

Agile Team Members Perceptions on Non-Functional Testing

Influencing Factors from an Empirical Study

Cristina Rosa Camacho
PUCRS University
Porto Alegre – RS – Brazil
cristina.camacho@acad.pucrs.br

Sabrina Marczak
PUCRS University
Porto Alegre – RS – Brazil
sabrina.marczak@pucrs.br

Daniela S. Cruzes
SINTEF - ICT
Trondheim – Norway
danielac@sintef.no

Abstract— Non-functional requirements define the overall qualities or attributes of a system. Although important, they are often neglected for many reasons, such as pressure of time and budget. In agile software development, there is a focus on the feature implementation and delivery of value to the customer and, as such, non-functional aspects of a system should also be of attention. Non-functional requirements testing is challenging due its cross-functional aspects and lack of clarity of their needs by business in the most part of projects. The goal of this paper is to empirically investigate how do agile team members handle non-functional testing in their projects, aiming to identify preliminary factors influencing the testing of non-functional requirements, specifically performance and security in agile development. We conducted interviews with twenty IT professionals in large multinational company. As result we could identify seven main factors influencing non-functional testing and four main practices adopted by them to overcome the challenges faced. We aim to replicate our investigation in a larger scale. Meanwhile, our work provides initial contributions to practitioners and inspires our future research.

Keywords; *non-functional testing, agile development, factors, empirical study.*

I. INTRODUCTION

In 2001, agile development has arisen aiming for a way to deliver working software and to add business value earlier in the software development process. Communication, working software, quality, and cross-functional teams are values highly appreciated in agile development as described in the Agile Manifesto [4].

Up to that moment, software testing was highly dependent on requirements and testing specialists dedicated to transform those requirements into test cases within the overall goal of system quality. In the agile context, this scenario is not anymore possible, once that there is not an established phase of requirements elicitation and analysis that produce a set of requirements that can be used as input for the testing process. On the traditional waterfall development, it was already noticed the complexity of defining, describing, and prioritizing non-functional requirements [22][9] due to issues such as the natural of the non-functional requirement of being general and not exactly "do" something as features and functions do. Also, due to the characteristic of being extremely related to the system or line of business that the application under development will attend [8][22]. Usually, non-

functional requirements lose priority against functional requirements, even when some functional requirements are affected when non-functional characteristics are not considered.

Along with that, the testing of non-functional requirements has not been taken seriously [26] and it is very often classified as low-risk due to its characteristics [11][23]. Non-functional testing requires long time of execution and an open minded approach. The necessity of an overall approach and the necessity of a long execution time can be also listed as an additional concern since agile development brings a focus on the feature implementation and faster delivery of value to the customer (generally functional requirements), bringing even more difficulty to identify non-functional aspects [10]. Non-functional testing needs can emerge from different sources, from business, as a customer need [10], as a technical piece of a given requirement, as part of an architectural change [3], or as part of the production behavior, which in agile would be for example part of the DevOps integration [16].

In this paper, we focus on the identification of the factors that influence the testing process of non-functional requirements (security and performance) in practice and, how team members handle those factors during their daily work. The following questions were posed:

- 1) What are the factors influencing testing of non-functional requirements in agile teams?
- 2) How do agile teams handle the identified factors?

The remainder of this paper is organized as follows: Section II summarizes literature in non-functional requirements and testing. Section III describes the methodology followed in our study. Section IV presents the results and Section V discusses them, including implications to research and practice as well as the study limitations. Section VI concludes the paper with our final remarks and future work.

II. BACKGROUND

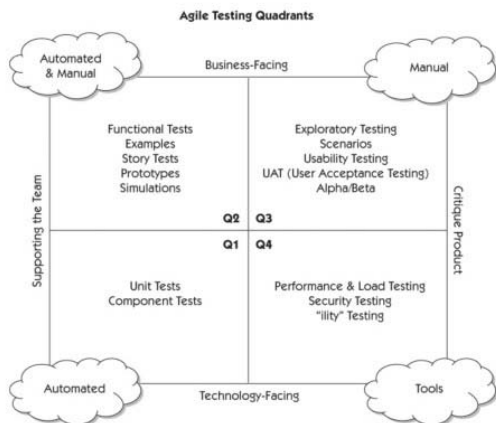
Non-functional requirements (NFRs), also known as technical requirements, quality of service, cross-cutting, or quality attributes focus on aspects that typically involve or cross-cut several functional requirements [1]. Although considered important and crucial to project success [19], it is common to see non-functional requirements losing attention in comparison to functional requirements [18].

The customer is the central point of agile development. However, many customers may focus on the business side of the application and do not necessarily understand the critical aspect of many non-functional requirements.

Crispin and Gregory (2009) [10] argue that with that business partners might assume that the development team will take care of non-functional requirements such as performance, reliability, and security, compromising the final product.

In addition, due to the agile philosophy that stimulates delivering user value early and often the prioritization of quality attributes can be hard in early deliverables increments resulting in hard-to-modify, unreliable, slow, or in-secure systems [2] [5] [30].

Testing happens for a lot of reasons. Different types of testing can be exercised to achieve different goals depending of the feature and software process which is being followed. This is not different in Agile software development, where development can even start from the test perspective [31]. Crispin and Gregory (2009) [10] organized the different types of agile testing according different perspectives [11], named the agile testing quadrants, as showed in Figure 1. The quadrants represent different testing approaches.



1. Agile Testing Quadrants, Crispin and Gregory (2009)

Tests executed in the Quadrants 1 and 2 are more related to requirements specification and design. Tests of Quadrant 1 are more related to design where unit tests and component tests ensure that code is not breaking. These tests are executed in regular basis, when a new code is promoted and a new deployment will be executed for example. The automation is generally "built-in" and part of the deployment process. Tests pertained to Quadrant 2 can be associated to the requirements design, since prototypes, examples, story tasks and simulations are used to validate if the understanding of the application was correct while functional testing aims to ensure that the code is doing what it should do.

However, testing pertained to Quadrants 3 and 4 criticizes the product a Business and Technical perspective. Tests of Quadrant 3 are related to Business scenarios, in order to ensure that the functionality is attending the business demands and generally are manual. Tests part Quadrant 4 are classified as technical, since specialized knowledge might be required to a better analysis of results or management of the required tools.

Performance testing is a technical investigation in order to validate several characteristics of a system such as: speed, velocity, scalability, and/or stability of a given system [20]. Generally related in improving the system in terms of response times, throughput, or resource-utilization

in order to meet the performance objectives of the application under test [11]. There are different types of testing approaches in order to reach different goals, such as:

- Performance test: executed to validate response time, scalability, and/or stability.
- Load test: executed to verify the system behavior under normal and peak load conditions.
- Stress test: executed to determine system's behavior under huge (peak) load conditions.

Security testing aims to validate the system in terms of security vulnerabilities such as data protection, confidentiality, integrity, authentication, availability and authorization issues [11]. Security testing is highly dependent on the security requirements imposed by the system line of business, e.g. a financial application. Several types of security test can be applied depending of the system, such as: Vulnerability scan, Vulnerability Assessment, Security Assessment, and Penetration Test [10] [11][15].

Due to the agile philosophy that stimulates delivering user value early and often the prioritization of quality attributes can be hard in early deliverables increments resulting in hard-to-modify, unreliable, slow, or insecure systems [6] [2] [30]. It is known that non-functional or cross-functional issues are classified as a low-risk for many systems and as consequence are not included as part of the testing tasks, even that know that can promote a high risk depending of the business with hacker attacks [17].

Even though that not all projects are concerned about non-functional testing, non-functional requirements should at least be checked in order to ensure they will be taken into consideration by the team and the customer will be asked how important they are for its business [10] [16]. These studies highlight that time to think about non-functional tests is during release or sprints planning. However, it is difficult to start testing without requirements [13].

Another issue is that agile development teams are generally composed by a small number of developers, who also many times act as testers [27]. However, some non-functional testing such as performance requires specialized tools such as profilers and might need specialized knowledge [10]. Given this need for specialized knowledge, a team member with specialized skills might be required for project success or to avoid issues in production [11]. Programmers might not be aware that non-functional testing such as performance and security might be a high priority and key to quality which touch a cultural mindset change. In addition, due to its nature of involving so many features of a given system, non-functional testing cannot be executed as part of an unit test, and as commonly take time and cannot be executed in a normal-continuous-integration-system cycle [5].

III. RESEARCH METHODOLOGY

A. Data Collection

In the study presented in this paper, we aim to investigate what are the factors influencing the testing of non-functional requirements in agile teams and how agile team members handle with those factors in order to

succeed. To answer our research questions, we conducted semi-structured interviews with twenty participants whose profiles are shown in Table 1.

TABLE I. PARTICIPANT PROFILE

Number of Participants	Role	Experience	Country
3	Performance Engineer	1-5 years	Brazil
2	Performance Engineer	6-15 years	Brazil
1	Performance Engineer	1-5 years	US
1	Performance Engineer	6-15 years	US
5	Performance Engineer	6-15 years	India
1	Software Architect	10 – 15 years	Brazil
1	Software Architect	10 – 15 years	US
1	Software Architect	10 – 15 years	US
1	Product Owner	10 – 15 years	US
1	Product Owner	10 – 15 years	India
1	Developer	10 – 15 years	Brazil
1	Developer	10 – 15 years	US
1	Developer	More than 15 years	US

The selection of the participants was made by convenience. The average duration time for the interviews was 35 minutes. Interviews with Brazilian and American participants were conducted face-to-face and remotely with participants from India. Detailed notes were taken during each interview. The Table 2 displays the interview guide.

B. Data Analysis

We used thematic analysis [25] to analyze the data from the semi-structured interviews we performed with team members. We first started generating the initial codes from the participants' citations with the support of ATLAS.ti qualitative data analysis tool. After that we searched for themes among codes. Once themes were identified and listed, we grouped them by factors, having a total of seven general main influencing factors encountered. We then classified these issues according to the testing challenges and, finally, the researchers interpreted and discussed the findings altogether in order to reach consensus.

C. Context

We investigated a large IT multinational company with offices spread out worldwide. Development teams are distributed in three locations: in the headquarters' office in the US, in Brazil and in India. The IT department follows a matrix organizational structure based on business areas (e.g., sales, manufacturing). Inside each business areas we have the development organization composed by developers, software architects and functional testers working as part of the feature teams. Usability, testing across multiple systems, called end to end testing and performance testers pertain to a dedicated organization focused on quality. Team members are either employees or contractors. Projects vary from the development of new products to the maintenance of legacy systems, including

the integration of applications and the customization of commercial packages.

TABLE II. SEMI-STRUCTURED INTERVIEW

Interview Guide	
1.	Could you please explain how non-functional requirements (performance and security) are elicited and tracked in your projects (key successful and unsuccessful projects)?
2.	When? In which moment this exercise is done?
3.	Following question #1, regarding non-functional requirements, what were the challenges faced and how did your team overcome them?
4.	Usually, do you need to perform non-functional testing (performance and security) in your projects?
5.	How non-functional testing (performance and security) are planned, defined, tracked, documented, prioritized and executed in your projects?
6.	Do you have non-functional specialists in your team? Who is responsible for non-functional testing (performance and security)?
7.	Literature says that non-functional testing (performance and security) are often neglected due to timing, budget or other restrictions. In your opinion, what are the factors that influence this situation and how do you proceed to avoid it?
8.	In your opinion what should be done to have a better non-functional testing coverage / prioritization in agile projects?

Project scope and deliverables frequency are defined based on prioritized business needs identified with the help of business representatives and product owners.

IV. RESULTS

During the data analysis, we identified seven main influencing factor coming from the interviews with testers, developers, software architects and product owners. The codes from the sources of information were organized into seven main themes: priority, time pressure, cost, technical issues, awareness, culture and experience.

A. Priority

During the analysis we identified that respondents consider the priority, which in this context means, the reasons for why the non-functional testing would add value in that moment to the business. The testing priority was reported as the main key factor impacting non-functional testing activities. A performance manager with more than fifteen years of experience and responsible for eighteen performance engineers (including badges and contractors) commented the following: "*Focus – it depends. moment, business priority, budget, feature characteristic*", P11.

However, many different aspects can determine the prioritization of the non-functional testing, such as feature/system characteristics, project type and criticality to business, as described below.

a.1 Feature / System Characteristic

The majority of the participants responded that the priority of the non-functional testing can be determinate based on its system characteristics, such as: 1) system type, e.g.: an internet banking with would require high security. 2) Customer behavior and user experience on a system which implies the needed of high response times or high performance. 3) Trend analysis of a system, e.g.: It is expected a huge trend of grow in resource utilization or data volume.

The P5 is a senior performance engineer with more than fourteen years working in the performance testing. During his interview he emphasized the priority based on the system type: *“Understanding the needs by the domain and technology of the application and the NFR as well as the testing. It differs based on domain w.r.to technologies like ecommerce, banking, networking, retail or client server based”*. P8, a senior product owner with more than fifteen years of experience (thirteen as business analyst and two as product owner), reinforce the characteristic highlighted by P5: *“Sensitivity of the data (how much potential damage would be caused by inadvertent disclosure) is an important item when defining security testing needs”*.

a.2. Project Type

According to the respondents, the priority also depends on the project type, e.g., a given project can handle the needed of the development of a brand new system or address the needed of development of a few enhancements in the software or just small updates or defect fixes. This was highlighted for a few interviewers and exemplified in the comment of the P10, a senior architect with more than fifteen years of experience: *“...depending on the moment, non-functional testing will get priority or not. For example in case it is a new system, then, what needs to receive the priority? The functionality itself because without it the non-functional will not exists as well once we don't have a system. However, the second priority will be non-functional, depending on the functionality, based on the business system's characteristics”*.

a.3 Criticality to Business

Criticality to business in this context means, the assessment of what are the client or business expectations about non-functional requirements and how the testing of non-functional requirements would impact the core business. These aspects are exemplified in the quotations from participants as P3: *“Business requirements & client expectations are the major driving forces to arrive at non-functional testing.”* (P3). And P6 commented: *“Non-functional testing is prioritized on the basis of criticality to business or end-user and release date if planned for a particular feature.”* P6.

B. Time Pressure

When defining priority of testing, time pressure was also largely cited as an influencing factor. Developers and testers state that functional testing takes the prioritization over performance testing due to development short time for a given sprint or release. However, according them, security is primordial and time for security testing is always guaranteed. That view is exemplified by the

comment below from a senior performance engineer: *“There is barely enough time to even complete the functional testing, and non-functional testing will always be lower priority than functional testing. Not sure we've solved it, but a possibility would be to reduce sprint velocity (commit to fewer story points) to allow sufficient time for both functional and non-functional testing.”*, P4.

Also, this is also reinforced by a daily challenge faced by a senior management responsible for performance: *“My team face issues regarding prioritization every single day. Teams are very busy with the functionality itself not with Performance, even when aware that it is important. Actually, the most senior developers understand that performance is important, however the functionality comes first.”* P11.

However, as stated earlier, security has always its space, as highlighted by a senior software architect: *“Security always have its space. Nothing goes to production without a set of Security testing – Fortify and then penetration test.”* P9.

C. Cost

Respondents consider the cost as one of the key factors impacting non-functional testing. However, the cost can be divided into two different meanings or contexts. In one hand they considered the cost of failure as a factor to determine the execution of non-functional testing. In this context, an analysis over which would cause an impact to business in case of failure of some non-functional aspect. As example, it was cited the cost with regards money in case of having the system down during an important sales period, or the damage for the brand in case of a security failure: *“The overall cost to fix to fix production performance, brand impact, revenue loss due to downtime are the key factors that influence the need for performance testing”*, P2.

In addition, respondents empathized the development cost aligned with return of investment as the factor that influences the decision of having non-functional testing or not: *“The non-functional requirements should be prioritized based on the cost of development and cost that will be incurred if the system fails to satisfy these kind of requirements which the market expects.”*, P4

However, according P7 there is also a budget issue, due to the lack of clarity from the client or customer of the impact on the business that can be generated as a result of a bad decision of not having budget allocated to non-functional testing: *“Challenges are to get the Client to realize that funds need to be allocated for these kind of requirements. Unfortunately, most don't understand how important they are until after an attack or performance issue has already occurred.”*, P17. The budget restriction was also emphasized by P15 an senior product owner: *“Security is critical to control the IT assets and prevent the loss of data, online fraud or theft. The challenge arises with the limitation of budget and conflicting requirements.”* P15.

D. Technical Issues

Technical issues in this context means, issues related to the system code deployed into production environment as a factor that may influence the testing of non-functional requirements in the next releases or any resource or environment issue that can avoid or impact or invalidate

the execution of non-functional testing. On the technical issues we have identified three different subcategories: production incidents,

d.1 Production Incidents

Respondents reported cases where the teams members (business, product owners, functional testers and developers) acted in a retroactive way and just analyzed and considered the needed of non-functional testing after facing a real issue in the production environment. *“Traditionally performance was always a reactive reaction where in the performance impact was seen in the production site.” P3.*

This view is also highlighted by P19, a senior developer with more than 20 years of experience and 10+ experience in agile, which worked in several multinational organizations in across the United States.

“I’ve seen three kinds of clients:

a. Clients that “get it” and understand the importance. Few and far between though. These clients have usually had bad experiences in the past in these areas and are aware of the impact that can be had.

b. Clients that “don’t get it” and unfortunately get burned by ignoring these requirements. I’ve seen everything from lawsuits for Security breaches/hacked apps to application servers being brought to their knees because of bad performance.

c. Clients that “don’t get it” and just haven’t been burned yet.”, P19.

d.2 Resource Utilization

Resource utilization in this context is the analysis of the need of non-functional testing, specifically performance in this case, depending on the expected or assessment of hardware / resource utilization required for a given system.

“Factors that influence performance testing depends on the type of application under test. However, throughput rate, response times, environment availability and system resource utilization such as memory, heap, CPU, disk space, network are common influencing factors.”, P7.

d.3 Environment

Respondents also emphasized the lack of a proper environment as a factor influencing the execution of the the non-functional testing. During the interviews they highlighted the concerns regarding the environment size and configuration setup. For instances, results obtained through a test executed against a performance environment too small or that does not reflect the system architecture may not be accurate.

E. Awareness

Respondents considered performance and security testing as part as technical coverage. They considered that factors such as business, developers and product owner’s awareness with regards the importance of at least the analysis of the non-functional aspects as an important factor that influences the non-functional testing. They stated that the knowledge is inside the people minds, but for some reason they are not externalized or documented. The quote below highlights the issue or not having business aware of the importance of the non-functional

requirements which as per P8 influences the non-functional testing: *“Business does not care about technical requirements, they ask for a thing and want that thing working.”, P8.*

The lack of overall clarity and description of non-functional requirements expectations from business and developers was also listed as a factor impacting non-functional testing.

“Challenges are business, developers know the requirements sub-consciously but when time come to lay it down for evaluating performance, and they go at the back front. Nothing comes explicitly from them which leads to ambiguity.

F. Culture

Culture in this context means to have the habit of thinking and considering non-functional testing in the development tasks. Participants provided some examples of how culture is an influencing factor, as listed below:

“Developers (especially agile developers) need to think on all things (functional + non-functional requirements when planning (sizing) and development a given feature.”, P10.

According to the respondents, culture is also important when planning a product. For instances, P18 commented about that instead of having a big product with a lot of functionalities, in the first releases if could have less features, but fast and secure.

“Unfortunately, many businesses don’t consider these kind of requirements important and would rather produce a feature-rich product rather than one which is secure and more performant but with fewer features.” P18.

G. Experience

Experience and skills set. Participants reported that generally the most experienced team members provide more attention to the non-functional aspects and defend the non-functional testing. In their view this happens due to past experiences they had in the past. The emphasize that mainly younger team members concentrate in what needs to be delivered (mostly functional items) and don’t think in the overall system or cross-functional items / features.

Unexperienced team members as a factor:

“...lack of knowledge from business teams about performance; developers never worked with performance team; unexperienced/non-technical performance engineers are factors that influence non-functional testing such as performance and security”, P12.

Challenges on having junior team members considering cross-functional aspects.

“Senior team members have this overall thinking in mind but generally they keep it for them and do not document it. Then junior developers will not learn as fast as they could. They think ahead but generally don’t expose, document it in a way that the whole team can access it. The knowledge stays in the head of the people.”, P2.

Challenges on having junior developers taking the ownership of the overall system quality.

“Younger developers basically want to deliver their piece of code and that’s it. Also, Performance is not an easy task, there is a lot of analysis required, and the most experienced folks should be assigned to work with the performance engineer on performance analysis and

tuning, due to the complexity involved. Also, there is a huge difference between performance engineering and performance testing, in the performance engineering you need to analyze all layers involved and purpose solutions, you need to think ahead.” P11.

Also, the stated that non-functional testing is a complicated task with requires expertise or specific skills.

“...performance testing is a complicated task since it involves architecture, infrastructure and several layers.” P19.

Experience aligned with technical skills and a solution oriented mindset.

“Yes, we faced challenges related to non-functional (performance and security) aspects in the projects I worked on. The knowledge and experience of the technical team was the key point to find the best solutions with regards these aspects.”, P18.

Experience aligned to testing skills and expertise in non-functional testing.

“I believe non-functional testing is always neglected due to the lack of experience of the teams and its inherent difficult in identify the testing needs and execution of the testing itself”, P4.

V. DISCUSSION

1) What are the factors influencing testing of non-functional requirements in agile teams?

When analyzing the results obtained, it was possible to observe factors already discussed in the literature such as cost and time pressure [7][12][14]but also aspects that can be more investigated such as experience. Experience was largely cited by respondents as being important factor influencing non-functional testing mainly due to their capacity of better identification of non-functional testing needs based on their past experiences. In additional, it was possible to identify that senior members also influence in the team culture with regards non-functional testing. The awareness and culture of non-functional testing importance was highly cited by experienced members and can also support to minimize other factor listed in the results, such as time pressure, for instance, due to their ability of negotiation. Once having a culture established and team members are aware of the importance of non-functional testing and have the quality on top of mind, factors currently being a pain point can be improved.

Priority was largely cited per respondents. Since the traditional waterfall development, literature reports the challenges on getting the priority to non-functional requirements and consecutively testing [21][22][24]. In agile, the priority is set based on value added to the customer [4][28]. This characteristic is highlighted in the results where criticality to business was listed as an aspect which also supports on the determination of the priority of non-functional testing. When analyzing an user story the teams verify which will add value to business according the project or feature being implemented, in case non-functional testing will add value in that moment, it will gain priority during development.

Technical aspects were also cited as factors influencing the non-functional testing. Technical items such as high resource utilization can influence the priority for instances, aligned with quality mindset of anticipate and avoid possible issues in production.

The main observation was that factors are related and influence each other. Priority is can be influenced by team experience and non-functional culture and awareness of its importance. Time pressure and cost can be better analyzed based by experienced team members. Issues as lack of a dedicated environment can get priority and budget once its acquisition importance is very clear for all team member and customers. According to the results, when importance of non-functional testing is not clear to all, factors observed in this paper culminates in no time enough to execute non-functional testing due to the lack of priority and time and, of additional budget (hours) are added, due to the lack of visibility from business and developers that it is important, culminating in possible production issues that can damage customers’ brand or productivity of real users.

2) How do agile teams handle with the identified factors?

Four main practices were reported by participants as practices established to better handle with the influencing factors obtained in the results, as shown in Table 3.

TABLE III. PRACTICES

Factor	Practices
Priority	Discuss non-functional aspects during project inception, sprint planning and user story development.
Culture Awareness	To have team members from different roles (developers, testers, software architects and product owners) reviewing non-functional requirements and testing needs.
Priority Culture Awareness Time Pressure Cost Technical Issues	Clear communication between team members and customers. Including production support teams (DevOps).
Priority Culture Awareness	Work with a quality mindset. Evangelize the team with regards the importance of non-functional requirements and testing.

During the interviews, participants listed key moments where non-functional aspects should be discussed and detailed. Participants mentioned the project inception as the moment where business characteristic should be reviewed. This analysis, according them, help on identifying that the non-functional testing needs supporting to define priority. The same exercise is done during sprint planning. Once reviewing the user story to be developed, senior team members reinforce the need of a deeper review under non-functional aspects related to that user story. As the whole team participates, this exercise helps on the dissemination of the non-functional awareness and culture. Participants reported that a good approach is to work with a “concerned” mindset. When having a risk mindset and an explanation about how impacted business can be, other factors can be minimized such cost and time, because non-functional needs can be better understood.

Participants reported the importance of having representatives from all roles during the agile ceremonials, especially inception and sprint planning. Developers, architects, product owners, business and in case possible,

non-functional experts. They also reported the importance of the presence of senior members during these meetings.

According to participants, the conversation about non-functional requirements and testing needs to continue during the development of the user story and, through discussions on the agile team ceremonies such as stand-up-meetings. They reported that non-functional aspects are related to an evangelizing task, which needs to be done always in order to produce a culture inside the team and company.

Communication was largely listed as a channel to handle with the factors impacting non-functional testing. Participants believe that a good communication around non-functional issues with the features teams aligned with DevOps communication or order to have developers aware of the behavior of the application in production can also help on the identification of non-functional testing needs. They believe that sometimes, developers and testers don't have information enough to elaborate a good testing strategy. The communication with other teams and real users can in their view, to support on identifying where the prioritization needs to focus on.

Also, work with the quality perception is crucial according to the participants. Quality on top of mind during the whole process and post-delivery. Participants also cited code review practice as an instrument that helps in the identification and execution of non-functional testing. All roles working with the quality in mind and awareness of non-functional needs.

VI. CONCLUSIONS

In this paper we presented the results from a study on agile team member's perceptions regarding the testing of non-functional requirements in agile teams. Experience, culture and awareness were emphasized as the most important factors influencing the identification of agile testing needs, execution and resolution of non-functional issues. Factors such as priority, cost and time pressure reported in the literature since the traditional waterfall development [7][12] were reinforced. Technical issues were also reported as factors highly aligned with cost and awareness about what are non-functional aspects and system behavior in production, high response times, for instance.

With regards to priority, culture and awareness, possible solutions are the analysis over non-functional aspects during the project meetings, such as project inception and sprint planning and non-functional reflection during the user story development. The participation of different roles during the meetings and, the experience of the team members can also support in minimizing the issues.

Cost and time pressure can be correlate to priority culture and awareness, since team members have a clear view of the importance to non-functional aspects those factors should be minimized by a good conversation with the customer and prioritization of the tasks. Technical issues could be solved by the engagement of the team with regards a better analysis improved by the experience of team members aligned with the communication with other teams, such as a better DevOps integration. Results may indicate an opportunity for more research over team

experience influencing software quality and non-functional testing.

As a limitation of our work we interviewed a small number of agile team members from the company in our study. The majority number of respondents were performance engineers and no security testers could be interviewed. Therefore, the results reported here could differ from what other teams within the company might have to say when considering their different contexts and culture backgrounds.

We expect these preliminary findings to motivate other researchers and practitioners to take a better look in non-functional testing practices to help non-functional testing to be better defined and executed in agile projects. We understand that our preliminary findings are limited (e.g., low number of participants, single company) and need to be confirmed. We plan to conduct additional interviews with practitioners from other companies and with distinct characteristics to minimize the impact of certain variables such as experience on the job and role played on the responses provided. We also aim to survey a larger population to confirm the results identified in our second round of interviews.

ACKNOWLEDGMENT

This work was supported by the SoS-Agile: Science of Security in Agile Software Development project, funded by the Research Council of Norway under the grant 247678/O70.

We would like to thank the PDTI Program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91) and the participants for their collaboration and time.

REFERENCES

- [1] Ambler, S. W. "Beyond functional requirements on agile projects", *Dr. Dobbs's Journal*, vol. 33–10, 2008, pp. 64–66.
- [2] Baca, D.; Boldt, M.; Carlsson, B.; Jacobsson, A. "A novel security-enhanced agile software development process applied in an industrial setting". In: *Availability, Reliability and Security (ARES)*, 2015 10th International Conference on, 2015, pp. 11–19.
- [3] Bass, L.; Nord, R.; Wood, W.; Zubrow, D. "Risk themes discovered through architecture evaluations". In: *Software Architecture, 2007. WICSA'07. The Working IEEE/IFIP Conference on*, 2007, pp. 1–1.
- [4] Beck, K.; colleagues. "Manifesto for agile software development". Source: <http://www.agilemanifesto.org/>, Nov 2015.
- [5] Bellomo, S.; Ernst, N.; Nord, R. L.; Ozkaya, I. "Evolutionary improvements of cross-cutting concerns: Performance in practice". In: *Software Maintenance and Evolution (ICSME)*, 2014 IEEE International Conference on, 2014, pp. 545–548.
- [6] Bellomo, S.; Gorton, I.; Kazman, R. "Toward agile architecture: Insights from 15 years of atom data", *IEEE Software*, –5, 2015, pp. 38–45.
- [7] Chung, L.; Nixon, B. A.; Yu, E.; Mylopoulos, J. "Non-functional requirements in software engineering". Springer Science Business Media, 2012, vol. 5.
- [8] Cohen, D.; Goldman, N.; Narayanaswamy, K. "Adding performance information to adt interfaces". In: *ACM SIGPLAN Notices*, 1994, pp. 84–93.
- [9] Collins, E. F.; et al. "Software test automation practices in agile development environment: An industry experience report". In: *Proceedings of the 7th International Workshop on Automation of Software Test*, 2012, pp. 57–63.
- [10] Crispin, L.; Gregory, J. "Agile testing: A practical guide for testers and agile teams". Pearson Education, 2009.
- [11] Crispin, L.; Gregory, J. "More Agile Testing: Learning Journeys for the Whole Team". Pearson Education, 2014.

- [12] Glinz, M. "On non-functional requirements". In: Requirements Engineering Conference, 2007. RE'07. 15th IEEE International, 2007, pp. 21–26.
- [13] Graham, D. "Requirements and testing: Seven missing-link myths", *Software, IEEE*, vol. 19–5, 2002, pp. 15–17.
- [14] Herrmann, A.; Paech, B. "Moqare: misuse-oriented quality requirements engineering", *Requirements Engineering*, vol. 13–1, 2008, pp. 73–86.
- [15] Hibshi, H.; Slavin, R.; Niu, J.; Breaux, T. D. "Rethinking security requirements in re research", Technical Report, Tech. Rep. Report CSTR-2014-001, Univ. Texas at San Antonio, 2014.
- [16] Httermann, M. "DevOps for developers". Apress, 2012.
- [17] Jaatun, M. G.; Cruzes, D. S.; Bernsmed, K.; Töndel, I. A.; Röstad, L. "Software security maturity in public organisations". In: *Information Security*, Springer, 2015, pp. 120–138.
- [18] Lawrence, B.; Wiegers, K.; Ebert, C. "The top risk of requirements engineering", *Software, IEEE*, vol. 18–6, 2001, pp. 62–63.
- [19] Martens, N. "The impact of non-functional requirements on project success", 2011.
- [20] Meier, J.; Farre, C.; Bansode, P.; Barber, S.; ea, D. "Performance testing guidance for web applications: patterns & practices". Microsoft press, 2007.
- [21] Myers, G. J.; Corey Sandler, a. T. B. "The Art of Software Testing, Third Edition". Boston, MA: Wiley, 2011, 256p.
- [22] Mylopoulos, J.; Chung, L.; Nixon, B. "Representing and using nonfunctional requirements: A process-oriented approach", *Software Engineering, IEEE Transactions on*, vol. 18–6, 1992, pp. 483–497.
- [23] Paetsch, F.; Eberlein, A.; Maurer, F. "Requirements engineering and agile software development". In: null, 2003, pp. 308.
- [24] Pavlovski, C. J.; Zou, J. "Non-functional requirements in business process modeling". In: *Proceedings of the fifth Asia-Pacific conference on Conceptual Modelling-Volume 79*, 2008, pp. 103–112.
- [25] R. E. Boyatzis, "Transforming qualitative information: thematic analysis and code development". Sage Publications, 1998.
- [26] Singh, P.; Tripathi, A. K. "Treating nfr as first grade for its testability", 2012
- [27] Sommerville, I. "Software Engineering, Ninth Edition". Boston, MA: Pearson, 2010, 792p.
- [28] Talby, D.; Keren, A.; Hazzan, O.; Dubinsky, Y. "Agile software testing in a large-scale project", *Software, IEEE*, vol. 23–4, 2006, pp. 30–37.
- [29] Tappenden, A.; Beatty, P.; Miller, J.; Geras, A.; Smith, M. "Agile security testing of web-based systems via httpunit". In: *Agile Conference, 2005. Proceedings, 2005*, pp. 29–38.
- [30] Wäyrynen, J.; Bodén, M.; Boström, G. "Security engineering and extremeprogramming: An impossible marriage?" In: *Extreme programming and agile methods-XP/Agile Universe 2004*, Springer, 2004, pp. 117–128.
- [31] Yahya, N.; Bakar, A. A.; Sham, N. "Test driven development contribution in universities in producing quality software: A systematic review". In: *Information and Communication Technology for The Muslim World (ICT4M), 2014 The 5th International Conference on, 2014*, pp. 1–6.