SecureScale: Exploring Synergies between Security and Scalability in Software Development and Operation

Inger Anne Tøndel and Gunnar Brataas
inger.a.tondel@sintef.no;gunnar.brataas@sintef.no
SINTEF Digital
Trondheim, Norway

ABSTRACT

Security and scalability are core software qualities, which as non-functional aspects share certain characteristics and challenges in how they are approached during software development and operation. Based on expert interviews, this paper explores interactions and dependencies between security and scalability, as well as similarities and differences in their challenges. It concludes that the current understanding of the relationship between security and scalability is not yet mature. Further, it points to future research needs to better understand the relationship between these two quality aspects and better support practitioners in addressing security and scalability in a more integrated fashion.

CCS CONCEPTS

• **Security and privacy → Software security engineering**; • **Software and its engineering → Software performance**; **Software design tradeoffs**.

KEYWORDS

quality requirements, non-functional requirements, agile software development, software security, scalability, software performance engineering, DevOps

ACM Reference Format:
Inger Anne Tøndel and Gunnar Brataas . 2022. SecureScale: Exploring Synergies between Security and Scalability in Software Development and Operation . In *Proccedings of the European Interdisciplinary Cybersecurity Conference (EICC 2022), June 15–16, 2022, Barcelona, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3528580.3528587

1 INTRODUCTION

A recent report estimated the cost of poor software quality in the US in 2020 to be $2.1 trillion [13]. The software quality *security* is considered important for most software development projects today [10], as software needs to function properly also under malicious attacks [16]. For systems with a high workload, the software quality *scalability* is also important to increase the system's capacity by consuming more (hardware) resources [8].

EICC 2022, June 15–16, 2022, Barcelona, Spain

ACM ISBN 978-1-4503-9603-5/22/06.
https://doi.org/10.1145/3528580.3528587

Current literature points to both these qualities and the related quality of performance co-existing in software systems. Good *performance* is a prerequisite for good scalability. You cannot scale if response times are too high with a modest number of users. In an interview study by Jarzębovicz et al. [10], security was considered important by virtually all interviewees, while the importance of other qualities like performance varied between sectors - meaning that projects with performance concerns also likely would have security concerns. According to Viega, the software security industry "now has to consider how its solutions impact the cost, reliability, and performance for busy cloud workloads" [22]. In a study by Olsson et al. [17], 48% of quality features identified were classified as performance, while 30% were classified as security. Also, API management serves as an example of a software technology that encompasses both security and scalability as central qualities [15].

We have experienced that practitioners we interact with see the need for security and scalability. Even five years ago these quality areas were important and rising. To illustrate, we point to the results of an informal survey with 60 participants in a practitioners conference in Norway in the autumn of 2017. This survey showed that 70% were concerned with security in their current work and 53% were concerned with scalability. About 60% of participants agreed that scalability was a key challenge in their projects and 90% considered scalability to become a more important challenge in the future. For security the numbers were quite similar, with a bit more than 50% considering security to be a key challenge in their current projects and 90% considering it to become more important in the future. Thus, these practitioners envisioned a current and future need to handle both security and scalability concerns during the software lifecycle. We see no reason why recent events should have reduced this need.

With security and scalability as important software qualities for current and future software systems, it is key to consider both how these qualities are related and can influence each other, and how one can successfully work with both during software development and operation. This paper explores the benefits of considering security and scalability together throughout a software product's lifecycle. This may occur during requirements elicitation and prioritisation, during architecture and implementation work, during testing, and in operations. To explore the relationship between security and scalability, we in 2021 performed expert interviews with people in architect or leader roles within the development. Based on the interview findings, we propose SecureScale as an important research area. This is motivated by security and scalability co-existing, sharing common challenges, and having interactions that software projects need to handle.

This paper is organised as follows. Section 2 introduces the background and related work on security and scalability. Section 3 explains the research method, while Section 4 presents the results from the interviews. Section 5 introduces the research area of SecureScale and discusses research needs based on the interview findings. Section 6 concludes the paper.

2 BACKGROUND AND RELATED WORK

Software security can be defined as the ability of software to function properly also under malicious attacks [16]. It can encompass a variety of practices throughout the software development lifecycle, such as security requirements, risk analysis, security-based testing, and security-oriented code reviews [16].

Scalability can be defined as the ability of software to increase its capacity by consuming more (hardware) resources [8]. In contrast to the concept of performance [20], scalability [8] focuses on workload growth: more users, heavier users, and even users demanding lower response times. The capacity of a system refers to the maximum workload a system can handle within its performance objectives, typically measured with a 90 percentile response time limit. The workload is the product of work and load. We separate between the amount of *work* involved in *one* operation and the *frequency* of the operations, the *load* [8]. Writing a large document once a day will invoke a high work but a low load, whereas reading a few bytes several times per millisecond is a low work but high load.

As quality aspects, security and scalability share some common characteristics and challenges. They can be considered to be subjective and open for interpretation [17], and can be experienced as "fuzzy" [12]. Literature on non-functional requirements in agile software development (ASD) shows that such requirements are commonly neglected [1, 4, 11], that there can be a lack of recognition of these requirements by stakeholders [4, 11] and that there can be challenges related to documentation [1, 4, 11], to name a few. As the same challenges influence security and scalability, they can "join forces" to handle these challenges more broadly, instead of fighting for attention individually. We explore this later in this paper. In a recent Systematic Literature Review (SLR) on non-functional requirements (NFRs) in ASD, Jarzebozicz and Weichbrot [11] identified practices that may help, including “Start focusing on NFRs early in the project”, “Involve NFR specialists”, and “Involve multiple roles and viewpoints to elicit and/or review NFRs”. The expert interviews reported on in this paper revealed similar suggestions for improvements.

Many authors have pointed out the importance of handling interactions between different types of quality requirements [1, 17, 19], e.g., when security mechanisms cause performance challenges [2]. Interactions between quality aspects can be challenging to identify, understand and communicate [2]. Thus, they can surface quite late in a project when they are more difficult to handle. Software development projects can thus benefit from actively looking for these interactions earlier on. Despite this need, there is, to our knowledge, few publications that consider the relation between security and scalability/performance. With this paper we aim to address this knowledge gap. In the following, we give an overview of relevant research literature on the relation between security and scalability/performance.

Sachdeva and Chung [19] proposed an approach to handle security and performance requirements for projects using Scrum. Their emphasis was on development that involved big data and cloud technologies. Key principles in their approach were to consider security and performance requirements early, quantify performance, and have security and performance as part of acceptance criteria or Definition of Done (DoD). In a case study using their suggested approach, they encountered a situation where there were conflicts between security and performance. One team started working on a requirement related to local directory search, but later when a security requirement was introduced that limited directory searches to logged in users, the performance suffered. A spike ("an investigational user story that is time boxed to understand and scope the work involved" [19]) was created to handle this conflict. They pointed out that these changes were costly (in terms of effort, time, and cost) and that this cost could have been reduces if the conflicts had been identified earlier and a compromise achieved quicker.

Interactions between quality aspects can be addressed through a method such as the Architecture Tradeoff Analysis Method (ATAM) [3], which is a comprehensive method for evaluating software architectures relative to quality goals. Another example is the Quality Triage method [9] to identify quality requirements and identify and address conflicts between various quality aspects in the early stages of ASD projects. That method was applied to an autonomous cars scenario where both the qualities of security and scalability were addressed, in addition to the qualities of safety and availability.

Ribeiro et al. [18] performed a study of moderating factors for verification of security and performance requirements. They identified the following moderating factors: "Organization awareness of the importance of security and performance", "Cross-Functional teams", "Suitable requirements", "Suitable support tools", "Suitable verification environment", "Systematic verification methodology", Security and performance verification planning", and "Reuse practices".

Though security and scalability (including performance) share common characteristics as quality requirements, some authors have pointed out some differences to consider. Weir et al. [23] point out that security’s idea of an attacker implies the need for different thinking. Olsson et al. [17] claim that performance is difficult to estimate upfront, more so than security. Note however that this claim by Olsson et al. constrasts with established wisdom from software performance engineering [20], where establishing performance requirements is the first step. Without performance requirements it is impossible to test and hard to code. Therefore, the ScrumScale method for scalability engineering starts by identifying the user stories with scalability challenges [7].

3 RESEARCH METHOD

We used semi-structured expert interviews to explore the need to address security together with scalability and performance in software development and operation. Five interviews were performed with six experts from five organisations. As can be seen from Table 1, interviewees were recruited from a broad set of companies that all had both scalability and security as key quality concerns. The interviewees were highly experienced individuals, with roles such as an architect or chief developer, and had experience with working

Table 1: Interviewees

Id.	When	Length	Sector	Type of software	Role
I1	May 2021	90 min.	Private	Forecasting systems	Chief architect / developer
I2	May 2021	45 min.	Private	Database systems	Department lead
I3	Sept 2021	60 min.	Public	Public portal A	1) Security architect and 2) solution architect
I4	Nov 2021	60 min.	Public	Public portal B	Technical architect, performance responsible
I5	Dec 2021	90 min.	Private	Fintech systems	Security Officer / consultant

with both scalability and security concerns. The interviewees were recruited through our network.

As shown in Table 1, most interviews lasted about 1 h. The first and second authors performed all interviews, and all but one interview (I2) were recorded upon interviewee consent. The Norwegian Centre for Research Data was notified about the study. The interviews followed an open interview guide, with the following main elements:

- **Introductory questions:** What is your role in the company? Explain the research objective of exploring the relations between security and scalability/performance.
- **Experiences with security and scalability:** Do you need to deal with both these issues? Do they interact, in your experience? How? What is challenging when it comes to scalability and security? To what extent can one consider security and scalability separately? Why/why not?
- **Needs and interest in this topic:** What solutions or knowledge would you want to have on this topic? What is your thoughts about this as a research topic?

All of the recorded interviews were transcribed. The transcribed notes and the interview notes from interview I2 were analysed and coded using four organising codes: interactions and dependencies, similar challenges, differences, and way forward.

This explorative research approach matched our needs to 1) explore a topic (SecureScale) where the current literature - and thus theory - were limited, 2) quickly validate whether industry representatives found this potential research area relevant, and 3) identify candidate topics for future research on the relationship between security and scalability.

4 RESULTS FROM EXPERT INTERVIEWS

In the following we introduce the interactions and dependencies, similar challenges, and differences identified in the expert interviews.

4.1 Interactions and dependencies

There were significant variations between the interviewees on articulating interactions between security and scalability. Some interviewees (I2, I3) initially stated that they did not see that many direct relations, whereas the conversation that followed brought up several interactions between these two qualities, exemplified by the following quote from I3: *"It is not that often one reflects on this. When we first start digging and we get some input, then the ideas and thoughts come."*

Several interviewees brought up examples where security had a performance and/or scalability cost. Examples included encryption, authentication, database storage, network security zones, firewalls, and patching. I1 explained that they had opted for a private network where encryption was not required. On the other hand, I4 needed to use encryption for a private network when synchronising two data stores at different locations. This encryption consumed processing resources, but more importantly, introduced latency (I4). I5 discussed the challenges of using older authentication standards centred around sessions, such as SAML, for mobile or cloud applications. I4 explained that although re-authentication could be considered a light call (with low work), even these light calls could cause challenges when the number of users needing to re-authenticate became considerable (high load). Further, I4 talked about the need to reconsider file storage solutions due to security mechanisms, and in this case, opting for HTTP that gave faster access to metadata. I4 moreover explained that network security zone models restricted direct communication. Thus, they had performance implications. Also, there was hesitation to use Web Application Firewalls (WAFs) because of the potential performance implications of falsely identifying traffic as malicious. I4 further told about instances of security patches with severe performance implications.

Interviewees brought up examples where security and scalability could be addressed together. I5 explained how newer security solutions, like web-OAuth, Fido 2, and security enclaves on cell phones, allowed for authentication in a more distributed manner. This enabled both better end-user performance and improved overall system scalability. eIDAS 2, which regulates the use of electronic identification, could support better performance if systems used the eIDAS levels of assurance (low, substantial, high) and did not require authentication on a higher level than necessary. This opened up for more authentication on the mobile device, reducing the load on critical central authentication resources. Otherwise, such centralised authentication resources could easily give scalability problems. Another example was to pose limitations on the use of central storage. I5 explained how a solution for storing shared state restricted state size, because of previous scalability problems where developers had exploited the central shared storage with too large objects. Such uncritical use of shared storage could also give security and privacy issues.

Security could have implications for how one chooses to address scalability. The scaling up of a solution did not automatically lead to more security vulnerabilities, although scaling up from one to several servers could increase the security perimeter (I2). However, common scalability approaches came with security implications. Many interviewees talked about cloud solutions, and how they supported scalability. However, because of security concerns there

were some data that they did not want to put in the cloud (I3). Also, autoscaling features could result in high costs in the case of a DDoS attack (I3, I5). To reduce response times, an option was to have local copies. But this came with privacy and security challenges since extensive use of local copies would hamper the individual's ability to access their data and have their data deleted or updated (privacy), as well as increasing the complexity of the system and adding more places to control access (security). Moreover, the organisation of I4 had previously experienced a well known incident where caching was the cause of a leak of personal data, where data from one customer was shared with subsequent customers. As a result this organisation had become cautious in using cashing to improve scalability, and therefore needed more hardware resources.

Good scalability could be helpful for security. I1 explained that one option to gain unauthorised access to a system was to stress the system's scalability and see if vulnerabilities appeared. I5 talked about the importance of building resilience into the system to avoid cascading failures; something that would benefit both scalability and security. I1 talked about solutions for extracting immutable static data that could be easily cached and replicated to improve scalability, which also limited access to vulnerable core data sets. Moreover, security services needed to scale and expertise on scalability was important to do this successfully. I4 provided an example of a single sign-on solution where adding more servers did not improve its performance but worsened it due to the Ringelman effect [7].

It did not seem that these qualities currently competed for resources or attention, although this was seen as a potential challenge. On the contrary, I5 provided an example where a security risk analysis triggered work on performance and scalability by identifying availability risks.

4.2 Similar challenges

The interviewees experienced limited support for NFRs within ASD (I3, I5). Though early attention was considered important for both security and scalability, NFRs were challenging to sell to management at an early stage (I1) and, accordingly, non-functional aspects could be delayed until late in development (I5). NFRs could have a different source than functional requirements (e.g., not come from system owners) (I3). Even when NFRs were part of the contract they might still not be included in acceptance testing, which focused on functional requirements (I3). Developers were motivated to work on functionality since this was where they experienced the greatest customer demands (I1). The willingness to pay for NFRs could be limited (I3). Not all customers were aware of their needs regarding security, scalability, and performance (I5). However, at least for fintech, security and performance were partly covered by compliance requirements for sturdy operations (I5).

Working with security and scalability called for different ways of thinking than when working primarily with functionality. I1 explained that development enjoyed change and typically ignored operational problems. On the other hand, governance and operations liked stability. I5 explained that some types of personalities enjoyed and mastered performance testing better, e.g., as they had more structured thinking and sense of order. Thinking like an attacker requires imagination and specific skills (I3). One could argue that this is also required to anticipate scalability challenges in complex systems. Understanding non-functional aspects were described as complex and the requirements could be vague (I1). Several interviewees pointed to challenges related to available resources and competence, and a need for more training, including at the university.

Technological aspects could have similar implications both for security and scalability. As cross-cutting concerns they both required cooperation between people with different types of competencies. In brownfield development, both were limited by legacy systems, which were hard to change (I2, I5). Moreover, both scalability and security could depend on components they did control (I1, I3, I5). One weak component in a chain of software components could cause challenges for the whole chain. For both security and scalability, it was necessary to have an architecture that supported these qualities (I5).

When it came to methods, I5 explained that in their software development process, they used one method for tagging both security and performance/scalability issues with backlog items, according to the ScrumScale method [7]. This suggests that software development projects can address both qualities in a similar fashion.

4.3 Differences

I5 pointed out that scalability was not relevant for all systems, while security seemed to be more generally relevant. Which aspect was considered most important, however, seemed to vary. I2 stated that security always won over performance because one could not risk having a reputation for not taking security seriously. This was somewhat in contrast to I1 who stated that their products failed if they did not scale, but at the same time, the products also had to be secure. Technically, the direction of influence could vary between security and scalability. I4 explained that in cases where there were a high workload, a security issue would create a scalability issue but not the other way around.

There were signs that security had a somewhat higher status than scalability in most interviewed organisations. Generally, the organisations had a security department, but not a scalability department (I1, I2). Top-down compliance requirements were more frequently mentioned for security compared to scalability. The security lead role seemed more established than the performance/scalability lead (or similar) role in agile teams (I5), even though performance/scalability leads also existed (I1, I4). On the other hand, there could be testers assigned to performance without this being the case for security (I5). I5 talked about training for all developers for security, while training for performance/scalability did not involve all developers.

The ongoing shift towards cloud-native solutions could potentially to change the way scalability was viewed and addressed compared to security. With cloud-native solutions, some of the cost used to code and tune a system would be shifted towards operations, and therefore costs became more visible; *"with cloud native architecture I have discovered that performance (...) come forward faster, as it concerns how much you must pay"* (I5). This seemed to increase training for and focus on performance and scalability, but not security.

Different experts could handle security and scalability , and organisational boundaries and culture could impact the ability for

scalability and security experts to contribute to each other's areas. A security architect (I3) perceived that he should not contribute to scalability because that implied entering someone else's area (I3). Note however, that I1 and I5 would be examples of experts with a strong competence and contribution in both security and scalability.

5 THE RESEARCH AREA OF SECURESCALE

Based on the interviews and the existing research literature, we find that the understanding of the relationship between security and scalability is not mature. The interviews documented several interactions between security and scalability. Still, it is our impression that some of the interviewed experts only became aware of these interactions through participating in this interview. This points to the need for SecureScale as a research area to support both researchers and practitioners to become more aware of existing interactions and their implications.

Though the limitations in awareness of the interactions between security and scalability point to SecureScale not being a primary challenge today, ongoing trends point to SecureScale becoming more important in the future. Bosch [6] has explained how the current evolution of software engineering is driven by three factors: speed, data, and ecosystems. Customer needs must be responded to at "unprecedented speeds", thus organisations need "enterprisewide agility". Companies increasingly collect large amounts of data from products in the field, to use this data to inform decision making. Moreover, companies are moving from emphasising one-to-one customer relationships to "creating and contributing to an ecosystem of players". All this requires new software architectures and new ways of working with software engineering [6], that also have implications for SecureScale.

As software is being delivered more in an ecosystem fashion, the reliance on components offered by other players will be an increasing concern. When software services consist of chains of software components operated by different entities, both security and scalability will be dependent on the quality of all components in the chain. This increases the importance of addressing security and scalability in all software development. Further, it points to a need to improve the ability to analyse the security and scalability of complex component chains and assess the quality of software components offered by others. Challenges related to component chains came up in the expert interviews. Thus, we point to SecureScale in component chains as an important research topic.

The increased speed and the drive towards increased agility organisation-wide have implications also for SecureScale. Interviews and literature both point to challenges related to quality aspects in ASD. Though many of these challenges are not necessarily specific for ASD but also present in more traditional software development approaches, they need to be addressed within an ASD context. Bosch [6] talks about a move towards cross-functional teams and self-management. Further, Bosch points to a move towards "unprecedented architecture modularity and flexibility" where "team autonomy, modularity and flexibility are being prioritized over other quality attributes" [6]. Some of the challenges expressed in the expert interviews that show a divide between security and scalability experts (I3) may thus be less relevant in the future. However, challenges with addressing the overarching concept of SecureScale in complex systems can become more challenging. Interviewees called for improved processes that foster early cooperation among experts. More research is needed on how to address SecureScale in large scale ASD projects.

Software security in ASD has been a research field for 20 years [22], and a lot of knowledge is available on challenges, practices and methods [5, 21]. For scalability, the research literature is less extensive regarding ASD, with a few exceptions [7, 8]. As the expert interviews identify several similarities between security and scalability, it is reasonable to assume that some of the knowledge about software security in ASD can bring relevant insights also to the scalability research field. The experiences from I4 on successfully using ScrumScale [7, 8] for tagging both security and scalability concerns in the backlog points to the benefits of moving towards a common approach for addressing these quality aspects in ASD. There are already many proposals for integrating individual quality aspects into ASD [1]. However, the resulting approach will become quite heavy with separate methods for each quality aspect, even if the individual methods may be easy and match well with ASD principles [1]. Still, there are real differences between security and scalability, e.g., scalability problems are caused by legitimate users, whereas malicious actors cause security problems. Thus, more research is needed on holistic methods for addressing quality requirements. The Quality Triage [9] is one possible method to build on in this respect.

This, however, leads to an important question that we grapple with in this paper: to what extent can security and scalability be considered separately as quality concerns? The interactions and dependencies pointed out in the expert interviews show that there are cases where there is a need to make trade-offs between security and scalability. However, it also seems that in some cases the fact that security and scalability concerns are both present in a given software system does not necessarily change the approach to security or scalability. Based on the interviews, it is not clear what characteristics to look for to know when these aspects should be considered together and when they can be addressed separately. More research is needed to be able to tell the difference and provide support to practitioners in this respect.

The interviews bring some evidence that security is more established as a quality concern in the organisations than what is the case for scalability. To a larger extent, there are compliance requirements and dedicated roles for security. Today, security is important for most systems, i.e., it should at least be considered for all Internet-connected systems. However, you only get scalability challenges with a high workload. Thus, security is more ubiquitous. An interesting observation from I5 was that security risk analysis could also bring up scalability and performance issues due to the consideration of availability risks. We disagree with Olsson et al. [17] that performance is particularly difficult to estimate up front. Rather, shift left is important for both scalability (including performance) and security. Interviews give the impression that currently, it is somewhat dependent on personal competence and interest to what extent both security and scalability are thoroughly addressed. Further research could explore the possibility of using the security roles and structures available in organisations to ensure that

both security and scalability are addressed in a structured manner throughout a software's lifecycle.

As security and scalability are just two of a larger set of quality aspects that can be important for a software system, it is relevant to consider to what extent there is anything special about the relationship between security and scalability or if the same types of interactions, similarities and differences would be present also for other types of qualities. The available literature points to security having a potential usability cost [14]. Also, challenges related to ASD have been identified for a broader set of quality aspects [1, 4, 11]. However, more research is needed to know what (if any) are the general characteristics of the relationship between quality aspects, and what is specific for SecureScale.

A major challenge expressed by interviewees is competence. This concerns competence on security and scalability individually and competence on their relation. However, this leads us to the question: Is SecureScale primarily a competence problem, or is it also a technological challenge? Posed differently, can SecureScale be achieved by training alone? From the interviews, it seems that the most pressing need is to build competence. However, interviews also point to the importance of technology that supports both. For example, I5 explains how the move from SAML to OAuth supported both scalability and security. We expect more such positive-sum solutions to reduce the need for future trade-offs between security and scalability. Some technological trends call for improved technology for SecureScale. The size of software systems is increasing, calling for new architectural approaches [6]. Also, the increased size of the systems, brownfield development, and the many technological alternatives available to system developers and operational personnel increase complexity (I5), thus making SecureScale more technically challenging.

6 CONCLUSION

This paper explores the need for SecureScale as a research area for software engineering. Through expert interviews, we found that the current understanding among practitioners of the relationship between security and scalability is not yet mature. At the same time, limited research contributes to understanding this relationship. Thus, we point out several research topics that should be explored.

These conclusions are based on interviews with six experts from five organisations. This limited number of experts was sufficient to explore this topic, and we experienced that interviewees shared many of the same experiences and reflections despite covering a broad set of organisations and systems. Still, five interviews are not enough to be confident that we have reached saturation of the topic. More research is needed to understand the field of SecureScale more broadly, and we would encourage researchers to perform more studies on this topic to complement and extend the findings presented in this paper.

ACKNOWLEDGMENTS

This work was supported by the SINTEF strategic project *Multilities: Agile Mastering of Security, Safety, Scalability and Availability Requirements*, and funded by the Research Council of Norway, grant # 309811, Capacity Guard of Public Digital Services. We thank all interviewees for access to mission-critical information.

REFERENCES

[1] Wasim Alsaqaf, Maya Daneva, and Roel Wieringa. 2017. Quality requirements in large-scale distributed agile projects–a systematic literature review. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 219–234.

[2] Wasim Alsaqaf, Maya Daneva, and Roel Wieringa. 2019. Quality requirements challenges in the context of large-scale distributed agile: An empirical study. *Information and software technology* (2019).

[3] Len Bass, Paul Clements, and Rick Kazman. 2012. *Software Architecture in Practice* (Third ed.). Addison-Wesley.

[4] Woubshet Behutiye, Pertti Karhapää, Lidia López, Xavier Burgués, Silverio Martínez-Fernández, Anna Maria Vollmer, Pilar Rodríguez, Xavier Franch, and Markku Oivo. 2020. Management of quality requirements in agile and rapid software development: A systematic mapping study. *Information and Software Technology* 123 (2020), 106225. https://doi.org/10.1016/j.infsof.2019.106225

[5] Lotfi ben Othmane, Martin Gilje Jaatun, and Edgar Weippl. 2017. *Empirical research for software security: foundations and experience*. CRC Press.

[6] Jan Bosch. 2015. Speed, data, and ecosystems: the future of software engineering. *IEEE Software* 33, 1 (2015), 82–88.

[7] Gunnar Brataas, Geir Kjetil Hanssen, Nikolas Herbst, and Andre van Hoorn. 2020. Agile Scalability Engineering: The ScrumScale Method. *IEEE Software* 37, 5 (September–October 2020). https://doi.org/10.1109/MS.2019.2923184

[8] Gunnar Brataas, Antonio Martini, Geir Kjetil Hanssen, and Georg Ræder. 2021. Agile Elicitation of Scalability Requirements for Open Systems: A Case Study. *Journal of Systems and Software* 182 (December 2021), 111064. https://doi.org/10.1016/j.jss.2021.111064

[9] Gunnar Brataas, Inger Anne Tøndel, Eivind Okstad, Ola Løkberg, Martin Gilje Jaatun, Geir Kjetil Hanssen, and Thor Myklebust. 2020. The Quality Triage Method: Quickly Identifying User Stories with Quality Risks. In *Societal Automation*. IEEE.

[10] Aleksander Jarzębowicz and Paweł Weichbroth. 2021. A Qualitative Study on Non-Functional Requirements in Agile Software Development. *IEEE Access* 9 (2021), 40458–40475. https://doi.org/10.1109/ACCESS.2021.3064424

[11] Aleksander Jarzębowicz and Paweł Weichbroth. 2021. A Systematic Literature Review on Implementing Non-functional Requirements in Agile Software Development: Issues and Facilitating Practices. In *International Conference on Lean and Agile Software Development*. Springer, 91–110.

[12] Pertti Karhapää, Woubshet Behutiye, Pilar Rodríguez, Markku Oivo, Dolors Costal, Xavier Franch, Sanja Aaramaa, Michał Choraś, Jari Partanen, and Antonin Abherve. 2021. Strategies to manage quality requirements in agile software development: a multiple case study. *Empirical Software Engineering* 26, 2 (2021), 1–59.

[13] Herb Krasner. 2021. *The cost of poor software quality in the US: A 2020 report*. Technical Report. Proc. Consortium Inf. Softw. QualityTM (CISQTM).

[14] Oksana Kulyk, Stephan Neumann, Jurlind Budurushi, and Melanie Volkamer. 2017. Nothing Comes for Free: How Much Usability Can You Sacrifice for Security? *IEEE Security Privacy* 15, 3 (2017), 24–29. https://doi.org/10.1109/MSP.2017.70

[15] Max Mathijssen, Michiel Overeem, and Slinger Jansen. 2020. Identification of Practices and Capabilities in API Management: A Systematic Literature Review. *arXiv preprint arXiv:2006.10481* (2020).

[16] Gary McGraw. 2006. *Software Security: Building Security In*. Addison-Wesley.

[17] Thomas Olsson, Krzysztof Wnuk, and Tony Gorschek. 2019. An empirical study on decision making for quality requirements. *Journal of Systems and Software* 149 (2019), 217–233.

[18] Victor Vidigal Ribeiro, Daniela Soares Cruzes, and Guilherme Horta Travassos. 2022. Moderator factors of software security and performance verification. *Journal of Systems and Software* 184 (2022), 111137. https://doi.org/10.1016/j.jss.2021.111137

[19] Vaibhav Sachdeva and Lawrence Chung. 2017. Handling non-functional requirements for big data and IOT projects in Scrum. In *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*. IEEE, 216–221.

[20] Connie U. Smith and Lloyd G. Williams. 2001. *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*. Addison-Wesley.

[21] Inger Anne Tøndel and Martin Gilje Jaatun. 2022. Towards a conceptual framework for security requirements work in agile software development. In *Research Anthology on Agile Software, Software Development, and Testing*. IGI Global, 247–279.

[22] John Viega. 2020. 20 Years of Software Security. *IEEE Annals of the History of Computing* 53, 11 (2020), 75–78.

[23] Charles Weir, Awais Rashid, and James Noble. 2020. Challenging software developers: dialectic as a foundation for security assurance techniques. *Journal of Cybersecurity* 6, 1 (2020).