw* 34(6):28–35
- Franch X, Palomares C, Quer C (2020) Industrial practices on requirements reuse: an interview-based study. In: *Proceedings of the 26th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2020)*. Lecture Notes in Computer Science, Springer, Cham, 12045
- Franch X, Palomares C, Gorschek T (2021) On the requirements engineer role. *Commun ACM* 64(6):69–75
- Franch X, Glinz M, Mendez D, Seyff N (2022) A study about the knowledge and use of requirements engineering standards in industry. *Trans Softw Eng* 48(9):3310–3325
- Franch X, Mendez D, Vogelsang A, Heldal R, Knauss E, Oriol M, Travassos G, Carver JC, Zimmermann T (2022) How do practitioners perceive the relevance of requirements engineering research? *Trans Softw Eng* 48(6):1947–1964
- Garousi V, Borg M, Oivo M (2020) Practical relevance of software engineering research: synthesizing the community’s voice. *Empir Softw Eng* 25:1687–1754
- Gervasi V, Zowghi D (2005) Reasoning about inconsistencies in natural language requirements. *ACM Trans Softw Eng Methodol* 14(3):277–330
- Glinz M, van Loenhoud H, Staal S, Bühne S (2020) Handbook for the CPRE foundational level according to the IREB standard, IREB
- Hotomski S, Charrada EB, Glinz M (2016) An exploratory study on handling requirements and acceptance test documentation in industry. In: *Proceedings of the 24th IEEE International Requirements Engineering Conference (RE 2016)*, pp 116–125
- Inayat I, Salim SS, Marczak S, Daneva M, Shamshirband S (2015) A systematic literature review on agile requirements engineering practices and challenges. *Comput Hum Behav* 51:915–929
- ISO/IEC/IEEE 29148 (2011) ISO/IEC/IEEE 29148:2011(E), Systems and software engineering—life cycle processes—requirements engineering. International Organization for Standardization
- Kassab M (2015) The changing landscape of requirements engineering practices over the past decade. In: *Proceedings of the 5th IEEE International Workshop on Empirical Requirements Engineering (EmpiRE 2015)*, pp 1–8
- Liebel G, Tichy M, Knauss E, Ljungkrantz O, Stieglbauer G (2018) Organisation and communication problems in automotive requirements engineering. *Requirements Eng* 23(1):145–167
- Liu L, Li T, Peng F (2010) Why requirements engineering fails: a survey report from China. In: *Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010)*, pp 317–322
- Lutters WG, Seaman C (2007) Revealing Actual documentation usage in software maintenance. *Inf Softw Technol* 49(6):576–587
- Martins LEG, Gorschek T (2017) Requirements engineering for safety-critical systems: overview and challenges. *IEEE Softw* 34(4):49–57
- Mavin A, Wilkinson P (2019) Ten years of EARS. *IEEE Softw* 36(5):10–14
- Méndez Fernández D, Wagner S (2015) Naming the pain in requirements engineering: a design for a global family of surveys and first results from germany. *Inf Softw Technol* 57:616–643
- Oates BJ (2006) *Researching Information Systems and Computing*. Sage Publications
- Palomares C, Quer C, Franch X (2017) Requirements reuse and requirement patterns: a state of the practice survey. *Empir Softw Eng* 22(6):2719–2762
- Palomares C, Franch X, Quer C, Chatzipetrou P, López L, Gorschek T (2021) The state-of-practice in requirements elicitation: an extended interview study at 12 companies. *Requirements Eng* 26(2):273–299
- Pohl K (2010) *Requirements engineering: fundamentals*. Springer, Principles and Techniques
- Patton MQ (2002) *Qualitative research and evaluation methods*. Sage Publications

32. Raatikainen M, Männistö T, Tommila T, Valkonen J (2011) Challenges of requirements engineering—a case study in nuclear energy domain. In: Proceedings of the 19th IEEE International Requirements Engineering Conference (RE 2019), pp 253–258
33. Renault S, Mendez-Bonilla O, Franch X, Quer C (2009) PABRE: pattern-based requirements elicitation. In: Proceedings of the Third International Conference on Research Challenges in Information Science (RCIS 2009), pp 81–92
34. Robertson S, Robertson J (1999) Mastering the requirements process. Addison-Wesley
35. Robson C (2002) Real world research: a resource for social scientists and practitioner-researchers. Blackwell Publishers Inc.
36. Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. *Empir Softw Eng* 14(2):131–164
37. Saldana J (2009) The coding manual for qualitative research. Sage Publications
38. Sikora E, Tenbergen B, Pohl K (2012) Industry needs and research directions in requirements engineering for embedded systems. *Requirements Eng* 17(1):57–78
39. Wagner S, Mendez Fernandez D, Kalinowski M, Felderer M, Mafra P, Vetrò A, Conte T, Christiansson M-T, Greer D, Lassenius C, Männistö T, Nayebi M, Oivo M, Penzenstadler B, Pfahl D, Prikładnicki R, Ruhe G, Schekelmann A, Sen S, Spinola R, de la Vara JL, Tuzcu A, Wieringa R, Winkler D (2019) Status quo in requirements engineering: a theory and a global family of surveys. *Trans Softw Eng Methodol* 28(2):9
40. Wiegers K, Beatty J (2013) Software requirements, 3rd edition. Microsoft Press.
41. Wohlin C, Runeson P, Host M, Ohlsson MC, Regnell B, Wesslen A (2012) Experimentation in software engineering: an introduction. Springer
42. Yu ESK, Amyot D, Mussbacher G, Franch X, Castro J (2013) Practical applications of i* in industry: the state of the art. In: Proceedings of the 21st IEEE International Requirements Engineering Conference (RE 2013), pp 366–367

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.