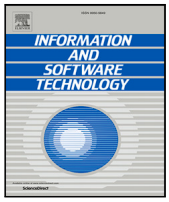
Information and Software Technology 153 (2023) 107067

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



Exploring the challenges in software testing of the 5G system at Nokia: A survey

Szymon Stradowski [a,b], Lech Madeyski [b,*]

[a] Nokia, Szybowcowa 2, Wrocław, 54-206, Dolnoslaskie, Poland
[b] Wrocław University of Science and Technology, Wyb. Wyspianskiego 27, Wrocław, 50-370, Dolnoslaskie, Poland

ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.6945430

MSC:
0000
1111

Keywords:
Software quality assurance
System level testing
Software engineering challenges
Test effort reduction
Efficiency improvement
5G technology

ABSTRACT

Context: The ever-growing size and complexity of industrial software products pose significant quality assurance challenges to engineering researchers and practitioners, despite the constant effort to increase knowledge and improve the processes. 5G technology developed by Nokia is one example of such a grand and highly complex system with improvement potential.
Objective: The following paper provides an overview of the current quality assurance processes used by Nokia to develop the 5G technology and provides insight into the most prominent challenges by an evaluation of perceived importance, urgency, and difficulty to understand the future opportunities.
Method: Nokia mode of operation, briefly introduced in this paper, has been subjected to extensive analysis by a selected group of experienced test-oriented professionals to define the most critical areas of concern. Secondly, the identified problems were evaluated by Nokia gNB system-level test professionals in a dedicated survey.
Results: The questionnaire was completed by 312 out of 2935 (10.63%) possible respondents. The challenges are seen as the most important and urgent: customer scenario testing, performance testing, and competence ramp-up. Challenges seen as the most difficult to solve are low occurrence failures, hidden feature dependencies, and hardware configuration-specific problems.
Conclusions: Our research identified several improvement areas in the quality assurance processes used to develop the 5G technology by determining the most important and urgent problems that at the same time have a low perceived difficulty. Such initiatives are attractive from a business perspective. On the other hand, challenges seen as the most impactful yet difficult may be of interest to the academic research community.

1. Introduction

Many software companies worldwide face enormous challenges in delivering quality products on time and within budget. According to the Standish Group's 2015 CHAOS report [1], only 29% of surveyed projects were completed within the planned estimations. Conversely, as much as 19% failed by being cancelled or not used after completion. Furthermore, the success rate falls dramatically with the project size. The success ratio for large and grand endeavours is only 11% and 6%, respectively. A similar trend is visible when the complexity of the product is taken into consideration, with complex and very complex success rates being 18% and 15% respectively. To make matters worse, the telecommunications industry has the second lowest chance of a favourable outcome, only slightly higher than government initiatives. An excellent example of a grand, complex telecommunication system is the 5G technology developed by Nokia.

Nokia is a Finnish multinational information technology company. It employs approximately 92 thousand people across 130 countries, and most of its business comes from developing cutting-edge, high-tech solutions designed for mobile network providers [2]. A significant part of its operation is committed to developing and improving 5G technology. Accordingly, testing the software solutions based on 5G is also a considerable challenge due to the increased scale and complexity of the process. The number of interfacing components, possible hardware combinations, and the spectrum of used frequency ranges requires a highly sophisticated development approach. Nokia recognises that the current software processes are no longer sufficiently effective and would benefit from scientific methods to improve them.

Software quality assurance is a vital part of the software development process. Not only inadequate testing leads to cost consequences from the customer side, but detecting faults in later stages of the

* Corresponding author.
E-mail address: lech.madeyski@pwr.edu.pl (L. Madeyski).

https://doi.org/10.1016/j.infsof.2022.107067
Received 8 February 2022; Received in revised form 14 August 2022; Accepted 13 September 2022
Available online 18 September 2022

software development life cycle is more expensive than doing so earlier [3,4]. In Nokia, the scale and complexity of the developed product, including aspects such as the number of lines of code, interdependent components, hardware variants, and the needed diversity of test environments, results in a significant volume of defects found at late stages of testing. Therefore, there are substantial opportunities in finding solutions to the faced problems, making the analysis of the challenges at Nokia a valuable area for scientific research.

The paper is organised into six sections. Section 1 is an introduction to the researched topic, highlighting the main goals and contributions. Section 2 consists of an outline of currently used system-level quality assurance processes. Section 3 describes the chosen research approach, posed research questions (RQs), and details of the survey method. Section 4 presents the main questionnaire results overview. Discussion of the obtained results and threats to validity are presented in Section 5. Section 6 contains the final conclusions.

1.1. Related work

We have found several publications in the field of software engineering (SE) on the challenges of testing similarly complex products [5–7], as well as 5G specific challenges [8–10]. Selected articles represent relevant research materials and have been valuable in adequately assessing and critiquing currently used practices for wireless communication technology in Nokia.

- Masuda et al. [5] accurately describe the difficulties in testing highly complex systems utilising the ISO/IEC/IEEE 29119 Software Testing standards [11]. Authors argue that when the number of requirements and functions is vast, it becomes impossible for a human to analyse all of them correctly. Furthermore, the authors propose a method for analysing such complex software by using the knowledge of software architecture and requirements engineering in international standards, called software-test-architecture. The proposed method consists of three core processes: software-test-architecture conceptualisation, software-test-architecture evaluation, and software-test-architecture elaboration. The authors provide a case study for applying the approach to an existing system.
- An autonomous car is a prime example of a complex system that needs to meet exceptionally high quality standards. Ben Abdessalem et al. [6] define this challenge as the difficulty of detecting and managing feature interactions, particularly those that violate system requirements leading to failures. The authors propose a technique to detect feature interaction failures by a search-based test generation algorithm. They explain the design of the algorithm, while its effectiveness is evaluated using two versions of an industrial self-driving system.
- Another software engineering challenge getting increasingly difficult to manage with increasing complexity is regression testing. Following Zhong et al. [7], the regression test selection (RTS) method intends to reduce the number of instances that need to be executed to only those that were affected by code changes. Authors argue that although a large number of RTS techniques have been developed, their adoption in large-scale web services is low. Such methods usually require direct code dependency between tests and code, which is arduous to maintain, or are not sufficiently efficient when applied in large-scale systems.
- Piri et al. [8] provide a compelling description of the technical architecture of the proposed test network for 5G application development and testing. Despite focusing more on the application side rather than the network itself, the study provides meaningful insight into the most significant characteristics of the whole technology like Radio Access Networks (RAN), Internet of Things (IoT), various interfaces, and cloud computing.
- Over-the-Air (OTA) measurement challenges were studied by Qi et al. [9]. The publication contains an overview of the reasons for the increased need for wireless verification of the air interface, anechoic chamber characteristics, chipset measurement possibilities and more. Importantly, the authors also provide an explanation of the reasons for increased OTA measurements needed compared to previous generations.
- Zhang et al. [10] authored an extensive overview of the solutions, opportunities, and challenges behind 5G testing. They discuss the relationships between key technologies and their respective requirements, provide a testing overview distinguishing four main areas of technology, architecture, application and equipment, and explain the challenges in channel modelling and OTA testing design. Furthermore, the authors also identify key challenges and open issues for future research.

Likewise, many publications are based on a similar survey-based approach that we have chosen. They attempt to analyse a particular aspect of software engineering by asking software practitioners to identify the most critical problems or by soliciting an assessment of preidentified difficulties.

- Work published by Garousi and Varma [12] in 2009 is a follow-up to a 2004 survey of test practices among several software organisations in the Canadian province of Alberta. The focus was on eliciting a wide range of software engineering aspects like used programming languages, extent of adoption of different testing stages, level of test automation, utilised tools and techniques, and available training programs. In addition, a significant portion of the analysis was committed to understanding how to manage the overall test process. Interestingly, one of the highlights offered by the authors is that the largest defect rate per thousand lines of code was observed in the telecommunications equipment industry.
- Another survey by Garousi and Zhi [13] was run to obtain a more extensive, nationwide view of the current software engineering practices. As many as 246 respondents answered 34 questions on utilised test practices, test stages, techniques, tools, metrics, management practices, training, level of automation, and interaction with academia. Authors analysed the responses and derived conclusions on the observed rising awareness of testing-related training needs, wider acceptance of the Agile methods, and growing tester-to-developer ratio. However, they also pinpoint a few areas of concern, such as disappointingly low interaction frequency with the research community or manual testing being more popular than automated testing. Arguably, such concerns are still relevant today.
- Begel and Zimmermann [14] published a list of 145 questions elicited in 2012 from a survey conducted among Microsoft engineers. The authors asked their respondents what question would they like to get an answer for from the scientific research community. After synthesising the preliminary answers to a list of 145 questions, they conducted a second survey among a different group of Microsoft engineers to prioritise the identified challenges. As a result, a compelling list of problems relevant for research, industry, and education purposes has been developed. It provides insight into many important issues like increasing practitioners' interest in how users interact with their applications, the effectiveness of product quality criteria, or difficulties in improving collaboration and knowledge sharing between teams.
- In a much more recent survey from 2019, Wang et al. [15] analysed the current state of the practices in software test automation. The maturity of the evaluated approaches is measured in five categories: process maturity, practice maturity, practice correlation, organisational factors, and response variation. Authors conclude that test automation processes that follow the modern software development models of Agile and DevOps tend to reach a higher

level of effectiveness than traditional approaches. Furthermore, the report states that the overall level of test automation varies significantly among organisations since the test automation practices they use are diverse, indicating significant improvements can be achieved by incorporating more mature and accepted procedures.

1.2. Contributions

Following Sjøberg et al. [16], we acknowledge that empirical software engineering should contribute to increasing the knowledge about which methods and solutions are most useful in which circumstances. Our survey aims to elicit insight into the real system-level testing challenges of a very complex product by analysing the perception of facing challenges by industry practitioners.

The most significant contributions of the article are similar to the ones presented by Begel and Zimmermann [14] — providing a prioritised set of software development questions that engineers from a selected company would like to ask the scientific research community. However, we highlight the most critical challenges in the context of agile-based large-scale software powerhouse and emphasise additional difficulties resulting from aspects specific to 5G technology. Our research provides:

- An overview of the primary quality assurance processes used by Nokia in testing wireless telecommunication systems.
- A description of the main challenges faced by Nokia in testing 5G technology.
- Detailed description and results of the conducted survey evaluating and prioritising faced problems.
- An offering of suggestions and recommendations to support further research in the area.
- Importantly, this paper does not describe the processes and best practices used to mitigate related challenges.

Moreover, as stated Basili et al. [17] successful software business requires understanding, continuous improvement, and packaging of experience for further reuse. Received and analysed responses will help Nokia understand where it should focus its efforts and investment while planning future software development practice improvement initiatives to get the most cost-effective results.

2. Nokia mode of operation (MoO)

Since 2007, Nokia focuses its business almost entirely on telecommunication infrastructure operations. The main goal of its Mobile Networks (MN) unit is to be a technology leader in 5G/New Radio (NR) and Single Radio Access Network (RAN) for the combined 2G/3G/4G/5G product. Having a broad technology portfolio allows Nokia to be involved in more than 188 5G/NR commercial engagements, with a total number of 44 live 5G networks at the end of 2020 [2]. To thrive in such a technology-heavy industry, Nokia has spent over €129bn in R&D investment over the past two decades. During this time, the company has evolved its mode of operations, knowledge-base, and used good practices following the official ISO/IEC/IEEE 29119 [11] and ISTQB [18] standards. Furthermore, Nokia is implementing principles of DevOps [19] to closely cooperate with its customers, shorten the feedback loop, and cut down the time to market. Importantly, Nokia also utilises many Lean Six Sigma methodologies Pyzdek [20] to remove operational obstacles, limit waste and inefficiencies, and provide a better response to the needs of the market.

2.1. 5G specifics

5G is the first mobile technology designed for machines as well as people and to enable very high transmission speed, low latency, and reduced error rate. The gNodeB[1] (gNB), which is the main focus of our study, connects the 5G User Equipment (UE) with 5G core using 5G air interface. The air interface, defined by the 3GPP specification, is divided into two frequency bands, FR1 (below 6 GHz) and FR2 (24–54 GHz), each having different propagation characteristics and requiring specific approaches to develop and test [21]. Importantly, advanced techniques such as massive MIMO (Multiple Input Multiple Output, using multiple antennas in the transmitter and receiver) and beamforming (sending signals at particular angles to utilise constructive and destructive interference) are used to achieve strict performance requirements [22], but also add increased complexity to the testing process.

Among many challenges that emerge during the design, development, and testing of 5G technology [10], three require special consideration in the context of our study:

- First, to test the system-level product performance effectively, verification occurs not only in simulators and conducted mode (via physical cabling connection between the antenna and user equipment) but also in real over-the-air (OTA) conditions [9]. Before 5G, OTA was also used to evaluate 2G, 3G, 4G, and User Equipment; however, due to physical characteristics of utilised high-frequency bands having greater propagation loss, previous generation OTA testing needed to be executed on a much smaller scale [22]. OTA measurements require a very high investment in building special anechoic chambers (visible in Fig. 1) to test the base station systems, antennas, and radio functionalities. This aspect requires an increased expenditure on infrastructure and meticulous planning of execution to use this infrastructure effectively.
- Second, Nokia's commercial deals at the end of 2020 included 139 customers and 44 already live 5G networks [2]. Each customer brings their own specific needs and requirements, translating to hundreds of features and thousands of software and hardware configurations. There is no possibility to test all of those combinations exhaustively; therefore, sophisticated heuristics and know-how are required to define the best target coverage on all levels of testing [18].
- The 5G system is comprised of a multitude of features, and each new software release introduces new ones incrementally. Due to the complexity and size of the system, it is extremely hard to predict all interactions on the specification level [6]. If the feature dependencies are not defined deterministically, they can still be discovered during testing. However, considering thousands of possible hardware and software combinations, the residual risk persists. Some may be missed throughout the whole development life cycle and be exposed as late as the large-scale deployment.

Together with the more generic test challenges of large-scale systems described in Section 2.4, the challenges resulting from the complexity of 5G gNB massively add to the difficulty faced by Nokia in its quality assurance effort.

2.2. Continuous development, integration, and testing

Nokia follows the principles of DevOps, extending Agile principles to the entire software delivery pipeline [19]. It aims to reduce the time between requirement generation, developing and committing a code change, and placing the change into production while at the same time ensuring the high quality of the end product. In Nokia, four main phases of the software delivery pipeline can be distinguished:

[1] gNodeB is the "Next Generation Node B" 5G base transceiver station, that is compliant with the 3GPP standard.

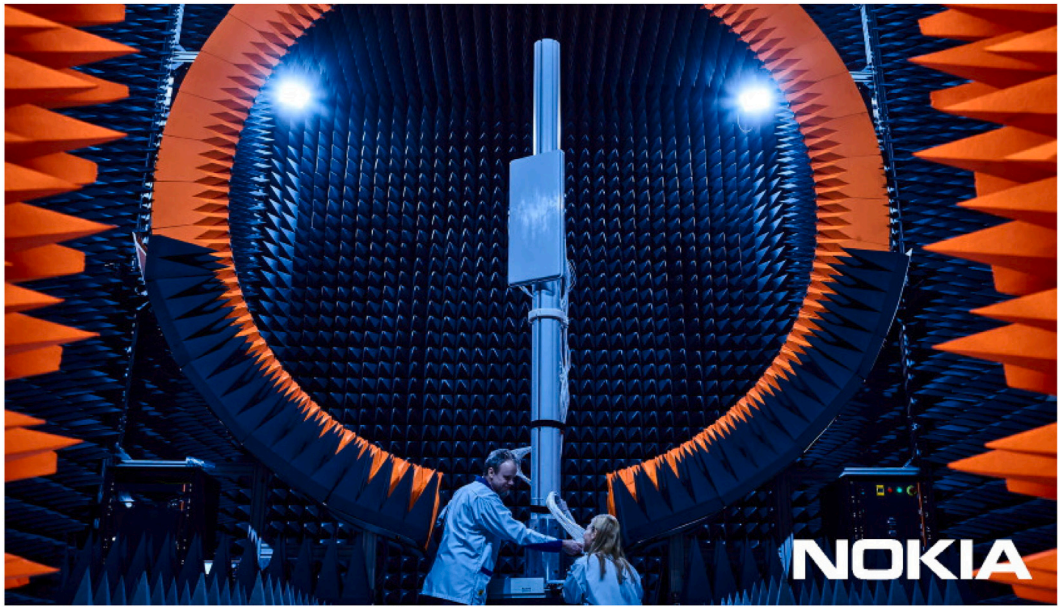


Fig. 1. Nokia's anechoic chamber in Oulu, Finland [2].

- Continuous Development is employed in Nokia to allow thousands of developers to commit their code to the common trunk as frequently as possible with the smallest possible changes, be tested, and released as often as needed. Agile principles are employed on time-box sprints and define the most important items to be worked on first. After the commit, code reviews and basic checks are executed automatically. If passed, the changes flow into the integration phase.
- Continuous Integration means merging new functionalities and hardware as quickly as possible. This approach applies not only to the individual components but also to the system and system-of-systems integration levels and usually is done at the Entity Test-level (ET). A high level of automation is required to deliver the code commits to the trunk, triggering sanity tests and posting the build to a common repository. All the steps need to happen in a dependable and repeatable manner across the whole development cycle.
- Continuous Testing is the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the quality. This mechanism allows the majority of the defects to be found very quickly after the development and integration stages. It requires all test cases that are part of the pipeline to be fully automated and triggered without human interaction. The same requirements apply to failure registration, symptom collection, and logging. Nokia's continuous testing consists of thousands of scenarios of varying complexity that can be triggered depending on the specifics of the committed changes.
- Continuous Delivery is a software development practice where code changes after commit, integration, and testing are automatically prepared for release to production. It enables multiple teams to deliver common software builds in short cycles, ensuring that the software can be reliably released at any time and without manual intervention.

2.3. System-level testing

From the organisational perspective, Nokia MN RAN is focused on the gNB part of the 5G system (see The 3rd Generation Partnership Project [21]) and built around strong program management, agile development units, system test unit, and customer teams. The agile development units create and integrate new functionalities, which then undergo testing on the system level, and afterwards are introduced to the customer networks, all managed under the system program umbrella. Such set-up supports the DevOps mode of operation, allowing fast execution and effective management of scale and complexity. However, it also creates significant challenges for the system-level test teams, such as raising competence requirements, test automation build-up, feature interaction tracking, maintenance creep, and many others. A simplified view of Nokia's Continuous Development, Integration, Testing, and Delivery (CDIT) is visible in Fig. 2 and described below:

- Feature teams focus on new functionalities making sure the testing is robust and adheres to the requirements (FT — Feature Testing).
- Automated complete regression cycle includes tens of thousands of test cases run in a two-week cadence (CRT — Continuous Regression Test).
- Automated daily regression aims to verify the current changes in the code, with a scope that balances the effort needs and coverage effectiveness (CIT — Continuous Integration Test).
- Customer delivery testing is triggered when a delivery candidate software version is selected for release and focuses on customised customer scenarios — SW feature combinations and HW configurations used in real customer network (CDRT — Continuous Delivery Regression Test).

Notably, the gNB system-level machinery has to catch as many defects as possible, both common and easy to detect operability defects, as well as rare and difficult low occurrence or performance issues. Automated test cases can be shifted between the mentioned mechanisms to optimise the effectiveness of the whole process and the cost of operation. However, managing effectiveness and cost is an arduous task, with severe consequences of mistakes — not contained defects found as failures during the live operation of the product.

As the gNB is responsible for establishing and maintaining the connection between the UE and the core network, the majority of the functional tests relate to the air interface. Both the User and Control Planes must adhere to strict 3GPP (see [21]) requirements such as latency or bit rate, but also complex mobility and carrier aggregation scenarios. Testing such functionalities at the early stages of product development can focus on software and hardware configuration of the gNB, or verify only the outgoing transmission characteristics with spectrum analysers. On the system level, however, test scenarios need

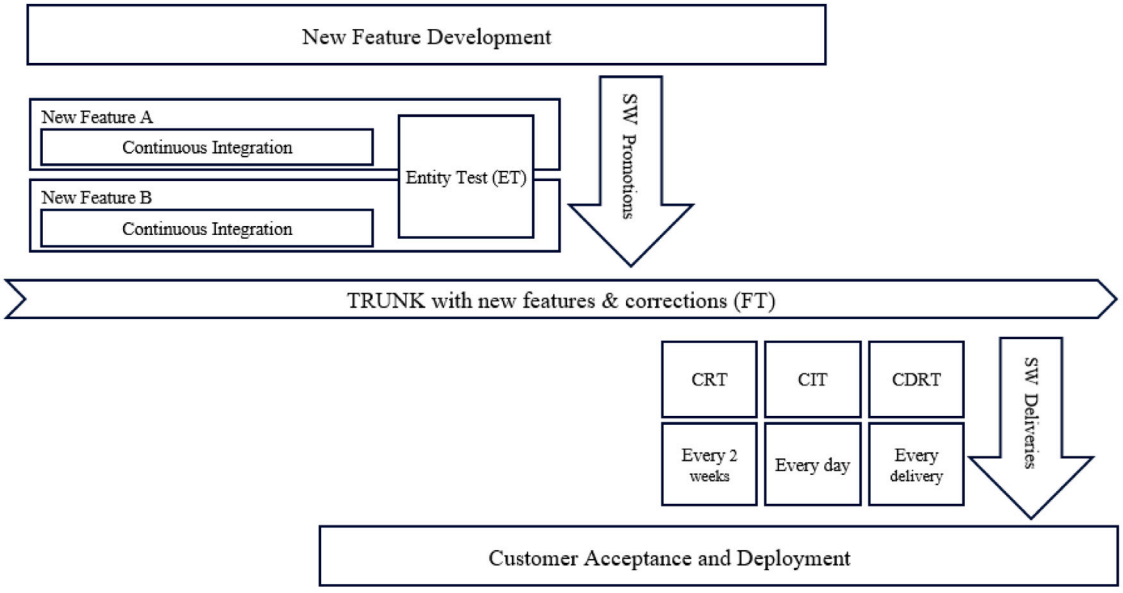


Fig. 2. Nokia CDIT (Continuous Development, Integration, and Testing) concept.

to be verified end-to-end using real UEs, real core, and real OTA interface. Also, certain user stories as max throughput or call stability can be sufficiently tested in a simple lab set-up; however, complex high-speed cell-edge measurements can only be achieved flying a UE attached to a drone circling a set of several gNB, or driving a car with multiple UEs through a dense urban environment. Such variation of test instances significantly increases the operational cost and requires meticulous planning and optimisation.

2.4. Test challenges overview

Apart from 5G specific issues described in Section 2.1, Nokia also needs to overcome generic requirements that arise when testing large scale systems. The company commits significant efforts to understand its challenges and create innovative solutions to mitigate them. The most critical challenges arising from the scale of the system, necessitating further research, are highlighted below.

- One of the main principles of the test theory emphasises the importance of testing early [23,24]. The shorter the time between defect introduction and discovery, the cheaper it is to correct. Therefore, considering the scale of the product, left-shifting not contained faults that could have been found earlier is critical. Systematic management of test scenarios that need to be run in each CRT, CIT, and CDRT cycle phase is one of the main operational targets.
- In highly-complex systems, some defects are near impossible to be found in simulated environments not utilising the whole live environment or the scale of mass-roll-out scenarios. Therefore, Nokia invests heavily in building sophisticated laboratories with hundreds of base stations and OTA chambers. A significant challenge arises in planning what scenarios need to be tested in those environments to utilise their purpose efficiently.
- Due to the size of the product, there are hundreds of smaller teams developing and testing different parts simultaneously. Planning and monitoring such combined effort is intricate and naturally includes work duplication and coverage gaps. Planning of effort distribution is always sub-optimal and carries significant cost-saving potential.
- Continuous Testing brings a constantly growing scope of test cases to be maintained. In the case of Nokia, there are thousands of test cases that can be executed at any moment, but the execution capacity and time are limited. Moreover, the number of feature interactions increases with each new release, continually adding to the maintenance burden.
- One of the primary obstacles in quality assurance is selecting test instances with the highest probability of uncovering errors in millions of possible scenarios. Despite the tremendous effort put into the continuous execution of thousands of different configurations in various environments, many defects are found during customer acceptance and deployment. Eliminating such failures to zero is realistically impossible, but consistently keeping the escaped defects low is one of the company's main business priorities [2].
- Overall, as in any large-scale system, there is a significant amount of findings that could have been found in earlier phase of testing. On a monthly basis, the average ratio of internal findings in each phase is the following: 44% system level, 29% entity level, and 27% other. Out of the 44% of internal findings, 36% were evaluated to be only discoverable during the system testing. With ideal phase containment, the rest could have been found in earlier phases.

3. Research method

The research used in this paper is an empirical study of software engineering challenges set in a real business context. We wanted to connect the research evidence with the domain expertise to improve Nokia's existing software engineering procedures. An way to collect, identify, and assess problems within used practices is to survey a large group of practitioners who face challenges on a daily basis.

Surveys collect quantitative data about the population — subjective as opinions or preferences, and objective as demographics. The purpose of a survey is to increase the understanding of a researched subject by producing statistics not available beforehand [25]. Moreover, the quantitative description of selected aspects and opinions obtained from the studied population allows deriving otherwise unattainable conclusions.

There are several guidelines on how to conduct a survey correctly [25–28]. All agree that this form of research requires a structured approach. First, researchers need to define the research questions, choose a correct sample, select the tools to collect the data, and analyse the findings.

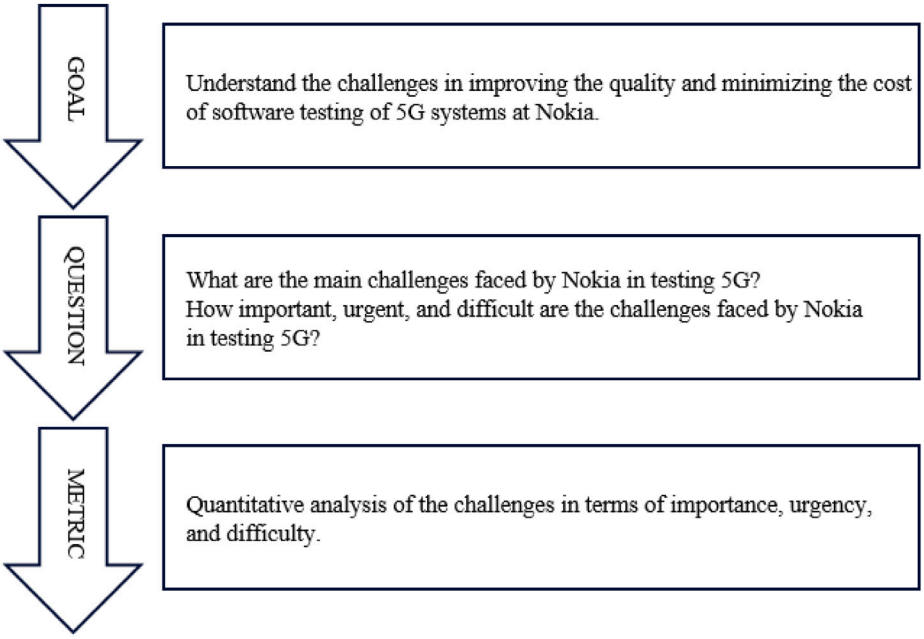


Fig. 3. Used Goal Question Metric (GQM) approach.

3.1. Research questions

First of all, any business effort should specify the objectives for itself. Then, it must trace the data intended to define those objectives and provide a means to measure the characteristics of those objectives. Therefore, we used Goal Question Metric (GQM) approach defined by Basili et al. [29], which helped us organise our efforts and enable adequate measurement.

The correct definition of the goal is critical to the successful application of the GQM approach [30]. Our purpose was to understand the issue of existing difficulties in the process of testing the 5G technology at Nokia from the system level test viewpoint. Since the subject matter experts already have a specific understanding of examined challenges, a comprehensive list aggregating said challenges needed to be created. Secondly, the list was translated into questions evaluating their importance, urgency, and difficulty (available in Section 3.3). Lastly, a measurement framework was built to analyse the practitioners' responses. Our approach is visualised in Fig. 3.

We followed the guidelines proposed by Ciolkowski et al. [31] in raising our four Research Questions (**RQs**), further broken down to survey questions, answers to which we quantitatively analysed in Section 4. The four RQs we defined are:

RQ1. What are the main challenges faced by Nokia in testing the 5G?
RQ2. Which challenges are the most important?
RQ3. Which challenges are the most urgent?
RQ4. Which challenges are the most difficult?

Two solicitation techniques were used to answer our Research Questions. First, a group of six subject matter experts, including quality managers, verification architects, and process owners from the gNB system-level testing unit within Nokia, was selected based on their cross-functional knowledge and experience. They organised a comprehensive list of known areas with the most considerable potential for improvement. The areas were pre-identified pain points of the system-level testing that the organisation encountered during the past years of its operation, resulting from the combination of 5G (see Section 2.1) and general large-scale test challenges (see Section 2.4). The list partially answered our **RQ1**. Second, the list was translated into a questionnaire addressing the challenges to be evaluated. Next, the questions were used in a closed-ended survey, where the target audience could express their opinion on their importance, urgency, and difficulty to answer our **RQ2**, **RQ3**, and **RQ4**. For the sake of completeness, an additional open-ended question was added to solicit any new challenges that might have been missed.

Notably, experts reflected also on four processes within Nokia that are generally regarded as very effective and thus were decided against being part of the survey and subjected to evaluation.

- The general concept of testing in Nokia, as depicted in Fig. 2, has been praised for adherence to the current industry best practices considering large-scale projects. Also, it allows steering between high quality and pace of release; however, it may be difficult to make those decisions.
- Secondly, the interviewed experts confirm the high efficiency of test automation in finding defects on entity and system-level. If given the proper amount of time and expertise in execution, the currently used framework is seen to find a vast amount of defects in each of the phases.
- Nokia also possesses a solid foundation in the very early steps of the Continuous Delivery process. Experts praised the code review process, automatic revert mechanisms, and SW builds promotion machinery.
- Lastly, experts decided that the fault management process currently used in Nokia is well established and effective enough to not require further investigation. However, it is worth pointing out that it was not a unanimous decision; the amount of management and bureaucratic overhead was considered moderately wasteful.

3.2. Survey plan

Significantly, considering that our research is run in a specific business environment, many elements of the survey methodology were imposed or substantially more convenient to execute in a particular way. Thus, in many aspects, our survey compromises between the most accurate method possible and what was achievable in the given circumstances. To guide our efforts, we have used the seven-stage survey research process proposed by Kasunic [27]:

- Identify research objectives.
  Our problem statement implies that the current software quality assurance processes are no longer sufficiently effective when dealing with testing the 5G technology. Thus, our GQM objective is to explore the faced challenges in software testing of the 5G system at Nokia, with the aim of identifying opportunities to improve the quality and minimise the cost.
- Identify and characterise the target audience.
  The survey's target population is the whole 5G system-test unit. Since the organisation is large, individuals work on different parts of the system, have different roles and priorities, and may vary in experience and competence. However, they share a common goal of assuring the system's quality and understanding their main challenges in achieving this goal. As part of the same organisation, they use the same terminology and are highly willing to influence its mode of operations.
- Design sampling plan.
  The target population could be precisely enumerated and consisted of 2935 people in globally distributed localisations. Due to the internal code of conduct, the responses must be fully anonymous, thus ensuring proper probabilistic representation of the whole audience is limited. Consequently, we used non-probability self-selection sampling based on purpose and convenience. Using non-probability sampling imposes several limitations on the derived conclusions [32]. Further consideration of the selected sampling approach is provided in Section 4.
- Design and write questionnaire.
  To support reproducibility of our research [33], the questionnaire is described in detail in Section 3.3 and presented in its original form in Appendix A (while raw data are available in Appendix B). A selected group of experts, in interaction with the authors, created an initial list of 17 questions derived from the main challenges identified by the system-level test organisation. The stated questions were used in a closed-ended survey. The target audience evaluated their importance, urgency, and difficulty in a five-point Likert item [27,34]. For completeness, we also added an open-ended question to solicit new challenges that were missing from the main list. Lastly, two demographic questions were added on the respondent's role and experience.
- Pilot test questionnaire.
  We used pre-testing to assess the reliability and validity of the used framework. We piloted the questionnaire among the six experts that were involved in the development of the initial list of 17 challenges (see Section 3.1). Consequently, several minor modifications were made to the wording and sequencing based on the obtained feedback. Pre-test answers were not considered in the final results as they only aimed to evaluate the validity of the survey. Additional details are available in Section 3.4.
- Distribute the questionnaire.
  The survey was run with online tooling supported by the company, and the invitation was distributed by email from the unit head to his whole organisation. As defined by the sampling plan, there was no selection process; thus, the questionnaire was distributed to the whole target population. Additional details on the execution are available in Section 3.5.
- Analyse results.
  We used the obtained data to create ordered lists in terms of importance, urgency, and difficulty. Open-ended answers were gathered and sorted, and demographic information allowed further insight and identified important response patterns. Main results are available in Section 4 and raw data in Appendix B to support independent validation and further analyses of the results of our research [33].

3.3. Survey questions

Following Begel and Zimmermann [14], we intended for the survey questions to reflect what a quality assurance practitioner would ask the research community to provide a hypothetical yet optimal answer to. The designed list of challenges was posed in terms of survey questions to be evaluated by the respondents in a close-ended manner. The list was not in any way prioritised and only sequenced in a logical way.

Q1. How **Important/Urgent / Difficult** it is to focus on **corner-case testing** to ensure high quality?
Q2. How **Important/Urgent / Difficult** it is to focus on **low occurrence failures** to ensure high quality?
Q3. How **Important/Urgent / Difficult** it is to focus on **performance testing** to ensure high quality?
Q4. How **Important/Urgent / Difficult** it is to focus on **customer scenario testing** to ensure high quality?
Q5. How **Important/Urgent / Difficult** it is to accurately identify and test **hidden feature dependencies**?
Q6. How **Important/Urgent / Difficult** it is to effectively plan **OTA test scope** to catch OTA-specific defects?
Q7. How **Important/Urgent / Difficult** it is to effectively catch **HW configuration-specific problems** out of thousands of possible HW configurations?
Q8. How **Important/Urgent / Difficult** it is to effectively design **exploratory testing** to improve quality with no diminishing returns?
Q9. How **Important/Urgent / Difficult** it is to define the optimal coverage in **maintenance testing** to ensure quality?
Q10. How **Important/Urgent / Difficult** it is to establish the **useful lifetime of a test scenario**?
Q11. How **Important/Urgent / Difficult** it is to mitigate **regression scope increase** not to endanger quality?
Q12. How **Important/Urgent / Difficult** it is to find **areas of increased risk** (defect prone) to be tested with more focus?
Q13. How **Important/Urgent / Difficult** it is to effectively **balance between CRT, CIT, and CDRT** test coverage?
Q14. How **Important/Urgent / Difficult** it is to build effective **defect prediction models**?
Q15. How **Important/Urgent / Difficult** it is to secure proper **competence ramp up** of test engineers?
Q16. How **Important/Urgent / Difficult** it is to accurately **measure test effectiveness**?
Q17. How **Important/Urgent / Difficult** it is to manage **duplication of effort** between test teams?

The target audience evaluated each challenge as an ordinal five-point Likert item from one (very low) to five (very high), usually used to measure attitudes and preferences [34]. Accordingly, created items can serve as a Likert scale measuring the general attitude towards testing challenges in 5G technology in Nokia. Also, we allowed an additional 'I don't know' response that is assigned the scale value of zero [27]. Each issue was evaluated in three categories as shown in Table 1.

- **Importance:** evaluates how much impact solving the challenges would have on process effectiveness. Low importance means the improvement would barely influence the quality or cost of software testing in Nokia. Accordingly, the high importance suggests the change would affect the quality and cost in a significant way.
- **Urgency:** evaluates how fast the issue should be addressed, and due to any reason, it should be solved prior to other problems. Together, importance and urgency can be used as a simple prioritisation matrix to decide the relative impact of the examined challenges [20].
- **Difficulty:** estimates how complex and costly solving the issue would be. The perceived difficulty is treated as a supporting metric attempting to get preliminary input on low-hanging fruit and decisions on further improvement opportunities [20].

Table 1
Exemplary question.

Q14. How **Important/Urgent/Difficult** it is to build effective **defect prediction models**?						
	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

In building the questionnaire, we utilised the lessons learned from Ciolkowski et al. [31]. We provided a concise introduction describing the purpose and continued with the first questions that were not threatening and clearly related to the announced intentions of the survey. Furthermore, we ensured that we used language appropriate for the intended respondents, avoided possibly ambiguous terms, colloquialisms, used correct grammar and spelling. We also made sure that each question expressed only one concept and was kept concise but complete [25]. We eliminated all proposed questions that were seen as too narrow to ensure that potential respondents possessed sufficient knowledge to answer them.

The 18th question in the survey was open-ended. It aimed to identify additional areas that were either missed by the subject matter experts in the initial analysis or decided to be of less importance. It served as an additional source of input to our '**RQ1.** What are the challenges faced by Nokia in testing the 5G technology?'.

Q18. What **other challenges** do You see in System Level Testing??

Lastly, we asked two background questions to collect demographic information on tenure and role. We decided to put those questions at the end of the survey in order not to deter potential respondents and we added an 'I don't know' answer [27].

Q19. How long do You work in the Software Engineering field?
Q20. What is Your **current role?**

3.4. Survey pre-test details

It is essential to have the small-scale pilot studies carried out by using the same artefacts and procedures, including invitation, explanation, questionnaire, tooling, and format, exactly as during the main execution [28,32]. Then, the pre-test offers the best chance for unexpected issues and difficulties to be discovered and fixed before the final version is published.

Before our survey was executed on a full scale, the original group of experts who formulated the questions provided their feedback in pre-test, resulting in several modifications related to the wording, sequencing, and visual aspects. Their answers were not taken into consideration in the final results but only aimed to evaluate the validity of the survey [27].

3.5. Survey execution details

The survey was designed and hosted on Microsoft's online survey service called MS Forms[2] and was open from January 17th to January 31st, 2022. The system-test unit head sent the invitation to a dedicated email distribution list, including the whole target population of all practitioners in all roles directly contributing to system-level testing. As a result, 2935 of Nokia's technical and management staff in China, Germany, Finland, France, India, Poland, Romania, and the United States, received our request.

[2] https://www.microsoft.com/en-us/microsoft-365/online-surveys-polls-quizzes

It is important to note that the heterogeneity of the population was analysed based on the context of our research [28]. Despite significant differences in many aspects like years of experience, area of expertise, role, or intermediate priorities, the whole population shares the same business goal and works according to the same processes. This common factor enables treating the entire group as homogeneous in terms of the research goal — providing insight into the facing challenges.

Conversely, based on our experience, conducting a survey in a business environment can be significantly less troublesome than other circumstances described in the literature [25]. For example, Nokia uses a dedicated function within the People Services organisation, specialising in eliciting data from a larger group of employees and supporting survey design and execution. They use tested practices, know-how, and sophisticated tools to mitigate risks and obstacles usually difficult to handle in a less structured ecosystem. Secondly, employees tend to be highly engaged in the developments inside the company and readily share their opinions. Taking part in surveys can be a meaningful way of influencing the company on both strategic and operational levels. The business environment also provides more convenient opportunities for executing follow-up plans and sending reminders to participants. On the other hand, corporate policies and personal data protection rules may significantly narrow the options of analysing the target population, for example, using selected sampling methods. In our case, we could not create a probabilistic sample without compromising personal data, posing a risk to the mandatory anonymity of the responses. This is an example of how challenging it may sometimes be to apply golden-standard academic approaches in business context [35].

4. Survey results

As non-probability self-selection sampling is the most viable option to reach a broader target audience within our company, there are several consequences to be mindful of [27]. A non-probability sampling utilises human judgement in selecting respondents and does not ensure that the sample is representative of the whole population. Therefore, there is no theoretical basis for estimating population characteristics. The findings apply to only those who respond and cannot be inferred to the whole population (all system-level test practitioners in Nokia). Although it would be incorrect to generalise the results we obtained for the entire company, we believe that meaningful conclusions can still be derived [27].

Secondly, the survey used self-selection, as members of the target audience chose to respond or ignore the request. Such an approach is prone to introducing bias and can be misleading if not interpreted correctly [27]. The survey invitation was not mandatory to create a census (i.e., a sample that includes all individuals in the population), and our respondents were predominantly active and highly-engaged employees. In contrast, a significant group of engineers who are usually unwilling to participate in such initiatives and focus primarily on technical aspects is underrepresented. Conversely, the more inclined to answer group can have a deeper understanding and a broader perspective on the overall situation in their organisations, well corresponding with the high-level nature of our Research Questions. Importantly, as the survey was not mandatory, probability sampling would also include similar bias. Even though a random representation of the group would be selected, only persons with specific characteristics could be more likely to respond.

During the 15 days when the survey was open, we received 312 responses (10.63% response rate) which were afterwards checked for consistency and completeness (see [25]). To support replication of our research [33], we provide the exact survey instructions and raw data of a complete list of questions with received answers in Appendices A and B.

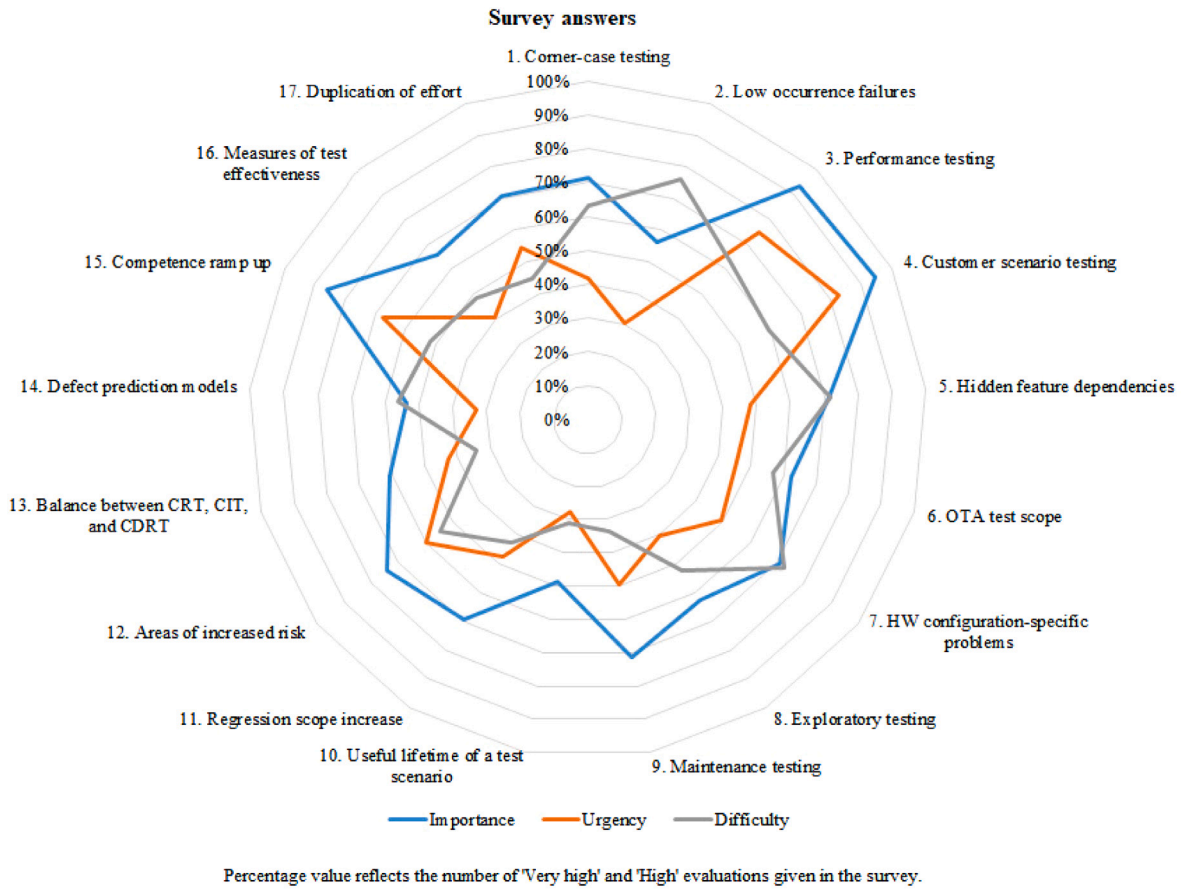


Fig. 4. Spider chart of received evaluations.

Table 2
Survey answers.

Q1–Q17. Challenge evaluations (percentage of 'Very high' and 'High' answers.)			
Challenge	Importance	Urgency	Difficulty
1. Corner-case testing	71%	42%	63%
2. Low occurrence failures	56%	30%	76%
3. Performance testing	93%	75%	63%
4. Customer scenario testing	95%	83%	60%
5. Hidden feature dependencies	71%	48%	72%
6. OTA test scope	62%	46%	57%
7. HW configuration-specific problems	71%	49%	73%
8. Exploratory testing	63%	40%	53%
9. Maintenance testing	71%	50%	34%
10. Useful lifetime of a test scenario	49%	28%	31%
11. Regression scope increase	69%	47%	43%
12. Areas of increased risk	74%	60%	54%
13. Balance CRT, CIT, and CDRT	61%	43%	34%
14. Defect prediction models	54%	33%	56%
15. Competence ramp up	86%	68%	52%
16. Measures of test effectiveness	66%	41%	49%
17. Duplication of effort	71%	54%	45%

4.1. Results: challenges evaluation

We used simple statistical analysis methods for analysing data and drawing our conclusions [28]. Single items were measured by calculating the distribution of different responses Table 2. We also depicted the received evaluations on a dedicated spider chart in Fig. 4 and funnel plots in Appendix C. Most importantly, our findings rely on the percentage of respondents who evaluated the given challenge as 'Very high' and 'High', excluding the 'I don't know' answers, following guidelines by Kitchenham and Pfleeger [25].

Result highlights:

- The most important challenges are related to '4. Customer scenario testing' (95%), '3. Performance testing' (93%), and '15. Competence ramp up' (86%). Notably, the difference to the next highest category is more than 10%, emphasising significant disparity.
- The most urgent challenges are '4. Customer scenario testing' (83%), '3. Performance testing' (75%), and '15. Competence ramp up' (68%). They are the same as the most important challenges, suggesting a correlation between both categories.
- The most difficult challenges are '2. Low occurrence failures' (76%), '7. HW configuration-specific problems' (73%), and '5. Hidden feature dependencies' (72%).
- The aggregated number of 'Very high' and 'High' evaluations of importance is substantial among all challenges showing the respondents believe significant improvements are necessary to improve the overall effectiveness.
- The least important and urgent challenges are '2. Low occurrence failures' (8% and 23%), and '10. Useful lifetime of a test scenario' (7% and 18%), by the percentage of 'Low' and 'Very low' answers.
- The least difficult challenge is '9. Maintenance testing' (18%), by percentage of 'Low' and 'Very low' answers.
- The challenge causing the highest number of 'I don't know' answers was '6. OTA test scope' (23%, 24%, and 25%). It was the highest in all three categories alike, showing limited familiarity with the subject among respondents.
- Interestingly, the second item with most 'I don't know' answers is '14. Defect prediction models' (19%, 21%, and 25%) demonstrating low usage of such solutions in the organisation.

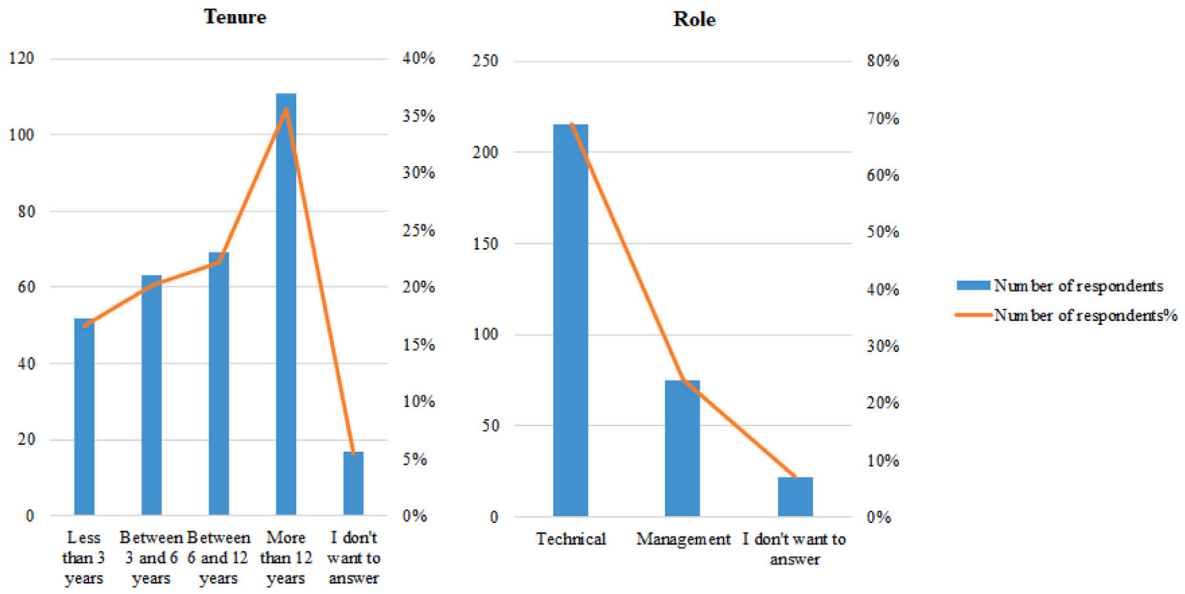


Fig. 5. Charts of working experience and roles of the survey respondents.

Table 3
Open-ended responses.

Q18. What other challenges do You see in System Level Testing?			
Challenge	Count	Challenge	Count
Deficient test infrastructure	25	Product specific issues	4
Time pressure	24	Reporting/management overhead	4
Competence ramp up (Q15)	12	UE issues with external vendor	4
Missing specification	11	Hidden feature dependencies (Q5)	3
Insufficient SW quality	11	Conflicting program priorities	3
Difficult troubleshooting	10	Regression scope increase (Q11)	3
Fault handling process	10	Organisational silos	3
ET phase containment	9	Low occurrence failures (Q2)	3
Inadequate planning	8	Low test automation efficiency	3
Duplication of effort (Q17)	6	Low manual test efficiency	3
Missing big picture	5	HW configuration problems (Q7)	2
Customer scenario testing (Q4)	5	Exploratory testing (Q8)	2
Comment not relevant	*5*	Low motivation	1
Mode of Operations	4	Individuals breaking the process	1

4.2. Results: missing challenges

Table 3 shows the cleaned and aggregated overview of the answers received for the open-ended question, soliciting additional challenges that might have been missed while creating the main list. As many as 127 respondents (41%) offered a non-null response; however, five responses (marked 'NA') we considered as non-relevant. After analysis, we identified 179 individual challenges grouped into 27 unique categories. Eight of those categories were already covered by the original challenges defined prior to the survey (more than 20% of all provided answers). Among the new challenges, the most frequent were: time pressure, deficient test infrastructure, missing specification, insufficient SW quality, difficult troubleshooting, and inadequate fault handling process. They constitute over 50% of all free-text answers.

4.3. Results: demographics

Demographic analysis shows that a typical respondent serves a technical role and has a substantial working experience in software engineering. Such a conclusion comes from the abundance of respondents with large working experience in software engineering (e.g., 36% with more than 12 years, while 78% with at least three years, see Fig. 5). At the same time a vast majority of the respondents (69%) serve a technical role vs 24% who identify as management (only 7% of respondents preferred not to answer the question, see Fig. 5). Interestingly, compared with the internal data of the company, such a ratio indicates that management roles are over-sampled and seemingly more inclined to participate in the survey.

5. Discussion of results

Synthesising both qualitative and quantitative data described in Section 4 provided insight into the challenges in software testing of 5G system at Nokia. In this section, we address our research questions, discuss the interpretation of the results, provide future recommendations, and define threats to validity.

5.1. Results interpretation

Our QGM goal was to understand the issue of existing difficulties in testing the 5G technology at Nokia, which was satisfied by analysing the results of our survey. The key findings show that the most important and urgent challenges were of the technical nature: '4. Customer scenario testing', '3. Performance testing', and '15. Competence ramp up', respectively. More process-oriented difficulties were evaluated as less significant. Additionally, three different challenges were evaluated most highly in terms of the perceived difficulty: low occurrence failures, hidden feature dependencies, and HW configuration-specific problems. As a consequence of the findings, further research efforts can be started based on empirical evidence that the studied problems are relevant and already have a preliminary indication of the expected difficulty level.

Secondly, the results were provided to the company's system-level test organisation to broaden the understanding of priorities for any further improvement initiatives. Results show which challenges are the most important and which should be solved with more urgency than others; thus, improving them would be most beneficial for the

company [20]. Importantly, valuable results could be obtained by seeking important and urgent challenges but with low difficulty. An excellent example would be '15. Competence ramp up'. From the business perspective low-effort and high-return activities to improve product quality should be very attractive [30].

Following Kasunic [27], we compared our results to those obtained in previous studies. We decided to validate our list with a much larger and solicited by a dedicated survey list by Begel and Zimmermann [14] run among employees of Microsoft.

- The list of challenges from Begel and Zimmermann [14] is more extensive, containing as many as 145 challenges in all aspects of the software development life-cycle and not only system-level testing. The study evaluates several categories that go beyond our scope, such as bug measurements teams and collaboration, services, development practices and processes, whereas ours focuses only on system-level testing. However, our study contains five challenges not discussed in Begel and Zimmermann [14]: '5. Hidden feature dependencies', '6. OTA test scope', '7. HW configuration-specific problems', '8. Exploratory testing', and '14. Defect prediction models'.
- Notably, both results show a very high interest in customer perception and focus, revealing the most crucial challenge to be related to customer scenario testing. Our '4. Customer scenario testing' had 95% importance evaluation and in Microsoft's study 'Q27 How do users typically use my application?' had 99.2% as worthwhile, with 'Q18 What parts of a software product are most used and/or loved by customers?' a close second.
- Our study shows the very high importance of performance testing, which may be indicative of 5G technology as one of the market differentiators. In Begel and Zimmermann [14], there were a few questions on product performance; however, they were evaluated much more neutral.
- Similarly, the evaluations in Nokia highlight the need for further '15. Competence ramp up' as it was the third most important issue to be addressed. In contrast, the other study discusses more the engineers' overall productivity — the best way to learn a new area, attributes of high-performing engineers, or knowledge sharing, with importance ranging visibly lower than in Nokia.
- Interestingly, a strictly quality-related question 'Q50 How effective are the quality gates we run at check-in' was ranked as third most important, whereas in our study, the corresponding '16. Measures of test effectiveness' had an average evaluation. We believe this is the result of Nokia having a straightforward gating system allowing precise quality evaluation, and the study in Microsoft elicited answers from engineers working on many different projects potentially having different gating systems.
- This comparison indicates that the insight gained by such surveys tends to be specific to the company characteristics, as each company has unique strengths, weaknesses, and market demands. It could be very beneficial for similar studies to be completed in other companies to benchmark and compare universal large-scale software testing challenges across different industries within software engineering.

5.2. Implications

The study's main findings regarding particular challenges resulting from large-scale of the tested system are briefly recapitulated below, along with their related implications:

- The results for '2. Low occurrence failures' indicate that this is a low importance and very low urgency challenge. At the same time, it is considered the most difficult to solve. As defects of non-permanent and semi-random nature require multi-repetitive testing to catch, the lower the occurrence rate, the more repetitions of a particular scenario needs to be run. Thus, such testing should primarily occur early in the cycle, utilising simulators and minimising high-cost laboratory environment usage [18]. Nevertheless, the challenge of catching low occurrence faults also results from the difficulty of accurately balancing the number of repetitions of each test and the time consumed. Furthermore, they also require robust logging mechanisms and suitable fault management processes to fix and verify accordingly. In a system on chip (SoC) architecture, such faults are caused not only by in-house software but also by other components adding to the overall complexity.
- '4. Customer scenario testing' described in Section 2.3, was evaluated as 'Very high' in importance by 71% of all respondents (221/312 vide Table B.1 in Appendix B), making it the most crucial problem to solve within the system-level testing organisation in Nokia. The root of the problem lies with the enormous number of possible functional and non-functional scenarios characteristic of a live network and the difficulty of recreating them in artificial laboratory environments. Moreover, it imposes high requirements on effective communication with customers to truly understand their needs. This concept is emphasised heavily by DevOps [19], the introduction of which is planned to be continued and expanded in the company.
- The implications for '5. Hidden feature dependencies' are very similar to the ones described by Ben Abdessalem et al. [6]. Nokia test professionals see feature dependency problems as relatively medium importance and urgency but acknowledge a high degree of difficulty in solving the issue. Such evaluation is mainly due to the inability to reliably connect hundreds of complex functionalities to the overall system behaviour and performance and implies the need for further research in the area. Releasing new software builds as frequently as possible, according to the Continuous Delivery concept, requires constant updates of architecture, requirements, and dependencies, not only for the current release, but also for past and future. This is especially visible in the number of functionalities described by the 3GPP standards [21].
- Results indicate that '15. Competence ramp up' continues to be one of the most significant potential improvement areas, showing very high importance and urgency, at the same time being considered less difficult. Together with studies by Garousi and Zhi [13], our survey results confirm that competence buildup through training programs and talent retention continues to be one of the most critical assets for a high-tech company. Furthermore, the issue of insufficient competence is, to a large extent, due to the size of the product and the difficulty in understanding the multitude of requirements in a large-scale system, but since system-level testing is predominantly black-box [18], it does not necessarily influence testing quality.

Secondly, main findings we see as specific to the 5G gNB testing are summarised below:

- '3. Performance testing' had the second most 'Very high' and 'High' importance and urgency evaluations out of all challenges. The result illustrates how critical system performance is to the whole 5G product, where high-speed data transfer is one of the fundamental requirements, and the majority of features contribute towards this characteristic [21]. Therefore, it is one of the two highest pay-off improvement areas and justifies further investment in test infrastructure, ensuring high quality in this critical aspect of the wireless telecommunication system.
- The '6. OTA test scope' challenge has a high degree of uncertainty in the results as more than 20% of all respondents answered 'I don't know' to at least one evaluation category (importance, urgency, and difficulty). Together with several open-text answers we received highlighting this difficulty, it may show how formidable is over-the-air testing and how difficult to operate infrastructure it requires. Most importantly, it also shows how

much more competence and experience is needed among test practitioners to improve this quality assurance aspect, which is notably specific to 5G technology (as also studied by Qi et al. [9]).
- Finally, the evaluations of perceived challenges mostly seem to highlight technical aspects, and process-related considerations do not fall into any extremity. Significantly, challenges related to exploratory testing, maintenance and regression scope, or balancing between CRT, CIT, CDRT are not frequently seen as of 'Very high' and 'High' importance or urgency. This specific observation leads us to believe that no major process changes are necessary, apart from a continuous improvement effort [20].

Importantly, our overarching research effort is committed to improving the quality and minimising the cost of software testing of 5G system at Nokia. Described implications resulting from challenge evaluation show that current software processes within Nokia can be further improved; therefore, our research on the implications will be continued and explained in future articles.

5.3. Further recommendations

We are also aware that additional findings can be obtained by further analysis of obtained data. For example:

- Calculating the degree of consensus and dissensus in evaluations between Management and Technical roles.
- Post-hoc analysis of differences in responses between different groups [20].
- Determining the correlation coefficient between importance and urgency categories [28].
- Building priority matrices to categorise the most important and urgent challenges [20].
- Detailed review of individual free-text answers, discussion of the answers with subject matter experts, and possible follow-up studies.

Above considerations are outside of the scope of this study and were recommended to the company management to be continued during a series of dedicated results read-outs. However, our research will be also continued with specific topics chosen for further long-term improvement effort and described in future publications.

5.4. Threats to validity

In order to improve the validity of our research, we addressed the most common categories of threats from Wohlin et al. [36],Zhou et al. [37] and, following Feldt and Magazinius [38], designed mitigation actions addressing those specific threats before, during, and after the survey execution.

- **Construct validity:** No analysis to prove that the observed differences are statistically significant was performed, as simple quantitative differences were sufficient to identify the answers to our RQs. Secondly, we ensured that the respondents could understand the question easily and that we used proper question-wording. As described in Section 3.4, we performed a proper pre-test and used official company nomenclature. However, we saw a significant threat in the willingness of participants to provide the requested information and not receiving a sufficient number of answers, limiting gained understanding and possible implications. The survey was scheduled for ten working days between the 17th and 31st of January 2022. According to the follow-up plan, the first reminder message was sent after seven days and the second after 12 days to increase the response rate. The remainders helped increase the response rate from 7.53% at the end of day seven to 10.63% at the end of day 15, which we see as satisfactory.
- **Internal validity:** Although a significant effort was put into building a comprehensive list of challenges faced in testing 5G gNB described in Section 3, essential aspects still might have been missed. To mitigate this risk, we have validated our list with a much larger and solicited a dedicated survey list by Begel and Zimmermann [14]. We did not find any significant discrepancies that needed to be addressed. Also, we acknowledge that the identified challenges might have been influenced by temporary problems Nokia faced in specific areas or by currently emphasised management priorities. Therefore, we recommended repeating the survey in the company on an annual basis to build a consistent baseline.
- **External validity:** The difficulties analysed in this paper result from the combination of 5G specifics and a large-scale software system. We do not claim that the results can be generalised to the whole field of 5G testing or large scale systems. However, we believe particular conclusions apply to many circumstances but must be carefully considered within their respective contexts. Each company has its own strengths and weaknesses in terms of created processes, and even the most significant challenge in one can be handled with ease in another (see Section 5.1). Therefore, although we believe that the same challenges are faced by any company in the wireless telecommunication industry, based on the operational efficiency of the used test process, challenges may and will be perceived differently.
- **Conclusion Validity:** A significant threat to our conclusions is related to using non-probabilistic convenience sampling, introducing vulnerabilities to biases. We highlight that our results cannot be generalised to the whole population as a consequence. Despite the limitations imposed by non-probability sampling, we believe the obtained results are valuable and satisfy the goal of our Nokia-focused study. Secondly, the degree to which the conclusions are credible is high due to the number of received answers. Nevertheless, the business context of the research requires constant monitoring, and before investments are made to improve any of the challenges, standard profitability and feasibility studies should be completed.

6. Conclusions

Our research aims to explore the main challenges Nokia company faces in gNB system-level testing of 5G technology. The central part of the identification was done by a group of cross-domain experts based on their experience. Secondly, a survey conducted within the system-level test organisation requested the target audience to assess the identified challenges in terms of importance, urgency, and difficulty. Obtained results satisfy our QGM goals of identifying and evaluating the main challenges faced by Nokia. The most important and urgent challenges are related to customer scenario testing, performance testing, and competence ramp up. Problems seen as the most difficult to solve are low occurrence failures, hidden feature dependencies, and HW configuration-specific problems.

Furthermore, the results show that software practitioners in Nokia see opportunities for improvements that can further increase the product quality and minimise software testing costs. There seems to be considerable interest among practitioners to benefit from academic research [39], and there are still visible discrepancies between industry challenges and applicability of solutions proposed by academia [35]. We believe our study can contribute to bridging this gap by offering an overview of the main challenges software engineering practitioners face in a real business context of 5G technology. Each of the highlighted challenges could be a subject of dedicated study customised to be applicable for problem-specific and industry-specific issues faced by the company, similarly as described by Kasoju et al. [40]. Specifically, we did not provide a broader description of the processes and best practices used to mitigate described challenges. Our effort will be continued by selecting and addressing a subset of the analysed challenges in future research. We believe that continuing this study is essential for Nokia as identified difficulties can be expected to become even more complex in 6G, the emerging next standard for wireless communications technology [41].

CRediT authorship contribution statement

Szymon Stradowski: Data curation, Methodology, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Lech Madeyski:** Conceptualization, Funding acquisition, Methodology, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Szymon Stradowski reports financial support was provided by Polish Ministry of Science and Higher Education. Szymon Stradowski reports a relationship with Nokia Corporation that includes: employment, equity or stocks, and non-financial support. As noted in the acknowledgement at the end of the manuscript, research has been conducted in partnership with Nokia Corporation, where the corresponding author is employed.

Data availability

Original forms and responses are available in Supplementary Material: https://doi.org/10.5281/zenodo.6945430.

Acknowledgements

This research was financed by the Polish Ministry of Education and Science 'Implementation Doctorate' program (ID: DWD/5/0178/2021).

Appendix A. Survey form

See Table A.1.
Original form are available in Supplementary Material: `SurveyForm.pdf`

Table A.1
Exact form of the questionnaire.

Challenges in System Level Testing

The purpose of the survey is to understand better the challenges in improving the quality and minimising the cost of System-Level Testing of 5G system at Nokia.

The survey consists of 3 parts and should not take more than 10 min to finish.
Questions 1 to 17 ask to evaluate the **'Importance', 'Urgency', and 'Difficulty'** of the presented challenge.
Question 18 asks about any areas that might have been **missed.**
Questions 19 and 20 ask about **experience and role.**

Importance evaluates how much impact would solving the challenge have on process effectiveness.
Urgency evaluates how fast the issue should be addressed.
Difficulty estimates how complex and costly would solving the issue be.

Note: the questionnaire is for research purposes only and is fully anonymous.

Q1. How **Important/Urgent/Difficult** it is to focus on **corner-case testing** to ensure high quality?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q2. How **Important/Urgent/Difficult** it is to focus on **low occurrence failures** to ensure high quality?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q3. How **Important/Urgent/Difficult** it is to focus on performance testing to ensure high quality?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q4. How **Important/Urgent/Difficult** it is to focus on **customer scenario testing** to ensure high quality?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q5. How **Important/Urgent/Difficult** it is to accurately identify and test **hidden feature dependencies**?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q6. How **Important/Urgent/Difficult** it is to effectively plan OTA test scope to catch OTA-specific defects?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q7. How **Important/Urgent/Difficult** it is to effectively catch HW configuration-specific problems out of thousands of possible HW configurations?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

(*continued on next page*)

Table A.1 (*continued*).

Q8. How **Important/Urgent/Difficult** it is to effectively design **exploratory testing** to improve quality with no diminishing returns?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q9. How **Important/Urgent/Difficult** it is to define the optimal coverage in **maintenance testing** to ensure high quality?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q10. How **Important/Urgent/Difficult** it is to establish the **useful lifetime of a test scenario**?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q11. How **Important/Urgent/Difficult** it is to mitigate **regression scope increase** not to endanger quality?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q12. How **Important/Urgent/Difficult** it is to find **areas of increased risk** (defect prone) to be tested with more focus?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q13. How **Important/Urgent/Difficult** it is to effectively **balance between CRT, CIT, and CDRT** test coverage?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q14. How **Important/Urgent/Difficult** it is to build effective **defect prediction models**?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q15. How **Important/Urgent/Difficult** it is to secure proper **competence ramp up** of test engineers?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q16. How **Important/Urgent/Difficult** it is to accurately **measure test effectiveness**?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

Q17. How **Important/Urgent/Difficult** it is to manage **duplication of effort** between test teams?

	Very low	Low	Medium	High	Very high	I don't know
Importance	○	○	○	○	○	○
Urgency	○	○	○	○	○	○
Difficulty	○	○	○	○	○	○

18. What **other challenges** do You see on System Test Level?

Enter your answer ...

19. How long do You work in the Software Engineering field?

- ○ Less than 3 years
- ○ Between 3 and 6 years
- ○ Between 6 and 12 years
- ○ More than 12 years
- ○ I don't want to answer

20. What is Your **current role?**

- ○ Technical
- ○ Management
- ○ I don't want to answer

Appendix B. Survey results

Original forms are available in Supplementary Material: https://doi.org/10.5281/zenodo.6945430

See Tables B.1 and B.2.

Table B.1
Raw results of the survey.

Q1–Q17. Challenge evaluations.									
Challenge	Imp. VH	Urg. VH	Dif. VH	Imp. H	Urg. H	Dif. H	Imp. M	Urg. M	Dif. M
1. Corner-case testing	93	39	96	130	91	101	54	115	70
2. Low occurrence failures	58	18	144	117	77	93	107	138	52
3. Performance testing	195	109	79	95	125	116	13	56	85
4. Customer scenario testing	221	172	81	74	86	105	14	48	95
5. Hidden feature dependencies	82	39	132	141	112	92	62	107	49
6. OTA test scope	91	47	78	103	95	99	43	83	51
7. HW configuration-specific problems	91	51	143	130	103	84	68	112	55
8. Exploratory testing	80	37	63	116	88	101	84	121	103
9. Maintenance testing	97	55	26	126	100	79	66	105	134
10. Useful lifetime of a test scenario	53	20	28	99	67	69	106	133	124
11. Regression scope increase	83	41	45	133	107	89	70	121	120
12. Areas of increased risk	115	62	71	116	125	99	55	85	98
13. Balance CRT, CIT, and CDRT	89	48	39	100	86	67	71	107	128
14. Defect prediction models	51	24	88	117	79	87	63	104	54
15. Competence ramp up	167	98	62	101	113	100	22	71	101
16. Measures of test effectiveness	83	41	50	122	86	102	76	125	108
17. Duplication of effort	105	68	48	116	102	92	61	101	112
Challenge	Imp. L	Urg. L	Dif. L	Imp. VL	Urg. VL	Dif. VL	Imp. IDK	Urg. IDK	Dif. IDK
1. Corner-case testing	12	41	16	2	4	2	21	22	27
2. Low occurrence failures	23	67	12	2	5	2	5	7	9
3. Performance testing	3	10	13	0	0	2	6	12	17
4. Customer scenario testing	2	4	12	0	0	3	1	2	16
5. Hidden feature dependencies	6	24	7	2	5	1	19	25	31
6. OTA test scope	2	9	5	2	3	0	71	75	79
7. HW configuration-specific problems	15	32	16	1	1	1	7	13	13
8. Exploratory testing	15	43	20	0	0	0	17	23	25
9. Maintenance testing	8	34	49	4	4	6	11	14	18
10. Useful lifetime of a test scenario	21	54	47	1	3	3	32	35	41
11. Regression scope increase	6	21	31	3	3	2	17	19	25
12. Areas of increased risk	4	14	12	0	0	0	22	26	32
13. Balance CRT, CIT, and CDRT	9	27	27	4	4	6	39	40	45
14. Defect prediction models	17	35	4	4	7	2	60	63	77
15. Competence ramp up	3	11	20	0	0	2	19	19	27
16. Measures of test effectiveness	14	37	28	3	6	3	14	17	21
17. Duplication of effort	17	27	39	0	0	0	13	14	21

Table B.2
Raw results of the survey.

18. What other challenges do You see on System Test Level?	
Responses	127 (41%)
Detailed on the text responses were not disclosed...	
19. How long do You work in the Software Engineering field?	
Less than 3 years	52 (17%)
Between 3 and 6 years	63 (20%)
Between 6 and 12 years	69 (22%)
More than 12 years	111 (36%)
I don't want to answer	17 (5%)
20. What is Your current role?	
Technical	215 (69%)
Management	75 (24%)
I don't want to answer	22 (7%)

Appendix C. Funnel plots

See Figs. C.1–C.3.

Appendix D. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.infsof.2022.107067.

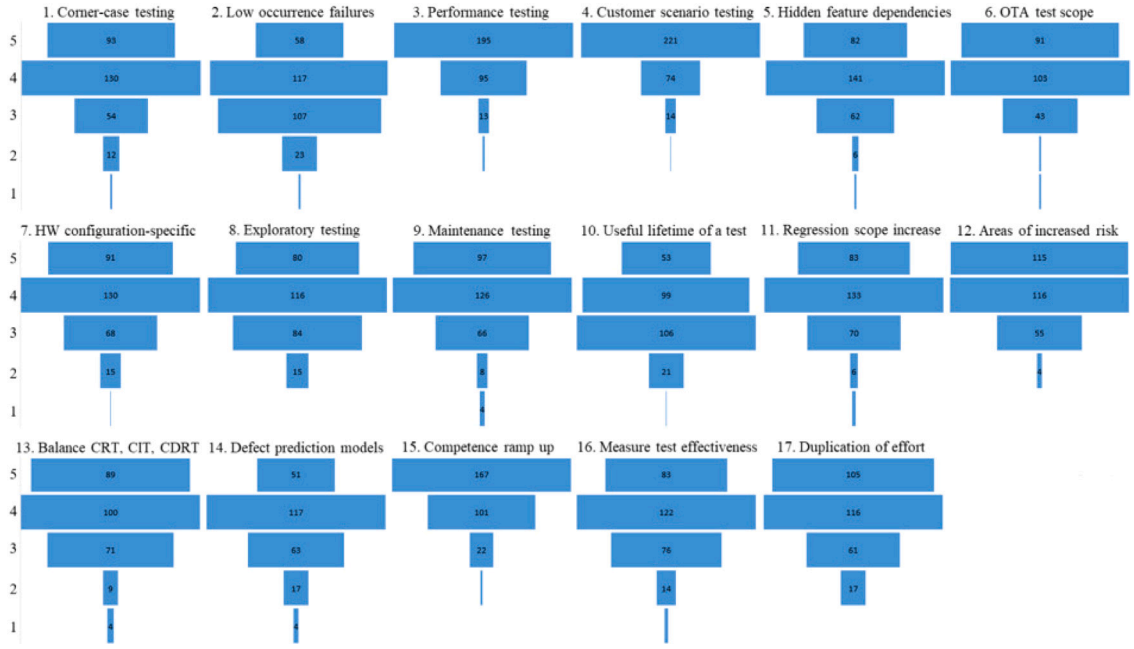


Fig. C.1. Funnel plots for 'Importance'.

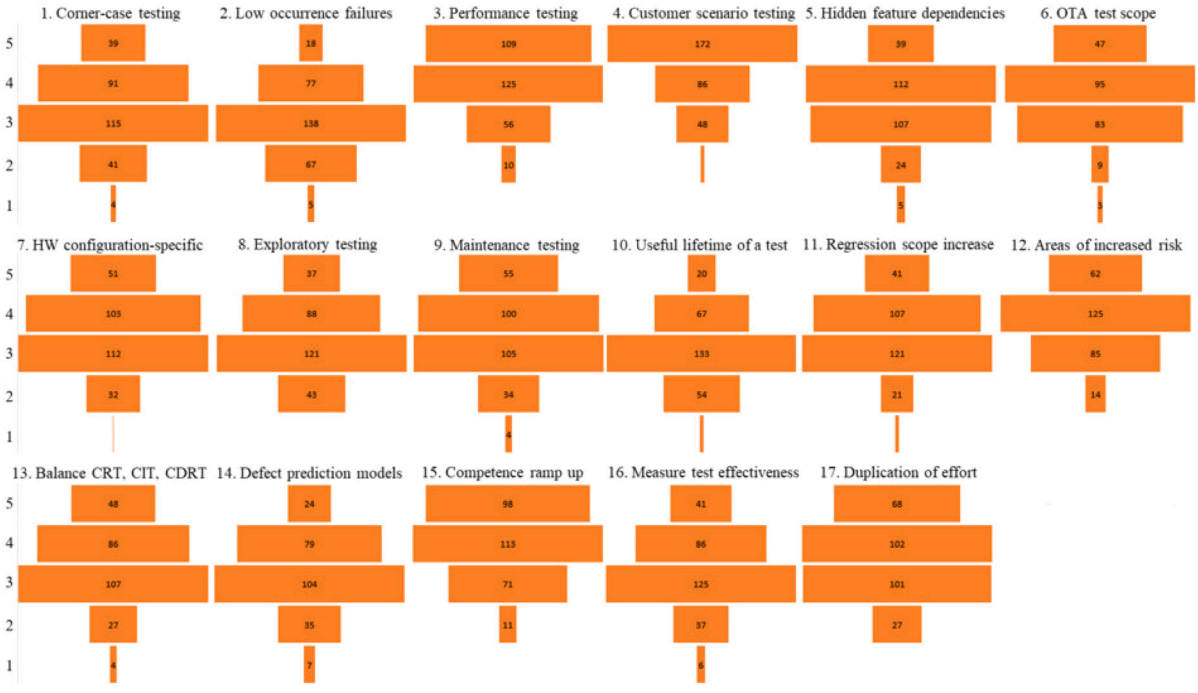


Fig. C.2. Funnel plots for 'Urgency'.

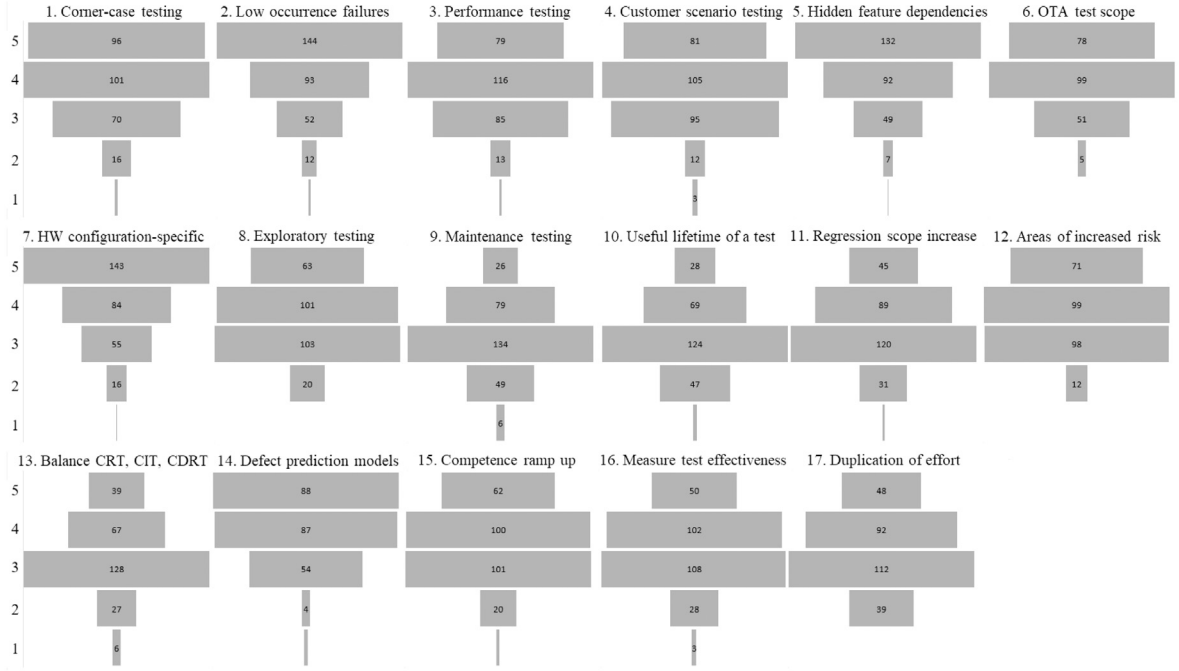


Fig. C.3. Funnel plots for 'Difficulty'.

References

[1] CHAOS REPORT 2015, Technical Report, The Standish Group International, Inc., 2015, URL: https://standishgroup.com/. (Accessed 20 November 2021).

[2] Nokia Corporation, Nokia annual report 2020, 2021, URL: https://www.nokia.com/system/files/2021-03/Nokia_Form_20F_2020.pdf. (Accessed 25 November 2021).

[3] A.R. Hevner, Phase containment metrics for software quality improvement, Inf. Softw. Technol. 39 (13) (1997) 867–877, http://dx.doi.org/10.1016/S0950-5849(97)00050-5.

[4] A. Endres, D. Rombach, A Handbook of Software and Systems Engineering, Addison-Wesley, 2003.

[5] S. Masuda, Y. Nishi, K. Suzuki, Complex software testing analysis using international standards, in: 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops, ICSTW, 2020, pp. 241–246, http://dx.doi.org/10.1109/ICSTW50294.2020.00049.

[6] R. Ben Abdessalem, A. Panichella, S. Nejati, L.C. Briand, T. Stifter, Testing autonomous cars for feature interaction failures using many-objective search, in: 2018 33rd IEEE/ACM International Conference on Automated Software Engineering, ASE, 2018, pp. 143–154, http://dx.doi.org/10.1145/3238147.3238192.

[7] H. Zhong, L. Zhang, S. Khurshid, TestSage: regression test selection for large-scale web service testing, in: 2019 12th IEEE Conference on Software Testing, Validation and Verification, ICST, 2019, pp. 430–440, http://dx.doi.org/10.1109/ICST.2019.00052.

[8] E. Piri, P. Ruuska, T. Kanstrén, J. Mäkelä, J. Korva, A. Hekkala, A. Pouttu, O. Liinamaa, M. Latva-aho, K. Vierimaa, H. Valasma, 5GTN: A test network for 5G application development and testing, in: 2016 European Conference on Networks and Communications, EuCNC, 2016, pp. 313–318, http://dx.doi.org/10.1109/EuCNC.2016.7561054.

[9] Y. Qi, G. Yang, L. Liu, J. Fan, A. Orlandi, H. Kong, W. Yu, Z. Yang, 5G over-the-air measurement challenges: overview, IEEE Trans. Electromagn. Compat. 59 (6) (2017) 1661–1670, http://dx.doi.org/10.1109/TEMC.2017.2707471.

[10] P. Zhang, X. Yang, J. Chen, Y. Huang, A survey of testing for 5G: Solutions, opportunities, and challenges, China Commun. 16 (1) (2019) 69–85, http://dx.doi.org/10.12676/j.cc.2019.01.007.

[11] International Organization for Standardization, Software and systems engineering — Software testing, 2013, URL: https://www.iso.org/standard/45142.html. (Accessed 20 November 2021).

[12] V. Garousi, T. Varma, A replicated survey of software testing practices in the Canadian province of Alberta: What has changed from 2004 to 2009? J. Syst. Softw. 83 (2010) 2251–2262, http://dx.doi.org/10.1016/j.jss.2010.07.012.

[13] V. Garousi, J. Zhi, A survey of software testing practices in Canada, J. Syst. Softw. 86 (5) (2013) 1354–1376, http://dx.doi.org/10.1016/j.jss.2012.12.051.

[14] A. Begel, T. Zimmermann, Analyze this! 145 questions for data scientists in software engineering, in: Proceedings of the 36th International Conference on Software Engineering, ICSE 2014, ACM, New York, NY, USA, 2014, pp. 12–23, http://dx.doi.org/10.1145/2568225.2568233, URL: http://doi.acm.org/10.1145/2568225.2568233.

[15] Y. Wang, M. Mäntylä, S. Demeyer, K. Wiklund, S. Eldh, T. Kairi, Software test automation maturity: A survey of the state of the practice, in: Proceedings of the 15th International Conference on Software Technologies, ICSOFT 2020, 2020, pp. 27–38, URL: https://arxiv.org/abs/2004.09210.

[16] D.I. Sjøberg, J.E. Hannay, O. Hansen, V.B. Kampenes, A. Karahasanovic, N.-K. Liborg, A.C. Rekdal, A survey of controlled experiments in software engineering, IEEE Trans. Softw. Eng. 31 (9) (2005) 733–753, http://dx.doi.org/10.1109/TSE.2005.97.

[17] V.R. Basili, G. Caldiera, H.D. Rombach, Encyclopedia of Software Engineering, John Wiley, 1994.

[18] International Software Testing Qualifications Board, Foundation Level Syllabus, 2018, pp. 30–38, URL: https://www.istqb.org/downloads. (Accessed 10 November 2021).

[19] M. Virmani, Understanding DevOps: bridging the gap from continuous integration to continuous delivery, in: Fifth International Conference on the Innovative Computing Technology, INTECH 2015, 2015, pp. 78–82, http://dx.doi.org/10.1109/INTECH.2015.7173368.

[20] T. Pyzdek, The Six Sigma Handbook: A Complete Guide for Green Belts, Black Belts, and Managers at All Levels, McGraw-Hill Companies, 2003, pp. 269–273, http://dx.doi.org/10.1036/0071415963.

[21] The 3rd Generation Partnership Project, 3GPP REL15, 2021, URL: https://www.3gpp.org/release-15. (Accessed 10 November 2021).

[22] M. Shafi, A.F. Molisch, P.J. Smith, T. Haustein, P. Zhu, P.D. Silva, F. Tufvesson, A. Benjebbour, G. Wunder, 5G: a tutorial overview of standards, trials, challenges, deployment, and practice, IEEE J. Sel. Areas Commun. 35 (6) (2017) 1201–1221, http://dx.doi.org/10.1109/JSAC.2017.2692307.

[23] S. Yoo, M. Harman, Regression testing minimization, selection and prioritization: A survey, Softw. Test. Verif. Reliab. 22 (2) (2012) 67–120, http://dx.doi.org/10.1002/stv.430.

[24] J.A. Jones, M.J. Harrold, Test-suite reduction and prioritization for modified condition/decision coverage, IEEE Trans. Softw. Eng. 29 (3) (2003) 195–209, http://dx.doi.org/10.1109/TSE.2003.1183927.

[25] B. Kitchenham, S. Pfleeger, Personal opinion surveys, in: Guide to Advanced Empirical Software Engineering, Springer London, 2008, pp. 63–92, http://dx.doi.org/10.1007/978-1-84800-044-5_3.

[26] J.S. Molléri, K. Petersen, E. Mendes, Survey guidelines in software engineering: an annotated review, in: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '16, Association for Computing Machinery, New York, NY, USA, 2016, http://dx.doi.org/10.1145/2961111.2962619.

[27] M. Kasunic, Designing an Effective Survey, Software Engineering Institute, 2005, http://dx.doi.org/10.1184/R1/6573062.v1.

[28] J. Linåker, S. Sulaman, R. Maiani de Mello, M. Höst, Guidelines for Conducting Surveys in Software Engineering, [Publisher information missing], 2015.

[29] V.R. Basili, G. Caldiera, H.D. Rombach, The goal question metric approach, 1994.

[30] R. van Solingen, E. Berghout, The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development, McGraw-Hill, 1999.
[31] M. Ciolkowski, O. Laitenberger, S. Vegas, S. Biffl, Practical experiences in the design and conduct of surveys in empirical software engineering, in: R. Conradi, A.I. Wang (Eds.), Empirical Methods and Studies in Software Engineering: Experiences from ESERNET, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 104–128, http://dx.doi.org/10.1007/978-3-540-45143-3_7.
[32] B. Kitchenham, D. Budgen, P. Brereton, Evidence-based software engineering and systematic reviews, 2015, pp. 233–242, http://dx.doi.org/10.1201/b19467.
[33] L. Madeyski, B. Kitchenham, Would wider adoption of reproducible research be beneficial for empirical software engineering research? J. Intell. Fuzzy Systems 32 (2) (2017) 1509–1521, http://dx.doi.org/10.3233/JIFS-169146, URL: http://madeyski.e-informatyka.pl/download/MadeyskiKitchenham17JIFS.pdf.
[34] R. Johns, Likert items and scales, Technical Report, Survey Question Bank, 2010, URL: http://www.surveynet.ac.uk/sqb/datacollection/likertfactsheet.pdf. (Accessed 16 January 2022).
[35] V. Garousi, M. Felderer, Worlds apart - industrial and academic focus areas in software testing, IEEE Softw. 34 (5) (2017) 38–45.
[36] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, Experimentation in Software Engineering, The Kluwer International Series in Software Engineering, 2000.
[37] X. Zhou, Y. Jin, H. Zhang, S. Li, X. Huang, A map of threats to validity of systematic literature reviews in software engineering, in: 2016 23rd Asia-Pacific Software Engineering Conference, APSEC, 2016, pp. 153–160, http://dx.doi.org/10.1109/APSEC.2016.031.
[38] R. Feldt, A. Magazinius, Validity threats in empirical software engineering research - an initial survey, in: Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering, SEKE'2010, 2010, pp. 374–379.
[39] D. Lo, N. Nagappan, T. Zimmermann, How practitioners perceive the relevance of software engineering research, in: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015, ACM, New York, NY, USA, 2015, pp. 415–425, http://dx.doi.org/10.1145/2786805.2786809.
[40] A. Kasoju, K. Petersen, M.V. Mäntylä, Analyzing an automotive testing process with evidence-based software engineering, Inf. Softw. Technol. 55 (7) (2013) 1237–1259.
[41] T. Sizer, D. Samardzija, H. Viswanathan, S.T. Le, S. Bidcar, P. Dom, E. Harstead, T. Pfeiffer, Integrated solutions for deployment of 6G mobile networks, J. Lightwave Technol. (2021) 1, http://dx.doi.org/10.1109/JLT.2021.3110436.