On Importance of Non-functional Requirements in Agile Software Projects—A Survey

Sylwia Kopczyńska, Mirosław Ochodek and Jerzy Nawrocki

Abstract *Context.* Among many Agile software development practices, over 30 concern Requirements Engineering (RE). However, none of them mentions explicitly non-functional requirements (NFRs). The question arises – how important are NFRs in Agile software projects? *Method.* We conducted a survey asking Agile software development practitioners how they perceive the importance of having NFRs defined in their projects. Then, we juxtaposed the answers with their opinions on the perceived importance of 31 Agile RE practices. *Results.* The opinions of 118 respondents from a wide range of countries around the globe allowed us to determine how important it is to define NFRs. Moreover, we showed their importance from the perspective of the ranking of Agile RE practices. We also identified some relationships between the demographic data such as experience and the perceived importance of requirements concerning quality. *Conclusions.* We found that over 77% of respondents perceive having NFRs defined in Agile software project as at least important, and for 30% it is critical. Also, the perceived importance of NFRs increases with the increase of respondents' experience.

Keywords Non-functional requirements · Importance · Agile · Agile practices · Survey

1 Introduction

Agile software development is built around Agile Manifesto [2] that is values and principles that shall be respected in every project. However, they are quite abstract and it could be difficult to decide how to implement them in a software project.

S. Kopczyńska (✉) · M. Ochodek · J. Nawrocki
Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland
e-mail: sylwia.kopczynska@cs.put.poznan.pl

M. Ochodek
e-mail: miroslaw.ochodek@cs.put.poznan.pl

J. Nawrocki
e-mail: jerzy.nawrocki@cs.put.poznan.pl

S. Jarzabek et al. (eds.), *Integrating Research and Practice in Software Engineering*, Studies in Computational Intelligence 851,
https://doi.org/10.1007/978-3-030-26574-8_11

Therefore, researchers started to investigate practices that are executed in agile software development projects that help teams achieve excellence. The practices are being cataloged (e.g., [19]), their adoption level, popularity, applicability are studied (e.g., [33, 35]). There have been over 30 agile practices that concern Requirements Engineering (RE) identified [19, 27].

Generally, requirements engineers use functional and non-functional requirements (NFRs) to describe a software product. The former describe so-called user-valued transactions (i.e., functionality that supports the users), and the latter state conditions under which the provided functionality is really useful (e.g., maximum response time).

NFRs play important role in any software project. In many cases a project failure can be traced, amongst others, to inappropriate management of NFRs (see e.g., [3, 4, 25, 26]). On the other hand NFRs are often neglected, especially those that are difficult to write or ostensibly obvious. Cao and Ramesh [6] showed that the problem of specifying NFRs also tackles agile software development projects. Moreover, none of the previously identified Agile RE practices mentions explicitly NFRs. Therefore, a question arises:

- **Are NFRs perceived as *important* in agile software projects?**

To answer the stated research question we asked Agile software development practitioners about their perceived importance of defining NFRs (see Sect. 2 for the design of the survey). Then, we analyzed the answers and compared them with the perceived importance of 31 Agile RE practices (see Sect. 6, and for validity threats refer to Sect. 4). We also showed how other researchers and practitioners assess importance of practices in agile software development and how they investigate the importance of NFRs from different perspectives (Sect. 5). The conclusions from our work were formulated in Sect. 6.

2 Survey Design

The goal of the study was to identify the perceived importance of NFRs in the context of agile software development projects. To achieve the goal we designed a survey using the guidelines by Kitchenham and Pfleeger [22].

2.1 Survey Instrument

To collect the opinions of a wide variety of respondents, we asked the participants of the Agile RE practices survey [27] to take part also in our survey. In an online questionnaire we asked about the perceived importance of the practice: "*How important is it to specify NFRs (to have them elicited and updated) for an agile project?*". Respondents were supposed to recall the project they had participated in and judge whether the practice was “critical”, “important”, or “additional” (helpful but also supplementary–could have been rejected without doing any harm to the projects). The respondents could choose the answer “other” and provide their own descriptive answer or could skip the question by selecting the “don’t have an opinion” answer.

Since it was required to participate in the Agile RE practices survey [27] first, we took advantage of this. In the analysis, we used the answers (1) on the perceived importance on the 31 Agile RE practices; (2) demographic information provided by the respondents who also answered the question concerning NFRs.

The short online one-question survey underwent proofreading by a professional linguist and a pilot study with three members of our laboratory. No problems were identified. The questionnaire was distributed using Google Docs Forms with the Agile RE survey.

2.2 Population

Our target population can be defined as agile software development project’s participants. We did not limit our focus to any specific types of applications or domains. However, we assumed that an individual belonging to the population needs to have at least one year of experience in agile software development. Our aim was to get a general view of the perceived importance of NFRs, which required to cover a wide group of people. Unfortunately, there is no obvious place in which we could identify and access the representatives of the defined population. Thus, we relied on non-systematic sampling methods (convenience sampling). Then, the accuracy concerning the error between population and sample cannot be determined [30]. However, we determined the characteristics of the respondents afterward to describe the sample.

The survey inherited from the Agile RE survey the two sampling approaches used in parallel to balance the strong and weak points of each method. First, we used self-recruiting [30]. We placed posts inviting to the Agile RE survey the members of social network groups related to agile and RE on LinkedIn, Facebook, Yahoo. Secondly, we sent direct invitations to people we knew to have experience in Agile, the members of Scrum Alliance using the communication tool available on their website, and to people who published their curricula vitae on the Internet (mainly on LinkedIn) indicating that they have experience in agile software development.

2.3 Data Validation and Analysis

We decided to validate the responses using the following criteria: (1) a participant had to answer more than 75% of questions from the Agile RE survey providing responses other than “I don’t have an opinion”, and (2) had at least one year of experience in agile software development. To analyze the responses we used the following methods and techniques:

- *frequency analysis* – to get a general overview of the opinions of the respondents;
- *relationship identification* – to investigate the relationships between the perceived importance of the practices and the responses given on the demographic questions in the Agile RE survey we used a non-parametric Kendall's τ correlation coefficient as the perceived importance of the practice was measured on an ordinal scale. Moreover, we used a test for association between paired samples for Kendall's τ [17] from the R statistical package with the significance level α set to 0.05.
- *ranking comparison* – we analyzed the perceived importance of the NFRs practice against the ranking proposed by the authors of the Agile RE survey. The aim was to see the relative importance of the NFRs practice that is over which Agile RE practices it is preferred.
- *open-text* – we used guidelines of Charmaz [7] to analyze the responses left by participants if they expressed their opinion on the perceived importance of the practice textually rather than choosing one of the given answers.

3 Results and Discussion

3.1 Respondents

Our survey followed the Agile RE survey [27] the invitations to which were sent in two runs: 09/10-20/12/2015 (the responses were collected until the end of June 2016), and 21/12/2017-11/01/2018 (the responses were collected until 28 February 2018). There were 147 responses to the Agile RE survey (2 of them did not meet the validation criteria) and 118 of respondents answered our survey concerning NFRs. We received the following numbers of responses from different sources:

- Social media interests groups related to Agile
  or Requirements Engineering = 76
- Direct invitations sent via Scrum Alliance community page = 12
- Direct invitation to known persons = 1
- Direct incitations based on CVs = 29

We obtained a wide coverage of countries around the globe in which the projects of our respondents were conducted (see Fig. 1).

The coverage of domains and types of software applications developed by participants was also high. As shown in Table 1, the four most dominating domains were services, financial, telecommunication, and banking. The most frequently developed types of applications were web and mobile applications, financial applications, and management information systems.

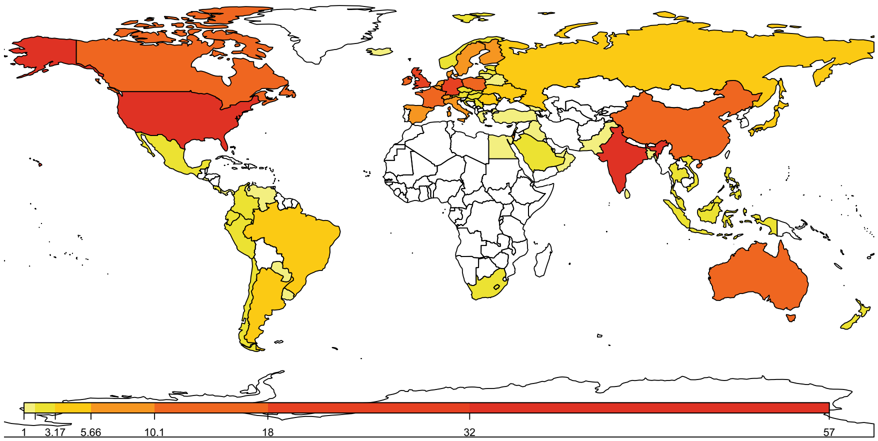


Fig. 1 Countries in which the Agile projects that respondents participated in were conducted (multiple choices allowed)

The most popular Agile methods were Scrum and Kanban. Slightly less popular were eXtreme Programming (XP) and Scrumban. Only a few respondents mentioned other methods such as DSDM, SAFe, or Crystal Clear.

The working experience also spread nearly through all of the activities we asked for in the survey. Although the dominating responsibilities were Scrum Master (or equivalent), project manager, and requirements analyst, we also had many participants with experience in quality assurance (QA), testing, programming, and software designing. The participants had on average 18 years of experience in SD and more than 8 years in Agile projects. The average number of Agile projects was 24; however, the meaning of the term project could be understood differently by the participants.

The participants had experience in Agile projects of different sizes, starting from small projects having up to five members of a team and total development effort smaller than one man-month, up to very big projects with more than 50 people involved and total development effort greater than 80 man-months.

3.2 Perceived Importance of NFRs Practice

The summary of the responses on the perceived importance of the NFRs practice is presented in Fig. 2.

It follows from the results that over **77%** of respondents (92 people) considers specifying NFRs as **at least important**. Interestingly, **30%** respondents (36) claim that the practice is **critical**.

Table 1 The results with respect ot the Agile projects in which the respondents participated

Domains			
(N = 116, multiple choices allowed)			
Services	53	Insurance	35
Financial	52	Entertainment	30
Banking	43	Government	27
Telecommunication	42	Trading	20
Electronics and computers	40	Manufacturing	17
Medical&Health Care	39	Other	32
Types of applications			
(N = 117, multiple choices allowed)			
Web or e-business	82	Document management	43
Mobile apps	61	Billing	33
Financial	56	Sales and marketing	32
Mgmt Inform. Systems	48	Logistics	28
Electronic data interchange	47	Personnel	23
Transaction or production systems	42	Trading	20
		Others	21
Methodologies			
(N = 90, multiple choices allowed)			
Scrum	86	Scumban	23
Kanban	52	DSDM	6
XP	34	Others	21
Responsibilities			
(N = 118, multiple choices allowed)			
Scrum master	91	Soft. Designer	58
Project manager	88	Programmer	53
Req. Analyst	80	Coach	13
TestingQA	60	Others	16
Size of teams			
(N = 118, multiple choices allowed)			
<= 5	57	20–30	19
5–10	92	30–50	20
10–15	53	>50	20
15–20	32		
Total development effort [man-month]			
(N = 118, multiple choices allowed)			
<= 1	21	8–20	66
1–2	33	20–40	49
2–4	54	40–80	42
4–8	59	>80	47

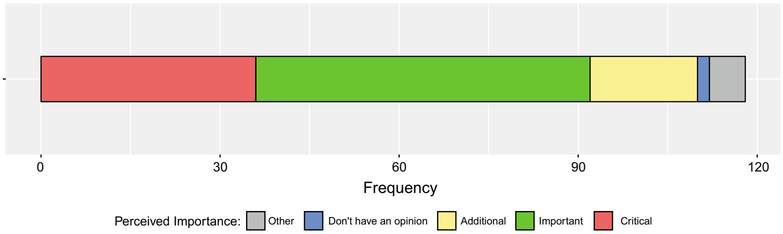


Fig. 2 Perceived importance of the practice concerning NFRs

3.3 Importance of NFRs Versus Other Practices

The ranking of relative importance of 31 Agile RE and the practice of defining NFRs is presented in Fig. 3.

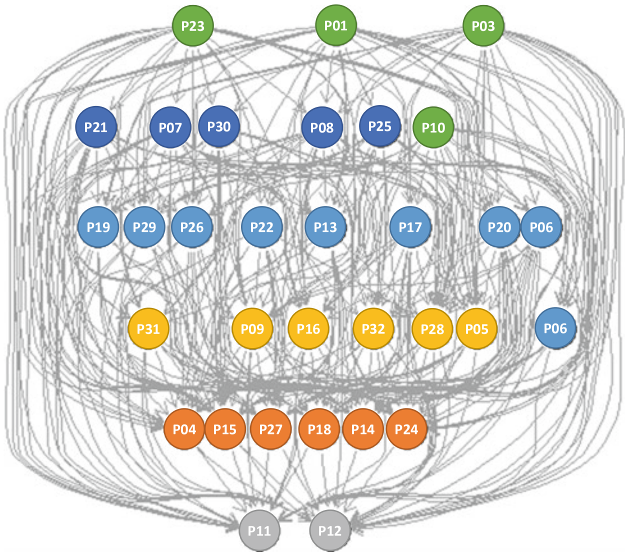
Based on the constructed graph of relative importance, the practices could be divided into *tiers*. The boundary of a tier is determined by practice P_A which has the lowest score and still is dominated only by practices from the higher tiers (tiers with lower index numbers).

The most important practices (Tier 1) could be characterized as those that seem fundamental for propelling a fast pacing, iterative, Agile project without a fully predefined scope and emergent requirements. The practice of defining NFRs is located at Tier 4. Similarly, its neighbor and a very similar practice P05: Define project and product constraints. First, it shows that consistency of our respondents as NFRs can be viewed as constraints. Secondly, it follows from the results, that among essential practices for Agile RE software development, although there are some practices perceived as more important, defining NFRs is still quite high – the fourth tier.

3.4 Perceived Importance of NFRs and Demographic Data

Following the method described in Sect. 2 we investigated whether the perceived importance of defining NFRs depends on the experience of Agile practitioners. The overall experience of respondents seems to have a very important role in how they perceive the importance of NFRs. We observed a correlation between defining NFRs and two factors (1) the number of years of experience in software development projects ($\tau = 0.21$, p-value = 0.005), and (2) the number years of experience in Agile software development projects ($\tau = 0.23$, p-value = 0.004).

It follows from our results that **the more experience** people have both generally in software projects and specifically in Agile software projects, **the more important** the practice of defining NFRs is perceived.



Tier	Practices
Tier 1	P01: Available/On-site Customer P03: Establish project shared vision P23: Make requirements testable P10: Provide easy access to requirements
Tier 2	P21: Requirements valuable to purchasers or users P07: Organize review meetings P30: Keep release length short P08: Organize retrospective meetings P25: Let customer prioritize requirements P22: Make requirements estimable
Tier 3	P19: Write short, negotiable requirements P29: Freeze release scope after it is agreed P26: Notation easy to understand by all stakeholders P13: Let customer define acceptance tests P17: Cover requirements with acceptance tests P20: Make complex requirements divisible P06: Organize everyday team meetings P02: Involve different stakeholders
Tier 4	P31: Define fixed release length P09: Provide informative workspace P16: Perform regression acceptance testing P32: Define quality requirements/non-functional reqs. P28: Negotiate release scope with customer P05: Define project/product constrains
Tier 5	P04: Create prototypes P15: Prepare acceptance tests before coding P27: Assess implementation risks for reqs. P18: Make requirements independent P14: Prepare and maintain automatic acceptance tests P24: Follow the user role modeling approach
Tier 6	P11: Maintain "bad smells" and best practices rel. to reqs. P12: Perform the Elevator Test

Fig. 3 The ranking of relative importance of the Agile RE practices enhanced with the practice of defining NFRs. The graph shows the preference degree $\pi(P_i, P_j)$ between the practices. The existence of an arc between P_i and P_j means that $\pi(P_i, P_j) = 1$

3.5 Analysis of Open-Text Questions

The analysis of open-text answers provided some interesting insights concerning NFRs. First, about the time when NFRs are important:

- after functional requirements are written – “*Important, **after functional requirements** have been written*”;
- early in the project – “*We need to know the non-functional requirements **eventually**; in some contexts we need to know them **very early in the project***”.

Similar insights had the participants of the survey concerning elicitation of NFRs [23], but they added also that in Agile projects NFRs shall be elicited and defined in the iterative approach throughout the project.

Secondly, two respondents pointed out that there are different forms of defining NFRs:

- as definition of done – “*should be agreed upon - in Scrum that is **the definition of ‘done’***”;
- as acceptance criteria – “*They should always be part of **user story acceptance criteria***”.

The same comments left the participants of the survey concerning elicitation of NFRs [23], but they added also that they are present also in definition of ready. The participants of the other survey also pointed out that those requirements are difficult to specify and to write, especially when discussing with user/client but two things could help:

- talking about consequences – that is either “*how much it would cost to implement a requirement*”;
- using abstract but easy to grasp terms by user and/or alternatives – for example asking “*would it be ok if the report refreshes once a day or shall it be live*”.

4 Validity Threats

The following analysis of the threats to validity to our study is based on the guidelines by Wohlin et al. [36].

Conclusion validity. There are two threats about the scale used to assess the perceived importance of the practice concerning NFRs and the same scale used in the Agile RE survey. First, if a respondent chose “don’t have an opinion” or “other”, then we were not able to use the information to evaluate the relative importance of the practice. Fortunately, there were only eight such respondents, which is negligibly small amount comparing to the number of all responses.

Secondly, the scale is subjective and might be interpreted differently by respondents depending on their background, culture, etc. We allowed providing open-text answers or resigning from providing any response to avoid biasing the results by forcing the respondents to answer about the practice they had not have any experience with or does not allow them to answer using the proposed scale. Thirdly, the scale is built around only positive (or neutral) influence of the practice on a project. We assumed that if a respondent had a negative experience with it, they would express that using the "other" option. One of the respondents shared such opinion stating that defining NFRs is pointless. Finally, we used a qualitative coding technique to analyze open-text responses. This approach is subjective by its nature, and we could have introduced some bias to the conclusions made based on the analysis.

Construct validity. Neither the practice of defining NFRs nor the 31 Agile RE practices used as the basis for the relative comparison, do not have formal and widely accepted definitions. Thus, they could have been understood by the respondents in a different way. Another threat relates to the word 'define' which, by looking at the names of other agile practices, frequently means 'specify'. In the study we treat them as synonyms. To mitigate the risks there were definitions provided for each practice. But still the understanding might have been different.

In the survey, we used the term non-functional requirements (NFRs) which also has no one agreed definition. By some practitioners and researchers, the term is used as the synonym to quality requirements (QRs), by some QRs are a subtype of NFRs [14]. Thus, the term could have been understood in a different ways.

To assure trust, transparency and repeatability, when designing and describing our study, we discussed and then reported as many details of the survey as possible. We followed the guidelines by Kitchenham and Pfleeger [22]. We also conducted a self-assessment using the Stavru's criteria [32] to verify if we reported all important elements. In the assessment we obtained the score equal to 0.89 (range 0–1). The result is higher than the highest score in the agile surveys examined by Stavru [32], which seems to be the sufficient level of detail.

Internal validity. Our control over collecting responses was limited. We are not able to reliably determine the number of people who received or read our invitation. Secondly, there is a question of trustworthiness of the respondents. They might have intentionally tried to influence the results of the surveys. However, the results of the sensitivity analysis run for the Agile survey [27] showed that it would not have a visible impact on the outcomes. Moreover, we could not find any reasonable rationale to explain bad intentions of the respondents.

The surveys were sent in two runs to receive more responses. To ensure that we ask each participant only once we added a question directly asking about the participation in the first run (the positive answer eliminated the participant's response from the further analysis).

Another threat to internal validity regards technical skills required to respond to the survey. Our assumption was that the members of the target population would not have problems in responding to an online survey and are fluent enough in English to understand the questions. We also conducted a pilot study to ensure correctness and ease of understating of the questionnaire correctness. A professional linguist also proofread the survey. We believe that the assumption was reasonable because English is a lingua franca of computer science. Besides, it turned out that ca. 86% of respondents of the Agile RE survey participated in projects conducted in the countries having English as an official language.

When it comes to the commitment of the respondents, we did not observe any suspicious responses that would suggest the lack of commitment or intent to sabotage the study. Besides, we believe that the topic was interesting to the participants because 72% of the Agile RE survey respondents voluntary shared their e-mail addresses to receive the summary of results.

The last two threats to internal validity regard the assessment of the importance of the practice. First, we do not know what particular events in the respondents' projects determined how they perceived the importance (e.g., was it a single project that failed because of lack of certain practices, or was a practice essential for all the projects?). Secondly, many of our respondents had long-term experience in agile projects, and thus their opinions could change in time (e.g., some facts from the earlier projects might have been blurred when they were filling in the questionnaire).

External validity. We intended to conduct our study among people who have experience in agile software development. We collected responses from participants taking part in projects undertaken in various countries and domains, also with experience in various project roles. Still, there are some threats related to the representativeness of the sample that we should consider. First, the sample of 118 respondents is still small comparing to the expected size of the population (however, this cannot be determined accurately). The sample may be too small and the obtained results could be different for a different sample. On the other hand, the results of the sensitivity analysis of the ranking conducted for the Agile RE survey show that the number of respondents seems sufficient to create a stable ranking of the practices.

The second threat regards using the non-systematic sampling methods. By sending invitations to social groups regarding agile and RE, we could reach people that are particularly interested in these topics (e.g., "Agile enthusiast"). Therefore, their view of agile practices might be somehow biased in comparison to other individuals in the population. We tried to mitigate this problem by sending direct invitations to people participating in Agile projects (balancing the sample with respect to the sources).

Finally, the responses of the participants reflect their subjective view of the subject. Without triangulation, we cannot firmly state whether the real importance is equivalent to the perceived one. However, we do not concern this threat as significant since the results of the Agile RE study seem, in general, convergent with the results of the studies regarding critical success factors in agile projects (e.g., [8]).

5 Related Work

Survey research studies are common in the context of Agile software development. There have been carried out many studies investigating adoption levels and popularity of Agile methods and practices. Some of these studies characterize usage of Agile methods in general (e.g., [24, 29]) other focus on particular countries (e.g., [5]) or concern applicability of Agile to particular domains, types of applications, or activities in projects (e.g., [18, 31]). There are also few studies focusing on the popularity of RE techniques and tools in Agile projects, e.g., [21, 34]. In general, the results of descriptive surveys seems attractive to practitioners since there are even surveys conducted annually by software vendors such as Version One [33]. This particular survey summarizes the current trends in the Agile community when it comes to methods, practices, and tools. It also aims at identifying the benefits and challenges of adopting Agile methods. However, despite its popularity, the survey received some criticism for lacking scientific rigor [32].

Williams [35] reported the results of a survey aiming at investigating which of the Agile practices are essential for a team to be considered Agile. Some insights into the importance of Agile practices are given by studies comparing Agile and non-Agile approaches to software development (e.g., Elshandidy and Mazen [11]), by systematic literature reviews (SLRs) or mapping studies discussing the benefits and challenges of Agile RE (e.g., [9, 19]), or by a clustering analysis of Agile practices by Fontana et al. [13].

Another relevant survey research study is the one conducted by Cao and Ramesh [6]. They surveyed 16 organizations about the degrees to which they follow Agile RE practices and what are benefits and challenges of using them. They found also that especially in Agile projects there are difficulties with eliciting NFRs.

On the other hand, there are also surveys trying to investigate the problems in software projects, e.g., [20] from which we can learn on which practices to focus.

In the area of NFRs, the researchers try to understand better the actions taken by project stakeholders connected with NFRs , e.g., [1, 10, 28]. There are also research initiatives that aim to understand the relationship between Agile software development and NFRs' awareness, e.g., [16]. Additionally, few works investigate project failures to provide evidence of what might happen when NFRs are treated as not important, e.g., [12]. Groen et al. [15] showed also that opinions of software product users might show which particular NFRs are essential.

6 Conclusions

The goal of our study was to investigate the importance of non-functional requirements (NFRs) in Agile software projects. To achieve the goal the respondents of Agile RE survey [27] were asked to provide their opinion concerning NFRs. Based on the answers of 118 participants having experience in software projects in wide variety of countries across the globe we formulated the following insights.

1. For over 77% of respondents the practice of defining NFRs is at least important; for 30% of respondents it is critical for agile project.
2. There are some more important agile practices such as establishing the vision of the project, having an available customer, ensuring that requirements are testable, etc. (see Fig. 3 for the all of them) than defining NFRs.

3. The longer the experience either in software projects generally or in agile software projects people have, the more important the practice of defining NFRs they find.

Some small fraction of respondents claimed that defining NFRs is an additional practice. The body of knowledge about agile software development indicates that specific practices might be more important in some contexts. Unfortunately, we were not able to identify any strong relationship between the perceived importance of defining NFRs and data on projects. Therefore, more studies are needed to understand how the perceived importance of NFRs depends on project context.

The participants also indicated that NFRs might be in some cases necessary early in the project lifecycle. Moreover, they mentioned that NFRs are used in different forms such as definition of done or acceptance tests. Thus, more research would be needed to investigate 'when', 'where', 'in which form', and 'how' NFRs are used in software projects.

Acknowledgements We would like to thank the participants of our study for sharing their opinions.

References

1. Ameller, D., Franch, X.: How do software architects consider Non-Functional Requirements: a survey. In: REFSQ (2010)
2. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for agile software development (2001). http://www.agilemanifesto.org/
3. Boehm, B., In, H.: Identifying quality-requirement conflicts. IEEE Softw. **13**(2), 25–35 (1996)
4. Breitman, K.K., Leite, J.C.S., Finkelstein, A.: The world sa stage: a survey on requirements engineering using a real-life case study. J. Braz. Comput. Soc. **6**(1), 13–37 (1999)
5. Buchalcevova, A.: Research of the use of Agile methodologies in the Czech Republic. In: Information Systems Development. Springer (2009)
6. Cao, L., Ramesh, B.: Agile requirements engineering practices: an empirical study. Software, IEEE **25**(1) (2008)
7. Charmaz, K.: Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis. Sage (2006)
8. Chow, T., Cao, D.B.: A survey study of critical success factors in agile software projects. J. Syst. Softw. **81**(6), 961–971 (2008)
9. Diebold, P., Dahlem, M.: Agile practices in practice: a mapping study. In: Proceeding of 18th International Conference on Evaluation and Assessment in Software Engineering. ACM (2014)
10. Eckhardt, J., Vogelsang, A., Fernández, D.M.: Are "non-functional" requirements really non-functional? an investigation of non-functional requirements in practice. In: International Conference on Software Engineering (ICSE), pp. 832–842. IEEE/ACM (2016)
11. Elshandidy, H., Mazen, S.: Agile and traditional requirements engineering: a survey. Internat. J. Sci. Eng. Res. **9**, (2013)
12. Finkelstein, A., Dowell, J.: A comedy of errors: the London Ambulance Service case study. In: Proceeding on 8th International Work Software specific Design, p. 2 (1996)
13. Fontana, R.M., Fontana, I.M., da Rosa Garbuio, P.A., Reinehr, S., Malucelli, A.: Processes versus people: how should agile software development maturity be defined? J. Syst. Softw. **97**, (2014)

14. Glinz, M.: On non-functional requirements. In: 15th IEEE International Requirements Engineering Conference (RE 2007), pp. 21–26 (2007)
15. Groen, E., Kopczynska, S., Hauer, M., Krafft, T.D., Doerr, J.: Users – the hidden software product quality experts. In: International Requirements Engineering Conference (2017)
16. Guzman, L., Oriol, M., Rodríguez, P., Franch, X., Jedlitschka, A., Oivo, M.: How can quality awareness support rapid software development? In: REFSQ (2017)
17. Hollander, M., Wolfe, D.A.: Nonparametric statistical methods. Wiley (1973)
18. Hussain, Z., Slany, W., Holzinger, A.: Current state of agile user-centered design: a survey. In: Proceeding of 5th Symposium of the Workgroup HCI and Usability Engineering of the Austrian Computer Society on HCI and Usability for e-Inclusion. Springer (2009)
19. Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S.: A systematic literature review on agile requirements engineering practices and challenges. Comput. Hum. Behav. **51**, (2015)
20. Kalinowski, M., Felderer, M., Conte, T., Spinola, R., Prikladnicki, R., Winkler, D., Fernández, D.M., Wagner, S.: Preventing incomplete/hidden requirements: reflections on survey data from Austria and Brazil. In: International Conference on Software Quality, pp. 63–78. Springer (2016)
21. Kassab, M.: An empirical study on the requirements engineering practices for agile software development. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 254–261. IEEE (2014)
22. Kitchenham, B.A., Pfleeger, S.L.: In: Shull, F., Singer, J., Sjøberg D. (eds.) Guide to Advanced Empirical Software Engineering, vol. 93, Chap. Personal Opinion Surveys. Springer (2008)
23. Kopczynska, S., Ochodek, M., Nawrocki, J.: An empirical study on catalog of non-functional requirement templates: Usefulness and maintenance issues. Inf. Softw, Technol (2018)
24. Kurapati, N., Manyam, V.S.C., Petersen, K.: Agile software development practice adoption survey. In: Agile processes in software engineering and extreme programming, pp. 16–30. Springer (2012)
25. Lindstrom, D.R.: Five ways to destroy a development project. IEEE Softw. **10**(5), 55–58 (1993)
26. Nuseibeh, B.: Ariane 5: who dunnit? IEEE Softw. **14**(3), 15–16 (1997)
27. Ochodek, M., Kopczynska, S.: Perceived importance of agile requirements engineering practices - a survey. J. Syst. Softw. **143**, (2018)
28. Palomares, C., Quer, C., Franch, X.: Requirements reuse and requirement patterns: a state of the practice survey. Empir. Softw. Eng. pp. 1–44 (2015)
29. Papatheocharous, E., Andreou, A.S.: Empirical evidence and state of practice of software agile teams. J. Softw.: Evol. Process. **26**(9) (2014)
30. Punter, T., Ciolkowski, M., Freimut, B., John, I.: Conducting on-line surveys in software engineering. In: Proceedings of the International Symposium on Empirical Software Engineering (ISESE). IEEE (2003)
31. Salo, O., Abrahamsson, P.: Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. IET Softw. **2**(1) (2008)
32. Stavru, S.: A critical examination of recent industrial surveys on agile method usage. J. Syst. Softw. **94**, (2014)
33. VersionOne inc: The 11th annual state of agile report. Tech. rep., VersionOne inc. (2017)
34. Wang, X., Zhao, L., Wang, Y., Sun, J.: The role of requirements engineering practices in agile development: an empirical study. In: Proceeding of 1st Asia Pacific Requirements Engineering Symposium, APRES. Springer-Verlag Berlin Heidelberg (2014)
35. Williams, L.: What agile teams think of agile principles. Communications of the ACM **55**(4), (2012)
36. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer Science & Business Media (2012)