IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 23, 2021, accepted March 3, 2021, date of publication March 8, 2021, date of current version March 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064424

A Qualitative Study on Non-Functional Requirements in Agile Software Development

ALEKSANDER JARZĘBOWICZ AND PAWEŁ WEICHBROTH
Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland
Corresponding author: Aleksander Jarzębowicz (olek@eti.pg.edu.pl)

ABSTRACT **Context:** Widespread use of agile software development (ASD) methods can be observed nowadays. Non-functional requirements (NFRs) are often reported to be a problematic issue for agile projects, since ASD methods tend to neglect NFRs while focusing on incremental delivery of functional features. Despite that, only very few studies have explored the requirements engineering practices used in ASD and dedicated particularly to NFRs. **Objective:** We aimed to identify and investigate the practices used in industrial ASD projects to identify, elicit and document NFRs. **Method:** We conducted a systematic literature review (SLR) and used its findings to prepare an interview guide. We then conducted a series of semi-structured interviews with 10 experienced practitioners. **Results:** The SLR revealed a number of strategies related to the timing of NFR identification as well as 13 elicitation practices and 20 documentation techniques. At least some of these findings show discrepancies between ASD theory and practice. The interviews provided a more in-depth understanding of the practices used, and their context. The main findings from the interviews include: practitioners' attempts to start identifying NFRs early in the project, a lack of elicitation techniques aimed at NFRs only, and different choices about documentation techniques, including the additional techniques introduced to cope specifically with NFRs. **Conclusions:** It was not confirmed that requirements engineering activities related to NFRs are perceived by ASD practitioners as a problem, as they developed effective practices to deal with this issue. Moreover, our findings show that different approaches to NFRs can be used and give satisfactory results.

INDEX TERMS Requirements engineering, non-functional requirements, agile software development, systematic literature review, interviews.

I. INTRODUCTION

During the last two decades, agile software development (ASD) methods have gained recognition and have been widely adopted by the IT industry worldwide [1]–[3]. Such methods postulate emphasis on working software and have a tendency to minimize documentation, but the importance of stakeholder management and requirements engineering in the agile context is still well recognized [4].

The agile approach however introduces its own values and practices which are different than those used in more traditional approaches, which also includes somewhat different requirements engineering practices [5] - thus the term of ''agile requirements engineering'' (ARE) was coined [6] and now is in common use [7]–[9].

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaobing Sun.

ARE is reported to reduce several requirements-related problems *e.g.* coping with changing requirements [7], and inadequate communication with the customer [6], but at the same time it introduces its own challenges [7], [8], [10] that could and should be addressed in order to continuously improve software development processes.

Some of the most frequently quoted ARE challenges concern non-functional requirements (NFRs), also known as quality requirements [6]–[9], [11]. In particular, the problems of: neglecting NFR, while focusing on functionality [6]–[8], too-minimal requirements documentation to capture NFRs [6], [8] and insufficiency of available ARE techniques to deal with NFRs [7], [9], [11] are reported as problems encountered in practice. Such challenges could result in a major obstacle in technology acceptance and its further use, since there are numerous examples showing that insufficient system quality leads to its abandonment by the users [12]–[14] and NFRs are the main way of expressing quality expectations.

Requirements engineering is a holistic process aimed at capturing all requirements and their interdependencies, thus it is hardly possible to design a separate and isolated process for a specific requirements category like NFRs. However, it is still possible to apply various techniques and practices in the requirements engineering process, including dedicated techniques developed to cope with particular categories of requirements. It is especially worth trying in the situation when more universal techniques turn out to be insufficient.

Such cases are reported by a number of sources that discuss the limitations of several practices and techniques commonly used in ARE in the particular context of capturing NFRs. NFRs are considered less obvious to stakeholders and more difficult to express by them during, *e.g.* unstructured interviews, which results in omissions or misunderstandings of NFRs [15]. For that reason, additional elicitation techniques are proposed, *e.g.* the use of an NFR pattern catalogue to express NFRs in a clear and unambiguous way [16] or organization of dedicated NFR-oriented workshops [17].

Similarly, limitations of some ARE documentation techniques with respect to representation of NFRs are reported. Such concerns are voiced for techniques like user story [18] and story card [19], and in order to address such issues, new or adopted documentation techniques, designed with NFRs in mind, like, *e.g.* W8 Story Cards [20] are proposed. Such examples provide evidence about likely cases in which NFRs would be elicited or documented in a different way from FRs and serve as a motivation for a study focused on NFRs in agile projects.

Despite the frequently mentioned NFR-related challenges, there are reports that practitioners recognize the importance of NFRs in agile projects [21], [22] and the success ratio of agile projects is claimed to be higher than for projects using a plan-driven approach [23]. This can indicate that in real-life agile projects, the practitioners establish suitable ARE processes that enable them to effectively capture NFRs and then to provide a software product satisfying such requirements. Therefore, our main goal is to investigate how agile practitioners deal with NFRs and which ARE techniques and practices they use for this purpose.

To narrow down the scope of the planned study and enable a more in-depth focus, we decided to pose the research questions that address the most frequently reported challenges and related areas of ARE. The challenge of ''neglecting NFRs while focusing on functionality'' [7], [8] is an issue corresponding to requirements elicitation activities and indicates the possible deficiencies in elicitation techniques/practices used or their inappropriate use. Some sources [6], [24]–[26] however report that this challenge is caused by ignoring NFRs or treating them superficially in the early phases/iterations of the project, thus indicating that the timing of eliciting NFRs (rather than the elicitation techniques applied) is the crucial issue and thus may be worth investigating. The remaining quoted challenges (''too-minimal requirements documentation to capture NFRs'' [6], [8] and ''insufficiency of available ARE techniques to deal with NFRs'' [7], [9], [11]) both concern requirements documentation techniques and as such the activities of requirements documentation should be selected as the point of interest in our research.

Consequently, we put forward (within the context of ASD) the following three research questions (RQs):

RQ1. When are non-functional requirements identified in the project life cycle?

RQ2. How are non-functional requirements elicited?

RQ3. How are non-functional requirements documented?

Despite the fact that the challenges and solutions related to NFRs in Agile are mentioned in several sources (see Section II for details), a summary of practices and techniques dedicated to NFR identification, elicitation and documentation in Agile projects is still not provided in comprehensive manner by the existing literature. We identified several secondary studies covering research topics like: requirements engineering techniques used in Agile projects [10], [27], [28], ARE practices and challenges [29], requirements engineering challenges and mitigating practices [8], quality of agile requirements specifications [30] and main research areas in ARE [7], [31]. None of them however focused on NFRs and, as a result, they report only individual cases of findings (challenges, practices, techniques) related to NFRs. Only one secondary study with an explicit focus on NFR-related challenges and practices [11] was identified, which however assumes a more specific context (Agile Large Scale Distributed projects) and lists only the practices that provide solutions to specific challenges. Consequently, we believe a research gap exists and we attempt to address it in our study.

The major contributions of this article are as follows: *(i)* an evidence-based review and analysis of the non-functional requirements practices, adopted for ASD projects, *(ii)* a state-of-the-art list of the elicitation and documentation techniques used in an effort to recognize and specify NFRs, and *(iii)* the observations of practitioners regarding key issues on NFRs in their ASD experience.

The remainder of this article is structured as follows. Section II provides the background on the addressed subject. Section III describes the methodology of the study conducted. Section IV presents the findings obtained from the study. Section V reflects on the findings and discusses study limitations. The conclusions and future research directions are given in Section VI.

II. RESEARCH BACKGROUND AND MOTIVATION

This section is devoted to reviewing the basic theoretical foundations of requirements engineering, as well as summarizing other studies conducted in this domain. The findings underpin and motivate our study by identifying a research gap in the contemporary literature, and eventually formulating related research questions.

A. REQUIREMENTS ENGINEERING

Requirement engineering has received much attention in both academia and practice as the key to on-time,

on-budget and on-value delivery of software products [32]. By definition, requirement engineering is a decision-centric activity [33] that produces specifications including both functional (FR) and non-functional (NFR) requirements [34]. By many requirements engineering is considered a key area of the software development process, which can significantly influence the result of the project [35]–[37].

During the software development process, the main focus is usually on the system features, documented as functional requirements (FRs). In other words, the FRs define the behaviors or actions that need to be supported by the system [38]. In contrast, the non-functional requirements (NFRs) relate to system properties which do not directly affect the features provided by the system [39].

A different view on FRs and NFRs is given by the ISO/IEC 25066 standard [40], where both categories are based on user needs and capabilities. Indeed, it appears that nowadays a user is placed in the center of modern software engineering practices that aim to meet identified requirements [41], [42] and track their changes [43]. This seems to be in line with the agile approach and its principles that focus on customer value, incremental, iterative and delivery, intense collaboration with users, small teams, self-organization, and small and continuous improvements [44], [45].

The most common practices such as interviews [46], user stories [47], observations [48], workshops [49] are frequently embedded in iteration-based agile methods. Moreover, the course toward promoting active user participation and involvement has been frequently documented in recent reports and reviews that discuss agile requirements engineering [47], [49], [50].

B. AGILE REQUIREMENTS ENGINEERING (ARE)

Agile has been widely adopted by software vendors [51], and other organizations [3], [46], [52] To point out the reasons explaining the shift from traditional to agile methods, literature reports show that adopting agile methods results in higher team productivity and morale [53], less rework [54], and more efficient defect fixing rates [55]. The current most popular agile method is Scrum, followed by Kanban, Scrumban as well as the practices of eXtreme Programming (XP) [22], [56].

Agile requirements engineering (ARE) is known to be effective and to reduce many requirements engineering problems, but at the same time, there are specific challenges associated with it [8].

The topic of ARE has attracted the attention of researchers and as a result a substantial body of knowledge is available. The following secondary studies that can be considered related works to ours summarize the published research studies on ARE. None of them however covers the research questions we posed in this study.

C. RELATED WORK

Heikkila *et al*. [7] conducted a mapping study aimed at identifying the main research areas concerning requirements engineering in the ASD context. They also searched for the reported benefits of ARE as well as for problems and corresponding solutions related to ARE. Their study mentions the problems of ''Insufficiency of the user story format'' and ''Reliance on tacit requirements knowledge'', which they report as applicable to NFRs (among others). Moreover, the fact of ignoring NFRs is confirmed by their study, though not listed as a separate problem. The recommended solutions include additional requirements documentation and managing requirements traceability. Their work does not specifically focus on NFRs, and besides the above-mentioned issues, no corresponding practices/techniques are mentioned.

Inayat *et al*. [8] published the results of a SLR study dedicated to requirements engineering challenges. They searched for agile practices that address such challenges on the one hand, and additional challenges stemming from agile practices on the other hand. The only NFR-related item reported by this study is ''Neglecting non-functional requirements'' (caused by insufficiency of user story format for such purpose) listed among the ARE challenges and juxtaposed with solutions suggesting the use of structured, more detailed user stories and associated tool support.

A systematic mapping study by Medeiros *et al*. [10] investigated how requirements engineering has been conducted in projects that adopt agile methodologies. Among the more specific research questions posed by this study, ''requirements elicitation techniques'', ''requirements specification techniques'' and ''challenges of ARE'' can be found. Their findings do not however specify which techniques are used to elicit/document NFRs. The long list of identified challenges includes only one NFR-related challenge which expresses that ARE techniques are weak in defining NFRs.

Elghariani and Kama [29] reported a systematic literature review on ARE practices and challenges. Their review was not NFR-oriented and thus it cannot be determined to what extent the reported practices apply to NFRs. Among the challenges, ''Ignoring non-functional requirements'' can be found, together with a suggested solution of using structured user stories to represent NFRs.

An article by Zamudio *et al*. [27] reports a SLR study on requirements engineering techniques in ASD. The authors focus on elicitation techniques reported as used in particular agile methodologies. Such techniques are not in any way analyzed with respect to their applicability to nor effectiveness at eliciting NFRs. In the concluding section, the authors state that ''more empirical results are required to better understand the impact of agile requirements engineering practices *e.g.* dealing with non-functional requirements (...)''.

Schon *et al*. [28] focused on ARE practices and techniques dedicated to requirements elicitation, requirements documentation and requirements management. NFRs are not explicitly considered though, therefore it is not possible to establish on the basis of their SLR study, to what extent the practices/techniques listed are applicable to NFRs. Moreover, the focus of the SLR study was narrowed down to ARE in the more specific context of User-Centered Design (UCD)

activities, which, despite its importance, does not exhaust the topic of ARE practices. They however present some NFR-related problems, namely: neglecting NFRs and a lack of formal acceptance tests for NFRs.

A systematic literature review with a primary focus on NFRs was conducted by Alsaquaf *et al*. [11]. The SLR aimed to summarize the practices used to engineer NFRs in agile large-scale distributed (ALSD) projects, but also to identify (for ARE in general) the NFR challenges and the solutions to cope with the challenge of neglected NFRs. The list of solutions includes 10 practices (plus 3 additional ones dedicated to particular categories of NFRs, *e.g.* security). Some of these practices concern requirements elicitation or requirements documentation, however this set of practices is not supposed to include *e.g.* all NFR-related elicitation techniques, but only those which address the single challenge of neglecting NFRs.

Heck and Zaidman [30] presented the results of a SLR on agile requirements specifications. Their goal was to identify the quality criteria of such specification mentioned in the literature. One of the identified criteria addresses the inclusion of NFRs in the specification. The associated recommendations state that NFRs should not be overlooked and that meetings to discuss NFRs should be arranged as early as possible. Three documentation techniques applicable to specifying NFRs are also mentioned. This study, due to its different focus does not exhaust the topic of NFR elicitation and documentation techniques nor the issue of positioning NFR identification in the software project life cycle.

Curcio *et al*. [31] conducted a systematic mapping study to identify ARE research topics discussed in the scientific literature and the remaining research gaps. Moreover, they summarized the challenges (obstacles) associated with ARE activities. As no specific distinction of NFRs was made, the research topics listed (*e.g.* requirements sources or elicitation techniques) consider various categories of requirements in a joint manner. The list of challenges does not include anything NFR-specific, but a general gap of insufficient support for engineering NFRs in agile methods is reported.

D. SUMMARY OF PREVIOUS STUDIES' FINDINGS

To summarize, none of the above-mentioned studies except [11] focuses on NFRs. The study by Asaquaf *et al*. [11] lists ARE practices and techniques addressing the challenge of neglecting NFRs. In the other studies, a small number of particular NFR-related practices and techniques can be found. No study however provides a comprehensive answer to any of our RQs, concerning: the timing of NFR identification, the NFR elicitation practices and the NFR documentation techniques.

The quoted literature points out the issues that provide a rationale for our research though. A general research gap on NFRs in ARE is identified [27], [31]. Moreover, several ARE challenges related to engineering NFRs are reported [7], [8], [11], [28]. In particular, the problem of NFRs being neglected, while focusing on functionality, is frequently mentioned [7], [8], [11], [28], [29]. Also, too-minimal requirements documentation to capture NFRs [8], [30] and insufficiency of the requirement documentation techniques used in ARE [7], [11] are often reported as encountered challenges. It shows that engineering NFRs in ASD is a problematic task and that ARE practices addressing such task are not sufficiently recognized.

Some solutions to problems regarding NFRs have been described in the literature, *e.g.* [8], [9], [11], [57], but no study has yet provided a comprehensive analysis of the practices and techniques used to deal with NFRs and related challenges in an agile context.

For this reason, we designed our research study. We aim to identify when ARE activities related to NFRs should start and what particular elicitation practices and documentation techniques are applied to engineer NFRs. We start with a systematic literature review to identify the proposals published in the scientific literature and then conduct a series of interviews with experienced practitioners to get a more in-depth understanding of how NFRs are addressed in the industry and find out which proposals found in the literature are used in industrial projects and how their effectiveness is perceived.

III. METHODOLOGY

In order to answer the RQs defined in Section I, we chose a qualitative approach which encompassed two subsequent steps - a systematic literature review (SLR) and a series of interviews with agile practitioners. In the first step, we intended to identify and record the practices described in the literature. The results of the SLR served as an input to the second step - we used them to develop the guidance protocol for semi-structured interviews with agile practitioners.

A. SYSTEMATIC LITERATURE REVIEW

The SLR study was aimed to identify the relevant NFR-related practices and techniques reported in the literature. We planned this study using the guidelines by Kitchenham and Charters [58].

1) PLANNING THE REVIEW

The process we planned (depicted in Fig. 1) comprised of 3 main phases:

1) Selection of a publication database to be used and definition of the keywords and search criteria. Execution of the search.
2) Manual review of the titles, keywords and abstracts of the articles retrieved from the search to exclude those not related to the topic of NFRs in an agile context.
3) Manual review of each remaining article's full text, and the final decision on whether to include it or not. Identification of information pieces relevant to the RQs and assigning codes to them.

As part of planning activities we also defined the inclusion and exclusion criteria.

Inclusion criteria: *(I1)* peer-reviewed articles; *(I2)* articles in English; *(I3)* articles published since 2008 (to include all

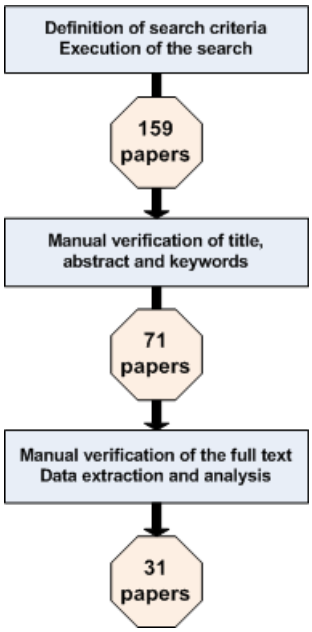

FIGURE 1. SLR phases and included sources.

works published in the 12 years before the conduction of the SLR, as we aimed to identify the current state of the art and not to analyze its evolution over time); *(I4)* articles related to the software engineering domain; *(I5)* articles concerning agile requirements engineering (ARE) and NFRs.

Exclusion criteria: *(E1)* articles not providing any information about ARE activities concerning NFRs; *(E2)* articles not available for download; *(E3)* articles dedicated to secondary studies (systematic literature reviews, systematic mapping studies); *(E4)* articles reporting the same results covered by another included source (in such cases the most comprehensive article was included).

2) PHASE 1

We chose to rely on a single publication database, as the SLR was not supposed to provide final answers but intermediary results to be used in the interview study. We used Elsevier's Scopus database for this purpose.

Being aware that sources dedicated to this topic of interest are rather scarce, we performed some initial searches before planning the SLR. This led us to a decision to cast a wider net and try to identify all of the sources focusing on NFRs in ASD context, thus we used more generic keywords instead of those exactly matching our RQs (*e.g.* ''elicitation'', ''gathering'', ''obtaining'', in the case of RQ2).

The following search string was used (explanation of its parameters and mappings to the inclusion criteria is given in Table 1). As Scopus does not allow to search full texts we had to choose the most comprehensible one from the available options (titles, keywords and abstracts of articles). The types of the documents (Table 2) match peer-reviewed articles. The specification of subject areas (Table 3) resulted from our knowledge of how some sources (especially series that include conference proceedings as their volumes) are classified by Scopus.

TABLE 1. The General Search Query Criteria.

Criterion	Search parameter
Search in article titles, abstracts and keywords	TITLE-ABS-KEY
Only articles written in English (I2)	LIMIT-TO (LANGUAGE, "English")
Only articles published since 2008 (I3)	PUBYEAR > 2007
Must include keywords related to agile approach (I5)	agile OR scrum OR lean OR xp OR kanban
Must include keywords related to NFR (I5)	nfr OR "non-functional requirements" OR "quality requirements"

TABLE 2. The Inclusion Criteria for Document Type.

Document Type (I1)	Search parameter
Conference paper	LIMIT-TO (DOCTYPE, "cp")
Article	LIMIT-TO (DOCTYPE, "ar")
Chapter	LIMIT-TO (DOCTYPE, "ch")

TABLE 3. The Inclusion Criteria for Subject Area.

Subject area (I4)	Search parameter
Computer Science	LIMIT-TO (SUBJAREA, "COMP")
Engineering	LIMIT-TO (SUBJAREA, "ENGI")
Mathematics	LIMIT-TO (SUBJAREA, "MATH")
Business, Management and Accounting	LIMIT-TO (SUBJAREA, "BUSI")
Decision Sciences	LIMIT-TO (SUBJAREA, "DECI")

Search string:

TITLE-ABS-KEY ((agile OR scrum OR lean OR xp OR kanban) AND (''non-functional requirements'' OR ''quality requirements'' OR nfr)) AND (LIMIT-TO (DOCTYPE, ''cp'') OR LIMIT-TO (DOCTYPE,''ar'') OR LIMIT-TO (DOCTYPE, ''ch'')) AND (LIMIT-TO (SUBJAREA,''COMP'') OR LIMIT-TO (SUBJAREA,''ENGI'') OR LIMIT-TO (SUBJAREA, ''MATH'') OR LIMIT-TO (SUBJAREA,''BUSI'') OR LIMIT-TO (SUBJAREA,''DECI'')) AND PUBYEAR > 2007 AND (LIMIT-TO (LANGUAGE, ''English''))

The search was executed on November 28th 2019. Despite the use of several alternative keywords, the search returned only 159 articles. This confirmed our initial suspicions that the topic of ''NFR in ASD'' is seldom addressed in the scientific literature at least as the main theme, because we are aware of articles that mention NFRs as one of numerous aspects of ASD (*e.g.* those presented in Section II).

3) PHASE 2

The results retrieved by the Scopus search engine (which include the title, keywords and abstract of each article found) were manually reviewed. This task was conducted by the first author. It allowed the findings to be verified against the I5 criterion more precisely than in the case of the automated search, and articles that reported nothing on ARE, but, *e.g.* on an architectural design satisfying a particular NFR, to be

rejected. Also, several articles referring to ''quality requirements'' turned out to interpret this term as ''well-documented or valid requirements'' instead of ''requirements regarding system quality''. The examples of articles rejected in Phase 2 are given in Table 4. At the end of this phase, 71 articles were retained.

TABLE 4. Examples of Articles Rejected in Phase 2.

Article	Reason of rejection
Bodnarchuk I., Duda O., Kharchenko A., Kunanets N., Matsiuk O., Pasichnyk V., "Multicriteria Choice of Software Architecture Using Dynamic Correction of Quality Attributes", Advances in Intelligent Systems and Computing, vol. 938, pp. 419-427, 2019	The article is not about requirements engineering (concerns architecture selection)
Jia J., Yang X., Zhang R., Liu X., "Understanding software developers' cognition in agile requirements engineering", Science of Computer Programming, vol. 178, pp. 1-19, 2019	The article is not about NFRs (concerns cognitive perception of requirements, the abstract: "achieving good quality requirements" refers to well developed requirements, not NFRs)
Mahato S., Rai Dixit A., Agrawal R., "Application of Lean Six Sigma for cost-optimised solution of a field quality problem: A case study", Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, vol. 231, no. 4, pp. 713-729, 2017	The article is not about software engineering (concerns optimization in an organization manufacturing consumer goods)
Agra C., Assis D., Sousa A., Jaqueira A., Lucena M., Maciel T., Alencar F., "Transforming i* model in user stories: An approach to agile documentation based on rationales, intentions and NFR", CIBSE 2016 - XIX Ibero-American Conference on Software Engineering, pp.304-317, 2016	The article is not in English (it has a double title in English and Portuguese, the abstract and full text are in Portuguese)

4) PHASE 3

In this phase, the articles were checked against exclusion criteria. All articles (except unavailable ones – E2) were read independently by both authors. Their decisions were later discussed and subject to consensus. Finally, 31 articles were qualified to extract information. During the review, apart from just deciding on a article's final classification, each author identified the sections relevant to the RQs and extracted the corresponding data, in particular: practices aimed at initial NFR identification (and their assignment to the phases of project life cycle), NFR-related requirements elicitation practices and techniques, and NFR-related requirements documentation techniques. The individual findings were later compared and merged into a final set, resolving the discrepancies in the consensus-based mode. For the majority of sources, our findings were very similar. It can probably be attributed to the fact that the search concerned particular practices and techniques which can be identified in a relatively easy manner. Differences in our individual findings were identified for three articles. Further investigation revealed that, in each of these cases, one author overlooked a practice mentioned in the article's text, while the other was more perceptive. The resolution of such differences was straightforward, as it only required finding the proper part of the article and verifying that it refers to a practice relevant to one of the RQs posed.

The extracted information pieces were grouped into higher-level themes. For example, the following quotes: ''*Once the micro-business owners have enumerated their business goals and diagrammed their business processes, they may proceed with a requirements definition meeting (. . .) with the software developer.*'' [59] and ''*The MEDoV provides use of business models and process approach, and participants in RE are elected to the role in the process. Process owners are responsible for accuracy and completeness of elicited business data.*'' [60] resulted in the ''On the basis of business process models'' requirements elicitation practice included among the findings provided in Section IV-A.

We also conducted a quality assessment of the 31 qualified articles and documented its results. We did not use the quality assessment as a basis to reject any articles though, as the purpose of the SLR was rather exploratory - to identify inputs to be used in the interviews.

For the purpose of quality assessment, we adopted the criteria elaborated by Curcio *et al*. [31]. Their criteria are based on the checklist developed by Dybå and Dingsøyr [61] and facilitate the collection and analysis of information which is valid from the point of view of the defined research questions. However, we introduced one modification. The approach by Curcio *et al*. assumes that articles of types other than ''research paper'' (*e.g.* experience reports or expert opinions) have to automatically receive a negative assessment and be excluded from further processing. We believe that in the case of our SLR study, which aimed to identify the practices and techniques used by ASD practitioners, there was no reason to exclude, *e.g.* practices mentioned by industrial experience reports. For this reason, we substituted this assessment criterion with a simple classification of the article's type. The other criteria by Curcio *et al*. were adopted as the following list:

QC1. The aims and objectives were clearly reported.
QC2. There was an adequate description of the context in which the study was carried out.
QC3. The design was appropriate to address the aims and objectives.
QC4. The study provided clearly stated findings with credible results and justified conclusions.
QC5. The study provided an explicit discussion of its value for research or practice.

Each qualified article was reviewed, classified and assessed in accordance with the QC1–QC5 criteria. For each criterion, the following rating was used: 0 – ''no''; 0.5 – ''partially''; 1 – ''yes''. The results of the quality assessment are included in Appendix C, which presents the list of the qualified articles.

B. INTERVIEWS

We planned a multi-case study that adopted a semi-structured interview strategy, following the guidelines by

Wohlin *et al*. [62]. In order to collect empirical data, we used purposive sampling techniques which are used in qualitative research, and may be defined as selecting interviewees based on specific goals that are associated with answering research question [63]. In particular, considering the fact our research gathers specific knowledge from individuals that have particular expertise, the adopted sampling technique falls into the assumptions of expert sampling.

First of all, expert sampling requires individuals with a high level of knowledge and skills relating to the subject of study. Secondly, their level of expertise must meet requirements regarding their education, position, years of professional experience, and the type of projects they have participated in. In this extent, Table 5 presents the assumptions taken equally for granted.

TABLE 5. Interviewees' Inclusion Criteria.

Criterion	Value
Education	Bachelor, Master or PhD degree
Position	any from the following: project manager, product owner, scrum master, system analyst, business analyst, developer
Experience	at least three years
Projects	participation in at least three IT projects compliant with agile methodologies

In our opinion, adopting all of these values should guarantee a reasonable level of the expert's expertise. Having prior knowledge, we targeted the sample from our social network members. This strategy paid off, and in a relatively short period of time, we were able to perform the in-depth interviews with ten individuals who met the inclusion criteria.

We prepared an interview guide beforehand, which included the main topics and questions addressing our RQs. The interview guide is included in Appendix A. The interviews were however conducted in an open-ended manner, the order and exact form of questions could be modified by the interviewer as long as all the issues included in the guide were covered. It also allowed the interviewee to suggest and follow up additional topics he/she found relevant. At the beginning, each interviewee was informed about the purpose and manner of the interview, guarantee of confidentiality, as well as of the estimated time necessary to conduct it (about 90 minutes). The interviews were documented by writing down the responses or by recording the conversation in digital format using a mobile device, since not all interviewees agreed to be recorded.

The study was launched on January 2020 and finished in April 2020. Each of the ten interviews lasted from 60 to 120 minutes. While only some interviewees agreed to be recorded, all answers were written down using a predefined template. In practice, this let us structure the obtained data and facilitated the analysis process. Ultimately, the notes from each interview were between 3 and 7 pages long.

After all interviews were conducted, both authors independently analyzed the open text responses resulting from the interviews. Following the qualitative data analysis guidelines developed by Charmaz [64], each author conducted the initial coding. Later, the initial codes were compared and discussed together by both authors to develop higher-level categories and assign final codes that led to the findings presented in Section IV-B. The example of initial coding applied to the interview text is shown in Appendix B.

IV. RESULTS

A. SYSTEMATIC LITERATURE REVIEW

The SLR resulted in reviewing the finally qualified 31 articles and extracting from them the pieces of information relevant to our RQs. Our findings with respect to each RQ are presented separately. It is worth noticing that about half of the articles (14) were published in or after 2017, which indicates the growing interest in this topic. The list of the qualified articles is provided in Appendix C.

RQ1. When are non-functional requirements identified in the project life cycle? The findings relevant to RQ1 are summarized in Table 6. The most common practice is the identification of NFRs in each iteration of agile development. Some other solutions are proposed though - recognition of NFR importance results in efforts to capture them early in the software development process. Some of such proposals still follow the iterative approach typical for agile (high-level NFRs identified during initial iteration but then refined in further iterations), but others seem to resemble the Waterfall approach (identification of NFRs at the beginning of the project). Also, cases where NFR identification starts in the late phase of the project are mentioned, but usually together with information that it caused problems and additional rework later.

TABLE 6. SLR Findings for RQ1.

Timing of NFR identification	Sources
In "sprint zero", resulting in definition of the overall software architecture	[65]
During initial stages of development process	[60], [66]
During initial iteration of the project (high level NFRs), followed in further iterations (NFRs refined to more detailed form)	[67]–[69]
During each iteration	[16], [22], [24], [70], [71]
After FRs are defined	[22]
In late stages of the project (causing rework)	[6], [24], [25]

RQ2. How are non-functional requirements elicited? Surprisingly, not many particular elicitation techniques are explicitly mentioned in the reviewed sources (see Table 7). Among such techniques, interviews, meetings and training sessions with stakeholders as well as group-work (workshops, brainstorming) seem to be most popular. Document circulation (that seems not to be in line with direct communication practices) is also proposed. Also, some resources like pattern catalogs, business process models, standards and other sources on NFR classification are reported as tools used in requirements elicitation.

TABLE 7. SLR Findings for RQ2.

Elicitation practice	Sources
Customer-developer meeting	[25], [59]
Interviews with users/customer representatives	[72]
Interviews with technical experts	[72]
Dedicated NFR elicitation workshops	[17]
Mini-QAW (Quality Attribute Workshop)	[73]
Requirements elicitation workshops	[74]
Brainstorming	[24], [73]
Prototyping	[24]
Moderated circulation of NFRs document between Product Owner and NFR stakeholders (*e.g.* experts)	[15], [75]
Instantiation of patterns from NFR catalog	[16] [76]
Using standards or other NFR classification sources as a reference	[72]
On the basis of a training provided to the customer in each iteration	[68]
On the basis of business process models	[59], [60]
By automated OCR and text processing of project documents and images	[77]

RQ3. How are non-functional requirements documented? A significant number of requirements documentation techniques used to specify NFRs was found by the SLR. The popular techniques, commonly used in Scrum, eXtreme Programming and other methods (user stories, features, acceptance criteria, definition of done) are utilized not to represent FRs only, but NFRs as well. There is however a number of techniques proposed with explicit intent to better represent NFRs *e.g.* technical stories, Agile Loose Cases, Semi-Structured User Stories, structured story cards and extended acceptance criteria (AC+). Some simplified solutions like wiki pages or whiteboard sketches are also mentioned. NFRs can also be defined in relation to particular FRs using *e.g.* constraints or traceability matrices. The full summary of SLR findings related to RQ3 can be found in Table 8.

TABLE 8. SLR Findings for RQ3.

Documentation technique	Sources
User stories	[25], [57], [66], [74], [78]
User stories with acceptance tests	[79], [80]
Constraints within user stories	[25], [65]
Semi-structured user stories (SUS)	[76]
Technical stories	[78]
Epics	[57], [74]
Features	[81]
Acceptance criteria	[22], [57], [66], [69], [74], [78], [82]
Extended Acceptance Criteria (AC+)	[71]
Tasks	[66]
Definition of Done	[22], [57], [69], [78]
Mock-ups/prototypes	[57], [71]
Whiteboard sketches	[57]
Structured story cards (with attributes expressing NFRs)	[19], [20], [70], [72]
Textual quantified descriptions	[74]
Traceability matrices (quality attributes vs. FRs)	[76], [83]
Agile Loose Cases	[75]
Wiki pages	[57]
Instances of requirements templates dedicated to particular categories of NFRs	[16], [17]
NFR Framework (Goal Oriented Requirements Engineering notation) models	[20], [75]

TABLE 9. Characteristics of the Interviewees.

Item	Values	Number
Gender	Male	7
	Female	3
Experience in years	Less than 3	0
	Between 4 and 6	1
	7 and more	9
Role	analyst	4
	developer	2
	project manager	3
	scrum master	1
Sector	automotive	1
	banking	2
	hardware	2
	maritime	1
	software	3
	public services	1

B. INTERVIEWS

A summary of the interviewees' characteristics is given in Table 9. With a slight majority of males over females, all interviewees except one declared having at least 7 years or more in ASD projects. We were able to collect data from professionals holding various job positions and working in diverse sectors.

In the following subsections, we first provide the contextual information we learned from the interviewees regarding their projects and the significance NFRs have in such projects, then we report the main findings with respect to each of the RQs posed.

1) CONTEXTUAL INFORMATION

Software development processes. An important observation stemming from such a consideration is that, despite using the agile approach to software development, this approach is hardly the only method followed by the team, department or organization as a whole. Some interviewees explicitly stated that they use a hybrid approach (integrating agile and plan-driven practices), others claimed that they use a specific agile method (most often Scrum), but at the same time, at the management level, some other method is applied (*e.g.* PRINCE2, ITIL).

Perception NFR importance. No single interviewee neglected the importance of NFRs, nor claimed that they are not identified in their projects. Naturally, depending on the business sector, different categories of NFRs were considered as crucial or important (4 or 5 on a 1–5 scale). For example, usability was crucial for public and commercial services (accessed by end customers), while in the case of back-office banking systems, it had a low or medium priority. Reliability and fault tolerance was essential to banking and automotive, while less important to other sectors. Similar differences regarding the perceived importance applied to

performance, availability, portability and maintainability. The only exception was security, which was considered crucial by virtually all interviewees (however to some extent, this was influenced by the recent adoption of the General Data Protection Regulation by EU countries).

2) RQ1: WHEN ARE NFRs IDENTIFIED IN PROJECT LIFE CYCLE?

Early attempts to identify NFRs. Only one out of ten interviewees claimed that NFRs are identified in an iterative, ongoing manner, while meeting with stakeholders and presenting them working versions of the software. All others insisted on early identification of NFRs, some even literally expressing that ''the sooner, the better''. Three main approaches could be distinguished here. The first one is to roughly identify NFRs at the beginning of the project and then to capture more details in later iterations. The second one, treats NFRs in a Waterfall-like manner (though FRs are still established iteratively) - an attempt is made to explore NFRs as much as possible at the beginning of the project. The third approach deals with NFRs even earlier - before the project is even established. NFRs are identified, their feasibility is assessed and if the risk of failing to meet an NFR is too high, the contract will not be signed.

Managing NFR expectations. Many interviewees mentioned a potential problem regarding customer representatives who ''overspecify'' NFRs in a way that challenges their technical and economic feasibility. The usual solution is to address this issue as soon as possible and to negotiate with the customer representatives and provide them information about the consequences of their demands. The interviewees noticed that face-to-face meetings and direct communication (common in ASD) facilitate reaching consensus in such matters. However, two interviewees revealed that the strategy of their organizations to cope with overspecification is to identify NFRs early, possibly before the project even starts, but at the same time discourage customer representatives from expressing them explicitly *e.g.* by avoiding this topic during the meetings. This results in a situation where, on the one hand, NFRs are identified very early by the supplier organization, but are not explicitly recognized as ''official'' requirements, or such recognition is delayed until later stages of the project. As this strategy seemed to contradict the agile principle of close collaboration with customers, we specifically inquired the interviewees about it. Both interviewees confirmed the use of such strategy in their organizations, explained that it originates from a risk management approach taken and claimed that the strategy is effective.

3) RQ2: HOW ARE NFRs ELICITED?

Analysts in agile teams. Despite the fact that most agile methods do not explicitly distinguish such a role, a frequent practice is to include an analyst in the agile team. Six of our interviewees mentioned business analysts or system analysts as the team members mainly responsible for requirements elicitation. In some cases, the analyst was also assigned the Product Owner (PO) role, in others, such roles were separated and the analyst's main task was to support the PO and, *e.g.* elicit additional requirements from other stakeholders. The involvement of qualified analysts facilitates the elicitation of NFRs, as their skills and experience make them aware of the NFR categories essential to the IT products developed for a given business sector. They are able to take a more proactive approach and query stakeholders about such NFRs instead of simply recording stakeholders' needs. The interviewees considered the presence of an analyst as a factor contributing to more thorough elicitation of NFRs.

Sources of requirements. Who or what is considered a source of requirements and how many of them are relevant in a given project? Such issues are strongly dependent on the project context (in-house or contract development, a standalone or an integrated system etc.). Some interviewees claimed that a single person (a business domain expert, often acting as a Product Owner) was able for providing the majority of the requirements. In other cases, multiple stakeholders and their viewpoints had to be considered and requirements elicited from each of them. This often included non-human stakeholders (other IT systems, standards).

What is interesting, is the fact that developers are also considered a source of requirements, as they discuss generic requirements, refine them and suggest some features for business stakeholders' consideration and acceptance. This seems to be consistent with agile values and practices promoting the attitude of responsibility for the product and intensive communication between stakeholders and developers. As for NFRs, the interviewees noted that their effective elicitation often required the involvement of additional stakeholders, namely technical experts, who provide input within their area of expertise (*e.g.* security). Also, the input from developers is considered important in the case of NFRs, as their questions or/and proposals lead to refinement of the NFRs expressed by business stakeholders.

Effort to elicit. As mentioned above, all of our interviewees recognized the significance of NFRs due to their experience, and recalled projects where NFRs were important to the stakeholders. Unfortunately, it does not necessarily mean that the stakeholders are always aware of their needs regarding NFRs. Almost all interviewees mentioned challenges with eliciting NFRs and the significant effort required to do this. Typical problems reported included: a lack of stakeholders' initial awareness about NFRs and a focus on FRs only; very generic (non-verifiable, non-measurable) NFRs; and partial coverage of NFRs (some categories, *e.g.* usability, are recognized, but others not). To summarize - the interviewees claimed that elicitation practices in their projects were effective, but an active approach of the development team and guidance provided to the stakeholders were necessary.

Requirements elicitation techniques. The techniques most commonly used are interviews and workshops (including brainstorming sessions and other kinds of group work) - virtually all interviewees mentioned them. This finding is in line with the agile approach relying on direct communication.

Other techniques, less frequently used, but still mentioned by multiple interviewees are: prototyping, document analysis, analysis of existing systems, and observations. There were no reports on elicitation techniques dedicated only to NFRs, apart from one mention of usability tests. All techniques listed are used to elicit all kinds of requirements, not just NFRs.

4) RQ3: HOW ARE NFRs DOCUMENTED?

Documentation techniques and their level of detail. The Agile Manifesto endorses development of the working software over comprehensive documentation, but on the other hand, poorly documented NFR can be misinterpreted and cause rework. The interviewees seem to apply different strategies to cope with this issue. They mentioned numerous specific requirements documentation techniques, but in general three approaches can be distinguished. The first, ''most agile'' one applies lightweight requirements documentation techniques (epics, features, user stories, etc.) which are also used to represent generic, simplified NFRs. Such a NFR is treated as a ''to do'' list item and, when the development team starts working on a given item, it is explained through a face-to-face discussion. The second approach uses detailed NFR representations, which usually exist in parallel with typical agile artefacts like a Product Backlog and its contents (features or user stories). NFRs, unlike FRs, are specified in a precise, measurable form as, *e.g.* structured descriptions and included, *e.g.* in the Software Requirements Specification or Master Test Plan. Developers mainly use a Product Backlog in their everyday work, but when necessary, they refer to such an external document. The third, ''hybrid'', approach makes use of typical agile artefacts and documentation techniques, but ensures that NFRs are specified on a sufficiently detailed level. For example, a Product Backlog consisting of features is used, but in the case of NFRs, the corresponding feature is not a short, informal statement, but a longer expression, including hard numbers and unambiguous acceptance criteria.

Maintaining documented NFRs. In general, the interviewees do not consider documenting requirements (and later maintaining them) as problematic tasks. Their teams adopted suitable tools that support collaborative work in this area, thus requirements can be disseminated to all interested parties as well as updated when a change occurs. Only one interviewee complained that the simplified documentation techniques used to specify NFRs in their project turned out to be insufficient and caused problems. All others, regardless of which approach (among the ones described in the previous finding) they followed, claim that the documentation techniques they use are sufficient for their projects and no significant problems are observed.

V. DISCUSSION

Our study has several implications for both researchers and practitioners, which are outlined in subsections V-A and V-B, respectively. Also, several possible limitations to this study are discussed in the end.

A. THEORETICAL IMPLICATIONS

This study combined two qualitative research methods (SLR and interviews) to investigate the issue of NFRs in ASD projects, and provide reliable answers to the RQs. We argue that the existing literature on the subject has documented the knowledge similar to the know-how explicitly obtained from the interviewees. We also provide new insights into when NFRs are identified in the project life cycle, and how they are elicited and documented.

In the case of the first issue, while some recognize this need almost immediately, others choose to wait until the project is advanced. The remaining ones argue for dealing with NFRs in an incremental way. Therefore, there is no consensus in this area.

In case of NFRs elicitation, few techniques were explicitly reported by the authors from the conducted studies. Therefore, the existing literature in this area still remains limited. However, this article enriches the research of examining agile practitioners' cognition in agile NFR elicitation techniques. In fact, the obtained findings revealed a gap which might be further investigated since over last years considerable effort has been undertaken to efficiently and accurately understand stakeholders needs (*e.g.* collaborative games).

In the case of NFR documentation practices, we have identified twenty different techniques. While some are embedded in the agile methods by design, others seem to be introduced to ARE processes with the sole purpose of documenting NFRs. Furthermore, we acknowledge that there are some remarkable differences between typical and specific requirements which in turn determine the use of particular technique(s).

The results from the SLR and the interviews are mostly consistent with each other. Certainly, not all practices/techniques found in the literature were declared by the interviewees as the tools they apply to NFR engineering, but at a higher level of abstraction, the more general strategies or approaches to elicit and document NFRs were confirmed. The main differences between the existing research described in the literature and the interview findings are: *(i)* the fact that the interviewees almost unanimously declared that they make attempts to identify NFRs in the early phases of software projects; *(ii)* the interviewees' feedback that despite their perception of NFR engineering as a non-trivial task requiring significant effort, they do not consider NFRs as a problematic issue that threatens the outcome of the project and they declare that the NFR-related practices and techniques they currently use are sufficiently effective; and *(iii)* despite several NFR-dedicated requirements elicitation techniques proposed in the literature, our interviewees reported that in their organizations, the same elicitation techniques (mostly based on face-to-face communication) are used to elicit both FRs and NFRs.

The findings of our study provide a contribution to the existing knowledge. As reported in sections II-C (Related work) and II-D (Summary of previous studies' findings), the number of studies directly related to our work is rather

limited. With respect to the more general studies focused on identifying NFR-related problems, we could not confirm the significance of the frequently quoted problem ''Neglecting non-functional requirements'' [7], [8], [11], [28], [29] as our interviewees expressed vital interest in identifying NFRs and reported that they focus on NFRs early in their projects. As for the problems related to requirements documentation techniques (''Too-minimal requirements documentation to capture NFRs'' [8], [30] and ''Insufficiency of the requirement documentation techniques'' [7], [11]), our findings indicate that they are encountered, but effective solutions are used by practitioners (*e.g.* if an NFR is difficult to express as a user story in the Product Backlog, it is specified in a separate document using a structured description). Such solutions can however raise questions about how they fit into existing agile methods and practices.

As for NFR-related practices and techniques that can be found in the literature, the following proposals were confirmed by our study: early identification of NFRs [28], [30], using additional requirements documentation techniques [7], [11], [29], adjusting the user story format to express NFRs in a more precise manner [7]. In most cases, it is not possible to directly compare our findings to results reported by others, as they do not explicitly focus on NFRs and thus it is not possible to tell which reported practices/techniques were applicable to NFRs (*e.g.* [27], [28], [31]). Also, some sources mention practices related to software project organization like division into teams or a customer-supplier cooperation model (*e.g.* [8], [28]), which also makes comparison difficult.

As our study was designed with the intent to *i)* focus specifically on NFRs; *ii)* identify particular elicitation and documentation practices/techniques, our findings are much more detailed in comparison to the previously available body of knowledge and provide a contribution to the ARE research area.

The review of the existing body of knowledge found during the SLR study and the comparison of the findings obtained from the SLR and from the interviews led us to some additional considerations. We were able to identify a number of theoretical implications for software engineering research and related open questions.

We were quite surprised that the interviews did not confirm the occurrence of the problem of neglecting NFRs, which was reported by several SLR sources. Such a difference can possibly be attributed to the observation that our interviewees represented more mature organizations. The interviewees were selected and invited through our social networks, which mostly included professionals interested in requirements engineering (*e.g.* attendees of postgraduate courses on this topic). It is thus possible that a higher level of awareness and maturity can be attributed to our sample (practitioners and their organizations) than the general population. This however prompts an open question about the factors leading to the described problem. We believe that research studies dedicated to the root causes of the ''Neglecting NFRs'' problem can be recommended.

Several alternative approaches to the early identification of NFRs were found using both research methods, SLR and interviews. In particular, some proposals advocated the attempt to establish NFRs in the initial phase/iteration, while others relied on rough identification of NFRs at the beginning and their later refinement in the subsequent iterations. In our opinion, this implies some open questions and research directions. The first question is: to what extent can NFRs be captured early in an agile project? The answer is probably not the same for all projects but dependent on some contextual factors, thus the next research direction would be to identify such factors. Another issue worth exploring is a more detailed investigation of the influence the approach advocating early NFRs identification has on agile processes and practices, as it seems not to be entirely consistent with ''standard'' ASD processes that assume continuous requirements identification or update in each iteration.

The observation that the SLR identified several requirements elicitation techniques dedicated solely to NFRs (*e.g.* NFR elicitation workshops or support of NFR pattern catalogs), while the interviewees denied that their organizations used such techniques, has another possible implication. It provokes an open question about the effectiveness of such dedicated techniques and the added value they bring despite the effort and cost associated with their usage. We identify a need for the empirical studies to investigate it. In the case that such studies demonstrated the effectiveness and efficiency of particular dedicated NFR identification techniques, it would provide arguments for the industry to adopt the considered techniques.

Another interesting research direction is dedicated to NFR documentation techniques. Both of our research steps (SLR and interviews) reported that several alternative strategies to documenting NFRs are used – some practitioners are able to make use of lightweight documentation techniques like user stories, while others apply alternative documentation techniques or even create separate documents/registers for NFRs only. The question arises, whether such a difference arises from the inability to use the agile techniques in an effective manner or from the fact that such techniques are insufficient (at least in the case of some projects and their requirements). The NFR-related problems reported in the literature (''Too-minimal requirements documentation to capture NFRs'' and ''Insufficiency of the requirement documentation techniques'') suggest the latter option, but still further investigation would be beneficial. In particular, it would be worthwhile to identify what factors (*e.g.* category of NFRs, conformance to norms/standards) determine whether lightweight requirements documentation techniques are sufficient or not for a particular IT project/product.

Last but not least, in our opinion, the unexpected finding of the revealed strategy of the software supplied to identify the NFRs but ''hide'' them by avoiding explicit communication with the customer representatives, also requires attention. We have not encountered a similar solution in the literature, thus it should be verified if such behavior is exceptional and

not encountered in the general population or that it is not revealed (as something potentially inconvenient to share). Also, as such an approach seems to clearly contradict the agile principle of close collaboration with customers, a more thorough investigation about its consequences on the whole IT project and customer-supplier cooperation would be advised as a possible direction of future research.

B. PRACTICAL IMPLICATIONS

We found that practitioners recognize the significance of NFRs, which confirms the results of earlier studies [21], [22]. The practitioners undertake effort to identify NFRs as soon as possible, even if such an approach can be challenged as deviating from core agile practices. The reported reason is the potential risk of omitting an NFR that can impact the architecture of the developed system and the outcome of the whole software project. Some practitioners work in a more iterative and incremental manner by limiting the initial NFR activities to identifying them only and postponing the exploration and refinement until later iterations. Others however try to capture detailed NFRs at the beginning of the project or even before. Nevertheless, this seems to prevent the frequently mentioned (*e.g.* [6]–[8]) challenge of key NFRs omission (in the literature *e.g.*, as our interviews except one did not reveal such a problem. This may be a clue to practitioners that several strategies are possible and each of them can work).

Despite their significance, NFRs are not easily elicited from stakeholders, who may not be aware of all of their needs or may have difficulty articulating them. This is not surprising, as such challenges are commonly reported by RE practitioners [37], [84], regardless of the approach used (agile, plan-driven, hybrid). What we learned however is that our respondents make an effort to learn all stakeholders' needs, not just those explicitly expressed. Direct communication (encouraged in agile approach) and related requirements elicitation techniques like interviews and workshops are the right tools for this purpose, since no other elicitation techniques specifically aimed at NFRs are used. It seems to confirm observations about the effectiveness of ARE elicitation techniques, based on face-to-face communication [6], [8].

The focus on FRs and simplified requirements documentation are often attributed to agile methods and listed as potential challenges [7], [9], [11], [21], leading to unclear and unmeasurable NFRs. What we found out, however, is that our interviewees claim that they cope with such challenges, even though they use different solutions. For some of them, simplified documentation techniques like features or user stories are applicable to NFRs, as the main way of explaining the requirements to developers is through direct communication within the project team. Others try to adopt simplified techniques, mainly by extending them to ensure more measurable NFR descriptions. Finally, some practitioners use detailed documentation in parallel with typical agile artefacts. This finding can also provide practitioners with ideas on how to handle requirements documentation in their projects, though an additional study comparing the effectiveness of such practices would be desirable.

C. LIMITATIONS

The limitations of the systematic literature review and interviews are considered separately.

1) SLR

A limitation of our SLR study is the fact that only one publication database was searched (Elsevier Scopus). Scopus was selected because it indexes a large number of journals and conferences [85] and provides a single search query access to items from a broad variety of publishers [86]. Several SLR studies that we identified as related work, *e.g.* [10], [11] followed a similar strategy and relied on Scopus only.

A limitation of any systematic literature review is the possible selection bias caused by a non-optimal search string. We made an effort to define the search string in an iterative manner that also included running searches and assessing the results. The search string included several synonyms and alternative terms used by software engineers. We cannot however completely exclude the possibility that some authors of the relevant articles could have used other, less common terms and as a result, such articles were not found.

Another possible limitation concerns invalid choices regarding a article's inclusion/exclusion. We minimized this threat by following the established guidelines [58] and by ensuring that each decision made in Phase 3 (review of full-texts) was a consensus made by both authors of this article. Similarly, to avoid risks related to data extraction, this task was conducted independently by both authors, who later compared their findings.

2) INTERVIEWS

The first possible limitation is the selection of interviewees. By design, nonprobability purposive sampling, and in particular expert sampling is a form of non-random technique that does not rely on a particular theory indicating the number of respondents. Therefore, the subjective and nonprobability nature of their selection impose the limitation to perform inductive generalization of the obtained results. However, considering the variety of inclusion criteria to be simultaneously met by the interviewees, covering areas such as education, position, and years of professional experience, this technique is claimed to deliver relevant information if the questions precisely correspond to the interviewees' expertise [87].

The second limitation found during this research concerns the size of the sample used. The total number of interviewees is 10, and therefore its representativeness is limited. Their responses thus cannot be synthesized in order to formulate definitive, general conclusions. Additional research is needed to compare our findings with those resulting from larger or different samples.

In the case of many interviewers present, they are claimed to have the opportunity to discuss and verify their

understanding [88]. In our case however, each interview was a one-on-one meeting. We argue that involving two interviewers per meeting would be double the workload, while each of us had sufficient knowledge and experience to act as interviewer.

The threat of a possible lack of an interviewee's honesty (caused, *e.g.* by the intent to make their company look better) was minimized by interviewing volunteers only, assuring the anonymity and confidence of all information provided, encouraging them to provide accurate examples, and allowing the refusal to answer any question without giving a reason. Despite all the attempts to mitigate this threat, we cannot entirely eliminate it, especially in cases that involved more subjective matters, *e.g.* the perception of the effectiveness of practices/techniques applied. It is a common limitation of the research method applied (interviews) and to overcome it, another method would have to be introduced. For example, the effectiveness of requirements engineering techniques can be evaluated on the basis of the data stored in software tools supporting development activities [43], but in the reported research study, we had no access to such data.

Moreover, the threat of bias introduced by a single point of view was minimized as both authors conducted independent reviews and interpretations of the interviews, which were later compared to reach a consensus.

External validity is difficult to achieve in the case of interview-based studies as interviewees provide information regarding particular settings (in this case – their organizations and projects) and their own opinions/viewpoints. Such findings cannot simply be generalized for any organization and any project. To some extent, we minimized such limitation by inviting practitioners from various industry sectors and representing different project roles/ job positions. On the other hand, all interviewees were from one country (Poland), which can be considered a limitation with regard to external validity.

VI. CONCLUSION

In our study, we intended to recognize the importance of NFRs in ASD projects, as well as to identify the up-to-date practices and techniques used in this area. To explore these topics, we conducted a systematic literature review which represents a rigorous and transparent approach to synthesizing the existing body of knowledge that eliminates human bias. Finally, to explore and strengthen the obtained findings, we conducted an additional qualitative study consisting of ten in-depth interviews with industry experts.

Our findings concerning the timing when NFRs are identified in the project life cycle indicate that there is no consensus on that matter among the practitioners. Only one interviewee (and a minority of literature sources identified in the SLR study) claim that NFRs are identified in a strictly continuous manner, in each iteration and in conformance to the iterative agile development processes. The majority of interviewees reported that their organizations opt for early identification of NFRs, but still their strategies were significantly different, *e.g.* attempts to capture all NFRs at the beginning of the project (or even before) vs. rough identification of NFRs at the beginning and continuous refinement in the following iterations. The only consensus we observed concerned the perceived importance of NFRs and the attitude to comprehensively identify them, in order to mitigate significant risks to the software project.

In order to elicit NFRs, in total, 14 techniques applicable to ASD have been reported by SLR sources. Most of these techniques seem to strongly rely on close collaboration with stakeholders. By design, agile expands the role of the user/stakeholder in software development by involving them on a regular basis. Such collaboration has been recognized as a key element in most agile NFR-related processes. Despite the fact that several techniques dedicated to the elicitation of NFRs were found in the SLR study, the findings from the interviews indicate that the interviewees' organizations do not use such techniques and rely on universal elicitation techniques applicable both to FRs and NFRs.

NFRs are commonly documented in natural language. In fact, the most frequent practice is to agree on the acceptance criteria regarding the particular facets of the software product. Secondly, *Definition of Done* is used to provide a shared understanding between the development teams and stakeholders by utilizing checklists. It is worth noting here that enriched user stories are used as a form of NFR documentation, containing related acceptance criteria and/or *Definition of Done*. Our findings indicate that the documentation of NFRs is often conducted in a more detailed manner than FRs. In some cases, it is enough to use documentation techniques common to FRs, but make extensive use of acceptance criteria and *Definition of Done*, while in others, NFRs were documented in different ways than FRs or even stored in separate documents/registers.

Our future research will cover an evaluation of the effectiveness of the identified NFR elicitation and documentation techniques and practices. Eventually, the outcome will let us assemble a generic framework, intended to facilitate NFR management. Secondly, we will investigate the factors related to NFR practices stemming from the adoption of agile methods across software development organizations.

APPENDIX A
INTERVIEW GUIDE

Introduction. Introductory information provided to the interviewee.

- Ask for the consent to record the interview. In case of refusal, proceed without recording.
- Inform the interviewee about the purpose and objectives of this research study and how interviews with practitioners contribute to it.
- Assure the interviewee about confidentiality issues (the information revealed will only be used for the purpose of this research study, the interviewee will remain anonymous, the published results will include clustered

information that prevents identification of individuals and organizations).
- Inform that the interviewee can refuse to answer any of the questions asked (without providing any justification of such refusal) and can conclude the interview at any time he/she chooses.

Contextual information. Establishing the context for further questions by learning about the interviewee, his/her organization and the organization's software development processes.
- Interviewee's professional profile – job position; experience in the current job position; total experience in the IT industry; experience in using agile methods.
- Organization the interviewee currently works for – size; sector; types of IT projects (single customer, multiple customers, in-house development, outsourcing etc.); organizational structure (and how project teams fit into it).
- Methodologies - software development methods used in the organization; adoption/tailoring of known methods to the organization's specific context; use of managerial methodologies; size and structure of project teams.

NFR categories. Questions to determine which NFR categories apply to software projects developed in the interviewee's current organization and how important they are considered to be.
- Which categories of NFRs are explicitly considered for the products developed in the organization's software projects? Note: first allow the interviewee to answer freely, later ask about the following categories: performance, usability, security, availability, reliability/fault tolerance, flexibility/maintainability, portability.
- What is the relative importance of the NFR categories discussed in the previous question? Note: ask the interviewee to evaluate them on a 1–5 scale (1 – lowest, 5 – highest).

NFR identification. Questions about initial NFR identification in software projects developed in the interviewee's current organization.
- When are NFRs first identified in the software project life cycle (*e.g.* at the beginning of the project; in a dedicated initial iteration/phase; in a continuous manner during each iteration; in later phases of the project)?
- Does the project team undertake actions to identify NFRs important in the context of a given project or do they consider it a responsibility of customer representatives?
- Is the approach to NFR identification used by the organization considered to be effective or does it result in problems encountered by project teams?

NFR elicitation. Questions about elicitation practices in software projects developed in the interviewee's current organization.
- Which project team member(s) are responsible for requirements elicitation?

TABLE 10. An Example of Coding.

Interview text	Initial codes
Researcher: OK, so in the case of some projects, the NFRs are included in the Call for Proposals or a similar document. But what about the other projects?	
Interviewee: We try to identify such requirements during the initial meetings. If we believe there may be a problem, we start negotiating.	
Researcher: And when does it take place? During initial phase of the project?	
Interviewee: Earlier, before the contract is even signed. We intend to identify potential risks before we make a commitment and start a project. Frankly speaking, we don't want to deal with lunatics who issue unreasonable requirements like the system shall respond in a split second and such (. . .).	NFRs before project NFRs as a risk factor "Overspecification"
(. . .)	
Researcher: So you try to figure out the customer's attitude and the relevant main NFR categories before the project is started?	
Interviewee: We will not start a project, we will not sign the contract if there are unmitigated risks. We discuss such risks with customer representatives and try to reach a consensus. However, even if such a discussion takes place, we sometimes decide not to include the details of overspecified NFRs in the system requirements specification document.	Negotiation of NFRs Avoidance of explicit NFRs specification
(. . .)	
Researcher: Who is responsible for such initial NFR identification before the start of the project?	
Interviewee: Management staff, which means our CEO and me (as a lead analyst). We also involve an analyst who helps us to specify, document such requirements and can later take over requirements engineering after the project starts. As for the CEO and me, we consider such activities to be part of our project risk management duties rather than as requirements elicitation.	Analyst in agile team Project management methodologies

- Which project team members participate in requirements elicitation activities?
- What does the collaboration with customer representatives and other stakeholders look like (one representative or more, the availability of representatives/stakeholders, involvement of non-human stakeholders *e.g.* documents, business models or existing IT systems; involvement of technical experts)?
- Which requirements elicitation techniques are used? Note: first allow the interviewee to answer freely, later ask about the techniques uncovered by the SLR study.
- Are there any dedicated techniques used for elicitation of NFRs (other than in the case of FRs)?
- Which techniques are most effective in NFR elicitation based on the interviewee's experience?
- Are customer representatives/stakeholders aware of NFRs and express such requirements or do they need to be specifically asked about them?
- Are there any activities preceding NFR elicitation necessary *e.g.* providing training courses to stakeholders, explaining the meaning of particular NFR categories?

TABLE 11. List of Qualified Articles With Quality Assessment Results.

ID	Ref.	Authors	Title	Venue	Year	QC1	QC2	QC3	QC4	QC5
S1	[83]	B. M. Aljallabi and A. Mansour	Enhancement approach for nonfunctional requirements analysis in agile environment	conference	2015	0.5	0	0.5	0.5	0.5
S2	[65]	W. Alsaquaf	Engineering quality requirements in large scale distributed agile environment	conference	2016	1	0.5	1	0	0
S3	[25]	W. Alsaqaf, M. Daneva, and R. Wieringa	Agile quality requirements engineering challenges: First results from a case study	conference	2017	1	1	1	1	1
S4	[67]	S. W. Ambler	Beyond functional requirements on agile projects strategies for addressing nonfunctional requirements	journal	2008	0.5	0.5	0	0.5	0
S5	[57]	W. Behutiye, P. Karhapää, D. Costal, M. Oivo, and X. Franch	Non-functional requirements documentation in agile software development: challenges and solution proposal	conference	2017	0.5	0.5	0.5	0	0
S6	[15]	M. Bourimi, T. Barth, J. M. Haake, B. Ueberschär, and D. Kesdogan	Affine for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies	conference	2010	1	1	1	0.5	0
S7	[68]	D. Çulha and A. Doğru	Towards an agile methodology for business process development	conference	2014	0.5	0	0	0.5	0
S8	[60]	S. Dragicevic, S. Celar, and L. Novak	Use of method for elicitation, documentation, and validation of software user requirements (MEDov) in agile software development projects	conference	2014	0.5	0.5	0	0.5	0
S9	[20]	W. M. Farid and F. J. Mitropoulos	Novel lightweight engineering artifacts for modeling non-functional requirements in agile processes	conference	2012	1	0.5	1	1	0.5
S10	[75]	W. M. Farid and F. J. Mitropoulos	NORPLAN: Non-functional requirements planning for agile processes	conference	2013	0.5	0.5	0.5	1	0.5
S11	[81]	A. Firdaus, I. Ghani, D. N. A. Jawawi, and W. M. N. W. Kadir,	Non functional requirements (NFRs) traceability metamodel for agile development	journal	2015	0.5	0.5	0.5	0.5	0.5
S12	[16]	X. Franch, *et al.*	Data-driven elicitation, assessment and documentation of quality requirements in agile software development	conference	2018	1	1	1	0.5	0.5
S13	[73]	T. de Gooijer	Discover quality requirements with the mini-QAW	conference	2017	0.5	0.5	0	0.5	0
S14	[82]	D. Ionita, C. *et al.*	Towards risk-driven security requirements management in agile software development	conference	2019	1	0.5	1	1	1
S15	[24]	M. Käpyaho and M. Kauppinen	Agile requirements engineering with prototyping: A case study	conference	2015	1	1	1	1	1
S16	[22]	S. Kopczyńska, M. Ochodek, and J. Nawrocki	On importance of non-functional requirements in agile software projects–a survey	conference	2019	1	1	1	1	0.5
S17	[66]	L. López, W. Behutiye, P. Karhapää, J. Ralyté, X. Franch, and M. Oivo	Agile quality requirements management best practices portfolio: A situational method engineering approach	conference	2017	0.5	0.5	0.5	1	0
S18	[77]	R. R. Maiti and F. J. Mitropoulos	Capturing, eliciting, and prioritizing (CEP) NFRs in agile software engineering	conference	2017	1	1	1	1	0.5
S19	[59]	R. Macasaet, L. Chung, J. L. Garrido, M. Noguera, and M. L. Rodríguez	An agile requirements elicitation approach based on NFRs and business process models for micro-businesses	conference	2011	0.5	0.5	0	0.5	0
S20	[71]	J. Medeiros, A. Vasconcelos, M. Goulão, C. Silva, and J. Araújo	An approach based on design practices to specify requirements in agile projects	conference	2017	0.5	0	1	1	0.5
S21	[74]	P. Mohagheghi and M. E. Aparicio	An industry experience report on managing product quality requirements in a large organization	journal	2017	0.5	1	0.5	1	0.5
S22	[17]	J. Nawrocki, M. Ochodek, J. Jurkiewicz, S. Kopczyńska, and B. Alchimowicz	Agile requirements engineering: A research perspective	conference	2014	0.5	0.5	0	1	0
S23	[79]	C. Patel and M. Ramachandran	Bridging best traditional SWD practices with XP to improve the quality of XP projects	conference	2008	0.5	0.5	0	0.5	0
S24	[19]	C. Patel and M. Ramachandran	Story card maturity model (SMM): A process improvement framework for agile requirements engineering practices	journal	2009	1	1	1	1	0.5
S25	[80]	C. Pecchia, M. Trincardi, and P. Di Bello	Expressing, managing, and validating user stories: Experiences from the market	conference	2016	0.5	0	0	0.5	0
S26	[6]	B. Ramesh, L. Cao, and R. Baskerville	Agile requirements engineering practices and challenges: an empirical study	journal	2010	1	1	1	1	1
S27	[76]	F. B. A. Ramos, *et al.*	A non-functional requirements recommendation system for Scrum-based projects	conference	2018	0.5	0.5	0.5	0.5	0.5
S28	[69]	V. Sachdeva and L. Chung	Handling non-functional requirements for Big Data and IOT projects in Scrum	conference	2017	0.5	0.5	0.5	0.5	0
S29	[78]	E. Terpstra, M. Daneva, and C. Wang	Agile practitioners' understanding of security requirements: insights from a grounded theory analysis	conference	2017	1	1	0.5	1	1
S30	[70]	T. Um, N. Kim, D. Lee, and H. P. In	A quality attributes evaluation method for an agile approach	conference	2011	0.5	0	0	0.5	0
S31	[72]	M. Younas, D. Jawawi, I. Ghani, and R. Kazmi,	Non-functional requirements elicitation guideline for agile methods	journal	2017	0.5	0	0.5	1	0.5

- Are the NFRs expressed by stakeholders generic/ambiguous (*e.g.* "the system must respond quickly") or specific/verifiable?
- How do project team members participating in requirements elicitation respond to generic/ambiguous NFRs? What do they do to clarify such requirements?

- Can the NFR elicitation practices used by the project team be considered effective? Or perhaps they result in some problems uncovered later (*e.g.* product presentation, acceptance tests)?

NFR documentation. Questions about documentation practices in software projects developed in the interviewee's current organization.

- How are requirements documented? Which requirements documentation techniques are used? Note: first allow the interviewee to answer freely, later ask about the techniques uncovered by the SLR study.
- What tool support is used to document requirements?
- Are different requirements documentation techniques used for the purpose of: a) eliciting requirements from stakeholders and validating them; b) communicating the requirements to developers, testers and other project team members?
- Are NFRs documented using the same techniques as other requirements or are there any techniques used solely for the purpose of NFRs specification?
- Is traceability between FRs and NFRs established and maintained? How?
- Is any informed choice regarding documenting NFRs made *e.g.* a) Documentation techniques which allow NFRs to be specified in a detailed and unambiguous way are selected; or b) "Lightweight", less detailed techniques are selected to document NFRs, but clarification is made by means of direct face to face communication?
- Can the NFR documentation techniques used by the project team be considered effective? Or perhaps they cause problems, *e.g.* the developers are unable to understand the documented NFRs or the product does not meet the customer's expectations?

Conclusion and closing remarks.

- Conclude the interview and thank interviewee for their participation.
- Ask if there are any additional questions or remarks.
- Notify the interviewee that a summary of this interview will be sent to him/her for authorization purposes.

APPENDIX B
AN EXAMPLE OF CODING APPLIED TO INTERVIEWS

See Table 10.

APPENDIX C
SLR STUDY - THE LIST OF QUALIFIED ARTICLES

See Table 11.

REFERENCES

[1] M. Kropp and A. Meier, "Swiss agile study 2014," Zürcher Hochschule für Angewandte Wissenschaften, Winterthur, Switzerland, Tech. Rep., 2014. [Online]. Available: http://www.swissagilestudy.ch/files/2015/05/SwissAgileStudy2014.pdf

[2] M. Kassab, "The changing landscape of requirements engineering practices over the past decade," in *Proc. IEEE 5th Int. Workshop Empirical Requirements Eng. (EmpiRE)*, Aug. 2015, pp. 1–8.

[3] S. Lauesen, "IT project failures, causes and cures," *IEEE Access*, vol. 8, pp. 72059–72067, 2020.

[4] T. Dreesen, P. Diegmann, B. Binzer, and C. Rosenkranz, "Journey towards agility—A retro-and prospective review," in *Proc. 52nd Hawaii Int. Conf. Syst. Sci.*, 2019, pp. 6950–6959.

[5] D. Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Reading, MA, USA: Addison-Wesley, 2010.

[6] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: An empirical study," *Inf. Syst. J.*, vol. 20, no. 5, pp. 449–480, Nov. 2007.

[7] V. T. Heikkilä, D. Damian, C. Lassenius, and M. Paasivaara, "A mapping study on requirements engineering in agile software development," in *Proc. 41st Euromicro Conf. Softw. Eng. Adv. Appl.*, Aug. 2015, pp. 199–207.

[8] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Hum. Behav.*, vol. 51, pp. 915–929, Oct. 2015.

[9] S. Wagner, D. Méndez Fernández, M. Kalinowski, and M. Felderer, "Agile requirements engineering in practice: Status quo and critical problems," *CLEI Electron. J.*, vol. 21, no. 1, p. 15, Apr. 2018.

[10] J. Medeiros, D. C. Alves, A. Vasconcelos, C. Silva, and E. Wanderley, "Requirements engineering in agile projects: A systematic mapping based in evidences of industry," in *Proc. Ibero-Amer. Conf. Softw. Eng. (CIBSE)*, 2015, pp. 460–476.

[11] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements in large-scale distributed agile projects—A systematic literature review," in *Proc. Int. Working Conf. Requirement Eng., Found. Softw. Qual.* Cham, Switzerland: Springer, 2017, pp. 219–234.

[12] A. Hussain, E. O. C. Mkpojiogu, and F. M. Kamal, "The role of requirements in the success or failure of software projects," *Int. Rev. Manage. Marketing*, vol. 6, no. 7, pp. 306–311, 2016.

[13] R. N. Charette. (2018). *The Biggest IT Failures of 2018*. Accessed: Jan. 21, 2021. [Online]. Available: https://spectrum.ieee.org/riskfactor/computing/it/it-failures-2018-all-the-old-familiar-faces

[14] F. Kolf and C. Kerkmann. (2018). *Lidl Software Disaster Another Example of Germany's Digital Failure*. [Online]. Available: https://www.handelsblatt.com/english/companies/programmed-for-disaster-lidl-software-disaster-another-example-of-germanys-digital-failure/23582902.html

[15] M. Bourimi, T. Barth, J. M. Haake, B. Ueberschär, and D. Kesdogan, "AFFINE for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies," in *Proc. Int. Conf. Hum.-Centred Softw. Eng.* Berlin, Germany: Springer, 2010, pp. 182–189.

[16] X. Franch, C. Gómez, A. Jedlitschka, L. López, S. Martínez-Fernández, M. Oriol, and J. Partanen, "Data-driven elicitation, assessment and documentation of quality requirements in agile software development," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer, 2018, pp. 587–602.

[17] J. Nawrocki, M. Ochodek, J. Jurkiewicz, S. Kopczyńska, and B. Alchimowicz, "Agile requirements engineering: A research perspective," in *Proc. 40th Int. Conf. Current Trends Theory Pract. Informat. (SOFSEM)*. Cham, Switzerland: Springer, 2014, pp. 40–51.

[18] B. Boehm, D. Rosenberg, and N. Siegel, "Critical quality factors for rapid, scalable, agile development," in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 514–515.

[19] C. Patel and M. Ramachandran, "Story card maturity model (SMM): A process improvement framework for agile requirements engineering practices," *J. Softw.*, vol. 4, no. 5, pp. 422–435, Jul. 2009.

[20] W. Farid and F. Mitropoulos, "Novel lightweight engineering artifacts for modeling non-functional requirements in agile processes," in *Proc. Southeastcon*, 2012, pp. 1–7.

[21] G. Rogers. (2016). *RE in Agile Projects: Survey Results*. [Online]. Available: https://re-magazine.ireb.org/print/re-in-agile-projects-survey-results

[22] S. Kopczyńska, M. Ochodek, and J. Nawrocki, "On importance of non-functional requirements in agile software projects—A survey," in *Integrating Research and Practice in Software Engineering*. Cham, Switzerland: Springer, 2020, pp. 145–158.

[23] J. Saltz, E. Anderson, and A. Sutherland, "Introduction to the minitrack on agile and lean: Organizations, products and development," in *Proc. 54th Hawaii Int. Conf. Syst. Sci.*, 2021, pp. 5423–5424.

[24] M. Käpyaho and M. Kauppinen, "Agile requirements engineering with prototyping: A case study," in *Proc. IEEE 23rd Int. Requirements Eng. Conf. (RE)*, Aug. 2015, pp. 334–343.

[25] W. Alsaqaf, M. Daneva, and R. Wieringa, "Agile quality requirements engineering challenges: First results from a case study," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Nov. 2017, pp. 454–459.
[26] V. Sachdeva, "Requirements prioritization in agile: Use of planning poker for maximizing return on investment," in *Information Technology—New Generations*, S. Latifi, Ed. Cham, Switzerland: Springer, 2018, pp. 403–409.
[27] L. Zamudio, J. A. Aguilar, C. Tripp, and S. Misra, "A requirements engineering techniques review in agile software development methods," in *Computational Science and Its Applications*. Cham, Switzerland: Springer, 2017, pp. 683–698.
[28] E.-M. Schön, D. Winter, M. J. Escalona, and J. Thomaschewski, "Key challenges in agile requirements engineering," in *Proc. Int. Conf. Agile Softw. Develop.* Cham, Switzerland: Springer, 2017, pp. 37–51.
[29] K. Elghariani and N. Kama, "Review on agile requirements engineering challenges," in *Proc. 3rd Int. Conf. Comput. Inf. Sci. (ICCOINS)*, Aug. 2016, pp. 507–512.
[30] P. Heck and A. Zaidman, "A systematic literature review on quality criteria for agile requirements specifications," *Softw. Qual. J.*, vol. 26, no. 1, pp. 127–160, Mar. 2018.
[31] K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr, "Requirements engineering: A systematic mapping study in agile software development," *J. Syst. Softw.*, vol. 139, pp. 32–50, May 2018.
[32] K. Redlarski and P. Weichbroth, "Hard lessons learned: Delivering usability in it projects," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Dec. 2016, pp. 1379–1382.
[33] Z. S. H. Abad, M. Noaeen, and G. Ruhe, "Requirements engineering visualization: A systematic literature review," in *Proc. IEEE 24th Int. Requirements Eng. Conf. (RE)*, Sep. 2016, pp. 6–15.
[34] P. Weichbroth and M. Sikorski, "User interface prototyping. Techniques, methods and tools," *Studia Ekonomiczne*, vol. 234, pp. 184–198, Dec. 2015.
[35] M. Attarha and N. Modiri, "Focusing on the importance and the role of requirement engineering," in *Proc. 4th Int. Conf. Interact. Sci.*, Dec. 2011, pp. 181–184.
[36] T. Clancy. (2014). *The Standish Group—Chaos Report 2014*. [Online]. Available: https://www.projectsmart.co.uk/white-papers/chaos-report.pdf
[37] D. Méndez Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, and T. Conte, "Naming the pain in requirements engineering," *Empirical Softw. Eng.*, vol. 22, no. 5, pp. 2298–2338, 2017. [Online]. Available: https://www.researchgate.net/profile/Daniel-Mendez-Fernandez
[38] J. D. Summers, S. Joshi, and B. Morkos, "Requirements evolution: Relating functional and non-functional requirement change on student project success," in *Proc. 16th Int. Conf. Adv. Vehicle Technol.*, Aug. 2014, Paper DETC2014-35023, V003T04A002, doi: 10.1115/DETC2014-35023.
[39] S. Kugele, W. Haberl, M. Tautschnig, and M. Wechs, "Optimizing automatic deployment using non-functional requirement annotations," in *Proc. Int. Symp. Leveraging Appl. Formal Methods, Verification Validation*. Berlin, Germany: Springer, 2008, pp. 400–414.
[40] *Systems and Software Engineering–Systems and Software Quality Requirements and Evaluation (SQuaRE)–Common Industry Format (CIF) for Usability–Evaluation Report*, Standard ISO/IEC 25066:2016(en), 2016.
[41] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, "A survey of user-centered design practice," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. Changing World*, 2002, pp. 471–478.
[42] G. Jurca, T. D. Hellmann, and F. Maurer, *Agile User-Centered Design*. Hoboken, NJ, USA: Wiley, 2018, p. 111.
[43] A. Jarzębowicz and K. Poniatowska, "Towards a lightweight approach for the evaluation of requirements engineering impact on other IT project areas," in *Integrating Research and Practice in Software Engineering*. Springer, 2020, pp. 171–186.
[44] M. Fowler and J. Highsmith, "The agile manifesto," *Softw. Develop.*, vol. 9, no. 8, pp. 28–35, 2001.
[45] I. Garnik, M. Sikorski, and G. Cockton, "Creative sprints: An unplanned broad agile evaluation and redesign process," in *Proc. 8th Nordic Conf. Hum.-Comput. Interact., Fun, Fast, Foundational*, Oct. 2014, pp. 1125–1130.
[46] P. Weichbroth, "Delivering usability in IT products: Empirical lessons from the field," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 28, no. 7, pp. 1027–1045, Jul. 2018.
[47] M. Younas, D. N. A. Jawawi, M. A. Shah, A. Mustafa, M. Awais, M. K. Ishfaq, and K. Wakil, "Elicitation of nonfunctional requirements in agile development using cloud computing environment," *IEEE Access*, vol. 8, pp. 209153–209162, 2020.
[48] S. L. Dorton, L. R. Maryeski, L. Ogren, I. T. Dykens, and A. Main, "A wargame-augmented knowledge elicitation method for the agile development of novel systems," *Systems*, vol. 8, no. 3, p. 27, Aug. 2020.
[49] K. Ossowska, L. Szewc, P. Weichbroth, I. Garnik, and M. Sikorski, "Exploring an ontological approach for user requirements elicitation in the design of online virtual agents," in *Proc. EuroSymp. Syst. Anal. Design*. Springer, 2016, pp. 40–55.
[50] M. Shafiq, Q. Zhang, M. A. Akbar, A. A. Khan, S. Hussain, F.-E. Amin, A. Khan, and A. A. and Soofi, "Effect of project management in requirements engineering and requirements change management processes for global software development," *IEEE Access*, vol. 6, pp. 25747–25763, 2018.
[51] T. Kaddoumi and M. Watfa, "A proposed agile enterprise architecture framework," in *Proc. 6th Int. Conf. Innov. Comput. Technol. (INTECH)*, Aug. 2016, pp. 52–57.
[52] M. U. Alhuseini and M. M. Olama, "5G service value chain and network slicing framework using ecosystem modeling, agile delivery, and user-story automation," *IEEE Access*, vol. 7, pp. 110856–110873, 2019.
[53] D. K. Rigby, J. Sutherland, and A. Noble, "Agile at scale," *Harvard Bus. Rev.*, vol. 96, no. 3, pp. 88–96, 2018.
[54] H. A. Mitre-Hernández, C. Lara-Alvarez, M. González-Salazar, and D. Martín, "Decreasing rework in video games development from a software engineering perspective," in *Trends and Applications in Software Engineering*. Springer, 2016, pp. 295–304.
[55] F Kišš and B. Rossi, "Agile to lean software development transformation: A systematic literature review," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2018, pp. 969–973.
[56] V. CollabNet. (2019). *13th Annual State of Agile Report*. [Online]. Available: https://stateofagile.com/
[57] W. Behutiye, P. Karhapää, D. Costal, M. Oivo, and X. Franch, "Non-functional requirements documentation in agile software development: Challenges and solution proposal," in *Proc. Int. Conf. Product-Focused Softw. Process Improvement*. Cham, Switzerland: Springer, 2017, pp. 515–522.
[58] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," School Comput. Sci. Math., Keele Univ., Keele, U.K., Tech. Rep. EBSE-2007-01, 2007.
[59] R. Macasaet, L. Chung, J. L. Garrido, M. Noguera, and M. L. Rodríguez, "An agile requirements elicitation approach based on NFRs and business process models for micro-businesses," in *Proc. 12th Int. Conf. Product Focused Softw. Develop. Process Improvement*, 2011, pp. 50–56.
[60] S. Dragicevic, S. Celar, and L. Novak, "Use of method for elicitation, documentation, and validation of software user requirements (MEDoV) in agile software development projects," in *Proc. 6th Int. Conf. Comput. Intell., Commun. Syst. Netw.*, May 2014, pp. 65–70.
[61] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, nos. 9–10, pp. 833–859, Aug. 2008.
[62] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012.
[63] I. Etikan, "Combination of probability random sampling method with non probability random sampling method (sampling versus sampling methods)," *Biometrics Biostatistics Int. J.*, vol. 5, no. 6, pp. 1–5, May 2017.
[64] K. Charmaz, *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*. Thousand Oaks, CA, USA: SAGE, 2006.
[65] W. Alsaqaf, "Engineering quality requirements in large scale distributed agile environment," in *Proc. 22nd Int. Work. Conf. Requirement Eng., Found. Softw. Qual. (REFSQ) Workshops*, 2016. [Online]. Available: http://ceur-ws.org/Vol-1564/
[66] L. López, W. Behutiye, P. Karhapää, J. Ralyté, X. Franch, and M. Oivo, "Agile quality requirements management best practices portfolio: A situational method engineering approach," in *Int. Conf. Product-Focused Softw. Process Improvement* 2017, pp. 548–555.
[67] S. W. Ambler, "Beyond functional requirements on agile projects-strategies for addressing nonfunctional requirements," *Dr. Dobb's J.*, vol. 33, no. 10, pp. 64–66, 2008.
[68] D. Çulha and A. Doğru, "Towards an agile methodology for business process development," in *Proc. Int. Conf. Subject-Oriented Bus. Process Manage.* Cham, Switzerland: Springer, 2014, pp. 133–142.

[69] V. Sachdeva and L. Chung, "Handling non-functional requirements for big data and IOT projects in scrum," in *Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2017, pp. 216–221.
[70] T. Um, N. Kim, D. Lee, and H. P. In, "A quality attributes evaluation method for an agile approach," in *Proc. 1st ACIS/JNU Int. Conf. Comput., Netw., Syst. Ind. Eng.*, May 2011, pp. 460–461.
[71] J. Medeiros, A. Vasconcelos, M. Goulão, C. Silva, and J. Araäjo, "An approach based on design practices to specify requirements in agile projects," in *Proc. Symp. Appl. Comput.*, Apr. 2017, pp. 1114–1121.
[72] M. Younas, D. N. Jawawi, I. Ghani, and R. Kazmi, "Non-functional requirements elicitation guideline for agile methods," *J. Telecommun., Electron. Comput. Eng.*, vol. 9, nos. 3–4, pp. 137–142, Oct. 2017.
[73] T. De Gooijer, "Discover quality requirements with the mini-QAW," in *Proc. IEEE Int. Conf. Softw. Archit. Workshops (ICSAW)*, Apr. 2017, pp. 196–198.
[74] P. Mohagheghi and M. E. Aparicio, "An industry experience report on managing product quality requirements in a large organization," *Inf. Softw. Technol.*, vol. 88, pp. 96–109, Aug. 2017.
[75] W. M. Farid and F. J. Mitropoulos, "NORPLAN: Non-functional requirements planning for agile processes," in *Proc. Southeastcon*, Oct. 2013, pp. 1–8.
[76] F. Ramos, A. A. M. Costa, M. Perkusich, H. Almeida, and A. Perkusich, "A non-functional requirements recommendation system for scrum-based projects," in *Proc. 30th Int. Conf. Softw. Eng. Knowl. Eng.*, Jul. 2018, pp. 148–149.
[77] R. R. Maiti and F. J. Mitropoulos, "Capturing, eliciting, and prioritizing (CEP) NFRs in agile software engineering," in *Proc. SoutheastCon*, 2017 pp. 1–7.
[78] E. Terpstra, M. Daneva, and C. Wang, "Agile Practitioners' understanding of security requirements: Insights from a grounded theory analysis," in *Proc. IEEE 25th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2017, pp. 439–442.
[79] C. Patel and M. Ramachandran, "Bridging best traditional SWD practices with XP to improve the quality of XP projects," in *Proc. Int. Symp. Comput. Sci. Appl.*, Oct. 2008, pp. 357–360.
[80] C. Pecchia, M. Trincardi, and P. Di Bello, "Expressing, managing, and validating user stories: Experiences from the market," in *Proc. 4th Int. Conf. Softw. Eng. Defence Appl.* Cham, Switzerland: Springer, 2016, pp. 103–111.
[81] A. Firdaus, I. Ghani, D. N. Abg Jawawi, and W. M. N. Wan Kadir, "Non functional requirements (NFRS) traceability metamodel for agile development," *Jurnal Teknologi*, vol. 77, no. 9, pp. 115–125, Nov. 2015.
[82] D. Ionita, C. van der Velden, H.-J. K. Ikkink, E. Neven, M. Daneva, and M. Kuipers, "Towards risk-driven security requirements management in agile software development," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer, 2019, pp. 133–144.
[83] B. M. Aljallabi and A. Mansour, "Enhancement approach for non-functional requirements analysis in agile environment," in *Proc. Int. Conf. Comput., Control, Netw., Electron. Embedded Syst. Eng. (ICCNEEE)*, Sep. 2015, pp. 428–433.
[84] A. Jarzębowicz and W. Ślesiński, "What is troubling IT analysts? A survey report from Poland on requirements-related problems," in *Engineering Software Systems: Research and Praxis*. Cham, Switzerland: Springer, 2019, pp. 3–19.
[85] Scopus. (2020). *Scopus Content Coverage Guide*. [Online]. Available: https://www.elsevier.com/solutions/scopus/how-scopus-works/content
[86] M. Daneva, D. Damian, A. Marchetto, and O. Pastor, "Empirical research methodologies and studies in requirements engineering: How far did we come?" *J. Syst. Softw.*, vol. 95, pp. 1–9, Sep. 2014.
[87] N. Rai and B. Thapa, *A Study on Purposive Sampling Method in Research*. Kathmandu, Nepal: Kathmandu School LaW, 2015.
[88] S. E. Hove and B. Anda, "Experiences from conducting semi-structured interviews in empirical software engineering research," in *Proc. 11th IEEE Int. Softw. Metrics Symp. (METRICS05)*, Sep. 2005, pp. 1–10.

ALEKSANDER JARZĘBOWICZ received the M.Sc. and Ph.D. degrees in software engineering from the Gdańsk University of Technology, in 2002 and 2007, respectively. He is currently an Assistant Professor with the Department of Software Engineering, Gdańsk University of Technology, where he is also the Deputy Head. He is also involved in the process of technology transfer conducted by the Gdańsk University of Technology's spin-off company named Argevide.

He has authored or coauthored more than 30 publications in journals and conference proceedings. He has participated in several EU-funded research and development projects. His current research interests include requirements engineering, agile software development, and the dependability of software-based systems with a particular interest in assurance cases.

PAWEŁ WEICHBROTH received the M.A. degree in statistics from the University of Gdańsk, Poland, in 2003, and the Ph.D. degree in artificial intelligence from the Katowice University of Economics, Poland, in 2014.

He was a Business Consultant and an IT Lecturer for more than 20 years. He is currently an Assistant Professor with the Department of Software Engineering, Gdańsk University of Technology. He has authored more than 40 research articles as journal and conference articles, and book chapters. His current research interests include software quality, machine learning, and knowledge management.

Dr. Weichbroth has been a member of the Scientific Community of Business Informatics, a member of several international conference program committees, and currently acting as a Reviewer in journals with an impact factor. Since 2018, he has been acting as an Expert of the Ministry of Digital Affairs in a project for the development of public digital services.