International Journal of Information Systems and Project Management (IJISPM)
2026, 14(1), e140102, DOI: 10.12821/ijispm140102

RESEARCH ARTICLE

Quality, security, and privacy assurance in software development: proactive integration or just workflow-slowing checkpoints?

Anne-Maarit Majanoja
University of Turku, Department of Computing, 20014 Turun yliopisto, Turku, Finland, anne-maarit.majanoja@utu.fi

Ville Leppänen
University of Turku, Department of Computing, 20014 Turun yliopisto, Turku, Finland, ville.leppanen@utu.fi

Abstract
In software development, the integration of assurance methodologies such as quality, security, and privacy practices is essential to producing high-quality, reliable, and compliant products. This paper investigates the adoption and effectiveness of these assurance practices within the daily operations of software development. Through an industry survey of 88 software development professionals in Finland, this study examines the order and consistency with which developers apply assurance practices during projects, and the challenges they face in performing these tasks. The results show that while developers recognize the importance of assurance, many organizations still treat it as a separate, secondary activity rather than a core part of the development lifecycle. Key findings show that quality practices are more consistently integrated into daily operations compared to security and privacy measures, which tend to be reactive. The paper highlights the tension between agile practices, which promote flexibility and continuous improvement, and the more rigid, process-heavy nature of assurance tasks. The study underscores the need for a shift in both industry practices and educational approaches to fully embed assurance into software development.

Keywords
quality; security and privacy; assurance practices; industry survey; software development practices; systemicity; compliance.

Received: 9 January 2025 | Accepted: 22 April 2025

1. Introduction

In any software development project, the importance of integrating assurance methodologies and best practices cannot be overemphasized. Software developers need to consider a variety of assurance methodologies, frameworks, and best practices that are necessary to ensure that the software development process is efficient, high quality, and secure. A basic understanding of quality, security, and privacy is part of the software engineering competency domain. These practices are not just formalities. Without them, even the most well-intentioned development efforts can fail, leaving vulnerabilities, inefficiencies, and compliance issues that can severely impact the success of a project.

Software developers must adopt a combination of proven frameworks, industry standards, and guidelines to deliver reliable software. These include practices such as Clean Code (Hutton, 2009), Agile (Agile Alliance, 2024), Software Development Lifecycle (Ruparelia, 2010), Dev(Sec)Ops (NIST, 2020a), and Continuous Integration/Continuous Development (Gallaba, 2019). These methodologies focus on iterative improvements, fast feedback, and continuous testing to ensure that the product is developed in a scalable and maintainable manner. Various quality and testing practices, such as Test-Driven Development (Beck, 2022), load and performance testing (Menascé, 2002; Neely et al., 2000), exploratory testing (Itkonen & Rautiainen, 2005), and penetration testing (Arkin et al., 2005) push the boundaries of quality assurance by not only identifying software defects, but also actively preventing security and privacy issues at the earliest possible stage. Security and privacy are critical in today's interconnected world, where breaches and leaks can cost organizations financially, legally, and reputationally. Adherence to OWASP (OWASP, 2024), the National Institute of Standards and Technology (NIST) Cybersecurity Framework (NIST, 2024b), and the International Organization for Standardization (ISO) 27001 (ISO27001, 2022) is not only good practice, and compliance with regulations such as the General Data Protection Regulation (GDPR) (GDPR, 2018) and the California Consumer Privacy ACT (CCPA) (CCPA, 2024) is not just a legal necessity. These are not optional tasks; they are critical to building software that works reliably. These practices are necessary to build trust with users who demand high-quality, secure, and privacy-enhancing solutions.

Implementing assurance practices is more than just ticking boxes. The adoption of assurance practices is essential to ensure robust and reliable results. Yet assurance activities can often feel like a "Band-Aid" applied to software development practices, perceived as slowing and complicating the workflow by addressing seemingly marginal issues. This paper emphasizes that *security* and *privacy* should not be subsumed under the broad term "*quality*" in software development, because of their distinct goals, characteristics, and critical importance. Although related, each of these areas addresses fundamentally different concerns and requires specialized methodologies, tools, expertise, and unique frameworks and standards. While security and privacy contribute to the overall quality of software, each has its own importance and requires separate evaluation processes. Framing them as quality, security, and privacy assurance emphasizes their equal importance while supporting collaborative approaches to align these areas with broader goals.

This paper aims to investigate how software developers engage with assurance practices throughout the software development process. We aim to understand the order in which developers follow and adhere to various practices during a project, and how well assurance practices are performed on a day-to-day basis. This paper addresses the following research questions:

RQ1: In what order do software developers follow and adhere to assurance practices during a project?

RQ2: How systematically do software developers select assurance practices and how are these practices regulated?

RQ3: How well do software developers implement or use the assurance practices in their organization?

To address these research questions, we surveyed 88 software development professionals in Finland to explore their quality and security assurance practices and their perceptions of software development processes. By answering these questions, the research aims to gain an understanding of the adoption, application, and performance of quality, security, and privacy assurance practices in the industry.

Clearly, the single term "quality" in software development does not fully capture the depth of developers' quality practices. In this paper, additional terms related to quality must be included to provide a broader understanding. *Quality* in software development encompasses various activities and aspects that ensure that the software meets user needs, business requirements, and technical standards. These include, such as functional quality, non-functional quality, process quality, operational quality, and considerations including compliance, satisfaction, and risk and change management. *Testing* plays a critical role in ensuring both functional and non-functional quality, while maintaining a sustainable and reliable software product throughout its lifecycle. Software quality must also address *technical debt* (Kruchten et al., 2012; Li et al., 2015; Tom et al., 2013). Technical debt has a direct impact on the internal quality, long-term maintainability, scalability, and performance of software, because it undermines the effectiveness of testing, increases the likelihood of defects, complicates test automation and regression testing, and degrades the overall quality of the software.

The paper is organized as follows: Section 2 provides an overview of previous studies and basic concepts in assurance practices. Section 3 outlines the research methods and details the research conducted. Section 4 presents the results, starting with the order of the following practices to be followed, then examining systemicity and compliance in the selection of assurance practices, followed by day-to-day assurance practices. Section 5 presents a discussion and Section 6 is the conclusion of the study.

2. Literature review

Software failures have repeatedly led to serious consequences, from financial losses and data breaches to loss of life, as seen in cases, such as the Y2K bug, Toyota's unintended acceleration, Ariane 5, and Therac-25 (Raygun.com, 2022;Medium.com, 2020;Wired.com, 2022) failures. These disasters highlight how failures in assurance and human error can have catastrophic consequences. Today, the risk of poor-quality, insecure software slipping through is higher than ever, and security and privacy breaches resulting from exploited vulnerabilities are common. For these reasons, knowledge and skills in quality, security, and privacy assurance are in high demand in software development. However, it is important to recognize that even these assurance practices do not guarantee a 100% error-free result.

Quality, security, and privacy form a triangle that covers the essential assurance aspects of information systems and processes. This triangular concept reflects the interrelationships and potential conflicts between these aspects, particularly in software development. By looking at them together, it is possible to ensure that a system or process is holistically reliable, secure, and privacy-aware. Looking at these practices together can help identify how changes or actions in one area affect the others. The following assurance definitions are defined and constructed using descriptions from ISO25002:2024 (ISO25002, 2024), IEEE 730-2014 (IEEE, 2014), CMMI (CMMI Institute, 2018), ISO27001 (ISO27001, 2022), NIST (NIST, 2024a), OWASP (OWASP, 2024), ISO15408 (ISO15408, 2022), GDPR (BBC News, 2021; GDPR, 2018), ISO29100 (ISO29100, 2024), CCPA (CCPA, 2024), and Privacy by Design - The 7 Foundational Principles (Cavoukian, 2009):

Quality assurance in software development is a systematic process designed to ensure that a product or service meets specific requirements. It involves systematic measurement, comparison to standards, process development and monitoring, and feedback loops that promote error prevention. Key aspects include documenting and enforcing defined processes during development, continuously monitoring processes, training team members, conducting audits and reviews to verify compliance with standards, and implementing feedback mechanisms to improve processes. (CMMI Institute, 2018; IEEE, 2014; ISO25002, 2024)

Security assurance in software development ensures that software is free of vulnerabilities throughout its lifecycle and performs as intended. The process involves assessing risks, defining security requirements, employing secure design and coding practices, and conducting thorough security testing. It also includes managing software configurations and changes, establishing incident response protocols, and adhering to relevant security standards and obtaining necessary

certifications to effectively mitigate potential security threats and ensure compliance. (ISO15408, 2022; ISO27001, 2022; NIST, 2020c, 2020b; OWASP, 2024)

Privacy assurance in software development ensures that software manages data in compliance with privacy standards and regulations, and respects user privacy throughout the data lifecycle. This includes building privacy into the earliest design phases, minimizing data collection to what is strictly necessary, and implementing strong access controls. It also includes encrypting and anonymizing data to protect user identities, complying with regulations such as GDPR and CCPA, and maintaining transparency and accountability in data use. (GDPR, 2018; ISO29100, 2024; CCPA, 2024; Cavoukian, 2009)

Webster and Watson (2002) emphasize that a literature review should summarize, synthesize, and critically analyze research to identify gaps, provide a conceptual framework, and suggest future directions. They recommend defining a clear scope, using systematic methods to evaluate sources, and organizing findings with an audience in mind, supported by visual tools such as tables and graphs. Because assurance practices are well known and have been extensively researched, this review focuses on the most recent research on assurance practices and differs from Webster and Watson's (2002) approach by not limiting the review to the timeframe of the industry survey. The identified practices frame the findings and structure the analysis in Section 4. The review is based on IEEE Xplore, a database that specializes in technical and engineering fields directly relevant to software engineering and computer science. Using a single database narrows the scope, simplifies the process, and avoids duplication while leveraging IEEE's role in developing and publishing standards. However, this approach excludes interdisciplinary perspectives and provides a narrower view of research. While IEEE includes strong work on quality and security, it was recognized that its coverage of privacy is less focused, although privacy-related research is expected to grow in the future.

We conducted a focused review of recent research (years 2020-2024) on quality, security, and privacy assurance in software development using IEEE Xplore. The primary search used the terms: ("document title": quality assurance) OR ("document title": security assurance) OR ("abstract": privacy assurance) AND ("document title": software) AND ("publication title": development), filtered by conferences, journals, and standards, yielding 210 results. Broader search terms such as "in software development" returned additional results: Privacy Assurance (56), Security Assurance (223), Quality Assurance* (504), Quality and Security Assurance* (124), Quality and Privacy Assurance* (17), and Security and Privacy Assurance* (47). After excluding articles that were not open access, not in English, or focused only on educational programs, 24 articles remained from the main search and 9 from the broader searches. These 33 articles were reviewed in full, and the results are summarized in Table 1.

The literature review (Table 1) shows that 80 % of the articles related to software assurance focus on quality assurance, and about 50 % of the articles deal with security assurance, so the importance of security assurance has already become relatively significant in the field of assurance. Of these, about 18 % deal only with security assurance, not combined with the quality assurance viewpoint. Surprisingly, only one article dealt with the privacy assurance aspect. This suggests that privacy assurance is not yet as strongly emphasized in assurance practices as quality or security. The importance of privacy assurance is expected to grow, as GDPR has only been in effect for six years and organizations are still integrating privacy assurance into their software development practices.

Based on the literature analysis, the research to date has largely addressed the aspects summarized in the definitions of quality, security and privacy assurance above. A very popular perspective (about 55 % of the papers) is to look at assurance in software development from the perspective of Agile/Waterfall methodology, the software development life cycle, and DevOps/DevSecOps. About 50 % of the articles dealt with testing (including security testing). The aspects covered in the articles varied. Several articles highlighted the importance of testing as part of assurance practices (Wong et al., 2022), how different testing techniques and practices are part of the development process, influence the outcome, and serve as quality control (Deshpande et al., 2023; Galindo-Francia & Auccahuasi, 2024; Jonathan et al., 2020). Articles also provide perspectives and differences between traditional waterfall and agile testing (Sinha & Das, 2021). Articles present ideas on

how to reduce the testing effort by increasing code review effort (Nakahara et al., 2021) or how to provide an overview of good practices in the area of testing of software solutions (Holjevac & Jakopec, 2022). Other topics include the impact of international standards on software quality and software testing (Zhao et al., 2021), and the use of automation (Gonen & Sawant, 2020). For security testing, perspectives emerge where cybersecurity should be built as an integral part of the quality assurance testing process (Haider & Bhatti, 2022), security testing methods (Khan et al., 2022), and testing practices such as penetration testing (Siang & Selvarajah, 2022) or practices where manual security testing should be performed (Arnold & Qu, 2020). The use of different tools, technologies or automation was also mentioned very often.

The third very popular approach in articles (about 36 %) is to offer proprietary solutions, frameworks, models, metrics, or approaches for developing or ensuring assurance. The following frameworks or models were proposed as solutions: the SQA-PAK model (Deshpande et al., 2023), a simulation model of software quality assurance (Nakahara et al., 2021), an adoptable guideline for mitigating the problems and implementing a quality framework (Shikta et al., 2021), the MAC methodology (Galindo-Francia & Auccahuasi, 2024), Scrumlity framework (Tona et al., 2021), proposals with standards (Zhao et al., 2021), and factors to balance control and autonomy to reduce security challenges in large-scale agile development. It was also suggested that the software quality assurance process should include additional steps such as code review and penetration testing for cybersecurity implementations (Haider & Bhatti, 2022). A conceptual framework for identifying the correctness, consistency, and completeness of security requirements (Janisar et al., 2023) and metrics for measuring the performance of regulatory attributes (Wagner & Ford, 2020) have been proposed.

The articles also cover various standards, models, frameworks, and metrics as part of software development or their impact as part of software development. For example, the ISO 9000, 25000, and 27000 series standards, CMMI, Six Sigma, Total Quality Management, EFQM, and Security Assurance Model were discussed (Deshpande et al., 2023; Khan et al., 2022; Mishra, 2023; Wong et al., 2022). Requirements and validation were also discussed in several articles (Cohen et al., 2021; Deshpande et al., 2023; Filipovikj et al., 2020; Hynninen & Jantunen, 2022; Nägele et al., 2023; Shikta et al., 2021). Security requirements have also been emphasized (Haider & Bhatti, 2022; Janisar et al., 2023; Khan et al., 2022; Mishra & Mustafa, 2020; Wagner & Ford, 2020).

Some of the articles discuss and highlight typical quality assurance practices, their success, or ways to improve practices. For example: (secure) coding and coding reviews (Haider & Bhatti, 2022; Hynninen & Jantunen, 2022; Jharko, 2021; Khan et al., 2022; Nakahara et al., 2021; Ramirez et al., 2020; Wong et al., 2022); design, analysis, and documentation of assurance and quality practices and plans (Deshpande et al., 2023; Janisar et al., 2023; Mishra & Mustafa, 2020; Shikta et al., 2021; Wong et al., 2022). In addition, the articles discussed process adoption and continuous process improvement, and the importance of roles and responsibilities as part of the process or software development (Deshpande et al., 2023; Galindo-Francia & Auccahuasi, 2024; Khan et al., 2022; Nägele et al., 2023; Shikta et al., 2021; Wong et al., 2022). Different metrics, key performance indicators, and service level agreements are discussed (Deshpande et al., 2023; Filipovikj et al., 2020; Galindo-Francia & Auccahuasi, 2024; Wagner & Ford, 2020; Wikantyasa et al., 2023; Wong et al., 2022), and corrective/preventive actions are also emerging in the context of validation and verification (Atoum et al., 2021; Deshpande et al., 2023; Jharko, 2021a; Khan et al., 2022; Nakahara et al., 2021; Shikta et al., 2021; Wong et al., 2022). The articles also covered some tools for assurance monitoring and management during development projects and management-related topics such as quality management system and risk management (Haider & Bhatti, 2022; Janisar et al., 2023; Jharko, 2021a; Khan et al., 2022; Khurana & Wassay, 2023; Mishra, 2023; Nägele et al., 2023; Niu et al., 2024; Wong et al., 2022). The importance of architecture and design as part of assurance is also noted in articles (Cohen et al., 2021; Janisar et al., 2023; Jonathan et al., 2020; Khan et al., 2022; Kharchenko et al., 2021; Wikantyasa et al., 2023). Only a few articles discuss the importance of training or maintaining and developing competencies (Deshpande et al., 2023; Hynninen & Jantunen, 2022; Khurana & Wassay, 2023; Nägele et al., 2023; Niu et al., 2024). Surprisingly, there were only two papers that addressed the entire supply chain and the consideration of software development vendors as part of the assurance practices (Filipovikj et al., 2020; Khan et al., 2022).

Table 1. Quality, security and privacy assurance literature review (years 2020-2024)

Articles	(Cyber) Security	Quality	Privacy	Standards, Models, Frameworks (ISO, CMMI, OWASP etc.), certification	(Secure) Coding and Code review, Software/code errors	Testing and testing methods	Security testing, penetration testing	Security risks/Risk mgmt	Security / Privacy requirements	Requirements and requirements validation	Tools and Techniques, automation	Quality Management system, Quality Process and Framework (EFQM)	Own Proposed solution/process/framework/metrics	Assurance/quality planning, Analysis, specifications, plans, documentation, checkpoints	Process implementation, Continuous development, Process improvements	Metrics, KPIs, SLA	Roles & Responsibilities	Verification & Validation	CAPA	Knowledge, training and education	Architecture and design	Agile/Waterfall methodology, software development lifecycle, DevOps	Vendor/Supply chain/outsourcing
Arnold & Qu (2020)	x	x				x	x															x	
Atoum et al. (2021)		x								x	x							x	x				
Cohen et al. (2021)	x								x	x											x	x	
Deshpande et al. (2023)		x		x		x				x			x	x	x	x			x	x			
Filipovikj et al. (2020)	x								x	x						x							x
Galindo-Francia & Auccahuasi (2024)		x		x		x					x		x		x	x							
Gonen & Sawant (2020)		x				x					x											x	
Haider & Bhatti (2022)	x	x		x	x	x	x		x	x	x	x	x										
Holjevac & Jakopec (2022)		x				x																	
Hynninen & Jantunen (2022)		x			x		x			x	x									x			
Janisar et al. (2023)	x	x						x	x				x	x							x		
Jharko (2021)		x			x							x	x						x			x	
Jonathan et al. (2020)		x				x					x										x		
Khan et al. (2022)	x	x		x	x	x		x	x						x			x			x	x	x
Kharchenko et al. (2021)		x																			x	x	
Khurana & Wassay (2023)	x							x												x		x	
Lochan (2021)		x																				x	
Lotz (2020)	x			x																		x	
Mishra & Mustafa (2020)	x								x					x									
Mishra (2023)		x		x								x											
Nakahara et al. (2021)		x			x	x							x					x				x	
Niu et al. (2024)	x	x						x												x		x	
Nägele et al (2023)	x	x						x		x			x				x			x		x	
Ramirez et al. (2020)	x			x	x		x															x	
Salahat et al. (2023)		x				x																	
Shikta et al. (2021)	x	x				x				x	x		x	x	x		x		x				
Siang & Selvarajah (2022)	x	x					x																
Sinha & Das (2021)		x				x																x	
Tona et al. (2021)		x									x		x				x					x	
Wagner & Ford (2020)	x	x	x						x				x			x						x	
Wikantyasa et al. (2023)	x	x											x			x					x	x	
Wong et al. (2022)		x		x	x	x					x	x		x	x	x	x	x	x			x	
Zhao et al. (2021)		x		x		x							x										

During the literature search, it became clear that IEEE does not yet contain many articles on privacy. This may be influenced by the technical focus of IEEE, and privacy-related studies are more likely to be found in other publishers. For example, the International Journal of Systems and Project Management has published privacy-related studies in recent years that focus on how the GDPR has affected consumers' knowledge, attitudes, and practices regarding their enhanced rights (Presthus & Sørum, 2024), the extent to which consumers are concerned about information privacy issues (Presthus & Sørum, 2019), and examining the violations and sanctions that have occurred following the implementation of the GDPR (Presthus & Sønslien, 2021). However, these articles do not focus on privacy aspects of software development and are therefore not fully within the scope of this literature review.

Previous research shows that quality practices are widely studied, and the topics covered are consistent with quality assurance definitions. The role and impact of security assurance is growing, requiring software developers to increasingly consider security assurance practices and requirements in their development work. Privacy assurance and related practices are not yet reflected in the articles, but their contribution is expected to be reflected in future assurance practices such as security assurance.

3. Research methods and survey implementation

This study uses a survey-based research methodology to gather insights from IT professionals on quality, security, and privacy assurance practices. The survey results are analyzed based on the most common assurance practices and focus areas as identified in the literature review. A key aspect is to look at the time taken to select assurance practices, i.e. how systematically and compliantly practices are selected and followed. The first step is to look at what typical quality, security, and privacy assurance practices are in place: plan, strategy, named process owner, writing requirements, defining targets, prioritizing requirements, defining acceptance criteria, and issue reporting. It then looks at the level/success of the assurance practices in use: documented processes, Corrective Action Preventive Action (CAPA), defined and documented roles and responsibilities, quality acceptance criteria, Key Performance Indicators (KPI), common rules, list of unsolved risks/defects, and a separate team for solving customer defects, and how well the companies implement the assurance practices in their day-to-day operations: software development practices, quality assurance, testing practices, security assurance, privacy assurance, technical debt management, risk management and reporting, clear Definition of Done (DoD), and use of standards and best practices. The decision to examine especially these areas is based on their recurring importance in the assurance literature as essential elements in maintaining software quality, security, and privacy assurance. In particular, the use of a separate team to address production issues presents a unique conflict: while a dedicated team can more directly address customer-side issues, this structure can also create communication gaps between the development and maintenance teams, potentially hindering awareness of production challenges.

The industry survey targeted Finnish software engineers and developers, using both an open online link (promoted by the University of Turku Alumni, FISMA and Sytyke ry) and direct email invitations to addresses publicly available on the Finnish Software Entrepreneurs Association website, following ethical and privacy guidelines by contacting only those companies with publicly listed email addresses or formats. The survey was conducted from October 2019 to February 2020, with reminders sent in November and December, and received 88 valid responses, about 71 % from the public link. Respondents were highly experienced, with 46% having more than 16 years of IT experience (3 % <1 year, 22 % 1-5 years, 16 % 6-10 years, 13 % 11-15 years, 46 % >16 years), and held roles in development, design, architecture, testing, and management. Respondents represented a range of company sizes, with 44 % from large companies (10 % <10 employees, 22 % 10-50 employees, 13 % 51-100 employees, 9 % 101-250 employees, and 44 % >250 employees). Most companies served the private sector (69 %) and provided complete software systems, consulting services, and SaaS products. The survey successfully reached experienced professionals in various roles, probably facilitated by FiSMA and Sytyke Ry's promotion of the survey, reflecting the Finnish software industry landscape. (Majanoja & Hakkala, 2023; Majanoja et al., 2023).

4. Results and analysis

This section presents the findings of the study, organized around the key research questions. The findings are divided into three sections. First, in Section 4.1, we examine the following order of practices, while Section 4.2 provides an overview of systemicity and compliance in selecting and following assurance practices. Next, we focus on day-to-day assurance practices, as Section 4.3 examines the implemented assurance practices (focus practices selected based on the literature review) and provides post-delivery findings focused on the assurance perspective. Taken together, these findings provide an overview of the role of assurance practices in day-to-day software development operations.

4.1 The order of adherence to practices

The order of adherence to practices is as follows: first is software development practices, and second is quality. Third is security, and fourth is privacy practices. The fifth followed practice is testing, which is partly combined with quality and software development practices. The least followed practice is technical debt.

When analyzing the results by company size (Fig. 1), small companies view quality and security as the most critical assurance practices.

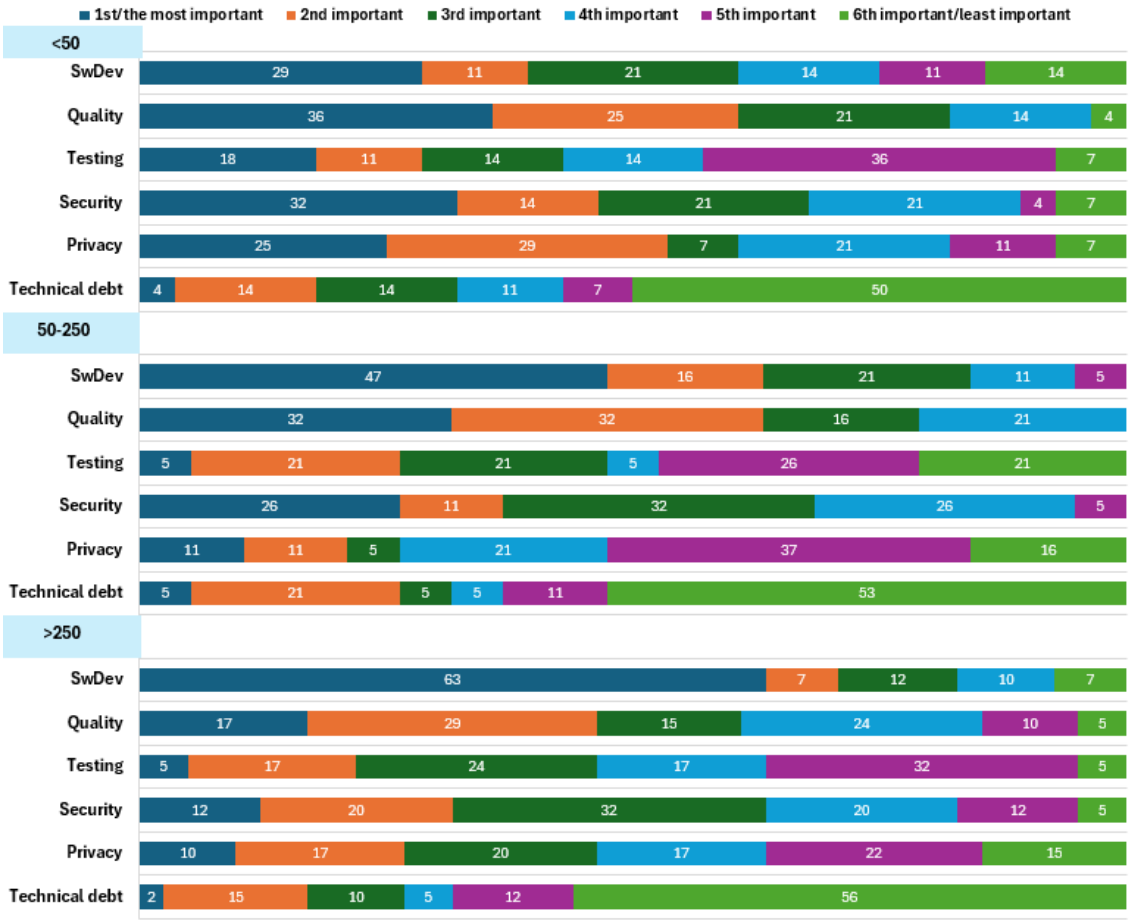

Fig. 1. Order of practice adoption in daily operations by company size (%)

Software development (SwDev) is a high priority, but slightly behind quality and security. Testing and privacy show more variability, with testing often considered a lower priority than the others. Technical debt is consistently ranked as the least important practice. For medium-sized companies, SwDev and quality are the most important practices to follow. This suggests that ensuring the development process and maintaining quality are critical areas of focus for developers. Security also ranks high, with a significant number of developers ranking it as their first or third most important priority. Testing and privacy are more likely to be seen as lower priorities. This may indicate that testing practices are seen as complementary to software development and quality practices. Technical debt is consistently ranked as the lowest priority. This suggests that while technical debt management is recognized, it is not seen as urgent compared to other assurance practices.

In large companies (Fig. 1), SwDev practices are by far the most important. Quality is usually ranked second, indicating its importance, but less so compared to software development. Security and testing show a strong presence in 2nd or 3rd place and are generally considered medium priority practices. Privacy is typically ranked lower, with the majority placing it in 3rd, 4th or 5th place, indicating that while important, it is often deprioritized compared to security or testing. Technical debt is overwhelmingly ranked last by the majority, indicating that it is generally considered to be the least important practice in this context.

4.2 The level of systemicity of the practices and the compliance with the regulations

Analyzing the results (Table 2), about 40 % of small companies (<50 employees) use the same practices for all projects or sometimes make some changes, especially in software development (60 %) and quality (70 %), while testing practices are often defined on a project or ad hoc basis (47 %). Many small companies take an ad hoc approach to managing technical debt (32 %) and rely on ad hoc or project-specific approaches for data privacy (34 %) and security (43 %).

Medium-sized companies (51-250 employees) tend to use the same set of practices or make occasional adjustments, especially for security (69 %) and privacy (63 %). Software development practices are fairly consistent across 53 % of projects, and 53 % of projects follow a standard testing approach, although 32 % still use an ad hoc testing approach. Technical debt management is largely ad hoc (53 %), indicating variability or a lack of standardized processes.

In large companies (>250 employees), 60-70 % rely on predefined practices with occasional adjustments. Testing and quality practices are more adaptable, with 39 % and 24 %, respectively, sometimes changing. Security practices are either standard (39 %) or ad hoc (27 %), and while 32 % manage technical debt ad hoc, 58 % use a standard approach with some customization.

Based on Table 2, small companies rely heavily on project-level accountability, particularly for managing technical debt (64 %), software development (54 %), quality (54 %), and testing (54 %), suggesting a case-by-case approach to compliance. Medium-sized companies also emphasize compliance at the project level, particularly in testing (63 %), software development (42 %), and security (42 %). At large companies, project teams are most often responsible for compliance in areas such as technical debt (61 %), testing (61 %), and software development (51 %), reflecting an autonomy-oriented approach.

Privacy and security are more heavily regulated across all company sizes, with large companies more likely to adhere to strict processes, particularly in the area of quality (41 %). This structured approach is less evident in smaller companies, where formal compliance processes are less common, likely due to resource constraints and the need for agility. However, project-based compliance remains the dominant model across all company sizes.

Table 2. Systematic implementation of practices and compliance with regulations (%)

Systemic approach in selecting methods and practices					Project compliance with regulations and policies			
Practice	Always same	Some-times changing	Systemati-cally defined on a per-project basis	Adhoc approach and adapting	Not regulated or described	Project's own responsi-bility	Broadly performed through regulations/ processes	Strictly in accordance with regulations/ processes
Small-size companies: <50								
SwDev	36	25	11	29	29	54	11	7
Quality	32	36	11	21	25	54	14	7
Testing	36	18	29	18	18	54	21	7
Security	29	29	29	14	29	36	18	18
Privacy	36	21	32	11	25	29	25	21
Technical debt	32	21	14	32	32	64	4	0
Medium-size companies: 51–250								
SwDev	21	32	21	26	5	42	32	21
Quality	16	47	21	16	0	47	26	26
Testing	21	32	16	32	0	63	16	21
Security	32	37	16	16	16	42	16	26
Privacy	42	21	11	26	21	42	11	26
Technical debt	11	21	16	53	42	42	5	11
Large-size companies: >250								
SwDev	27	39	10	24	7	51	29	12
Quality	37	24	20	20	2	46	41	10
Testing	27	39	17	17	5	61	27	7
Security	39	24	10	27	10	39	34	17
Privacy	41	24	17	17	10	32	32	27
Technical debt	34	24	10	32	24	61	12	2

The analysis shows that company size has a significant impact on the use of the same standard set of practices versus ad hoc methods, especially in security practices. Larger organizations tend to be more process-driven, with established work practices that allow for some flexibility. In contrast, smaller companies also rely on ad hoc approaches, likely due to limited resources and the need to remain responsive in a changing environment. Regulatory compliance is typically the responsibility of the project, regardless of the size of the organization. These findings suggest that projects do not spend much time defining the set of practices they use or how they approach compliance on a day-to-day basis. Roughly speaking, they simply follow the same predefined set of practices regardless of the project or customer.

4.3 Success in implementing and performing assurance practices in daily operations

Implemented assurance practices. The results in Fig. 2 show significant differences in the use of assurance practices (Not Used, Quality, Security, and Privacy). Across all company sizes, there is a relatively high rate of "Not Used" in areas such as defining KPIs and acceptance criteria, particularly among large companies, where 68 % do not use KPIs and 56 % do not use acceptance criteria. This indicates a widespread lack of consistent performance metrics or benchmarks.

In terms of quality practices implemented (Fig. 2), companies tend to prioritize quality practices over security and privacy, with more consistency in the use of defined acceptance criteria and planning, especially in small and medium-sized companies. However, security practices show greater variability and reliance on ad hoc methods. In small companies, 64

% manage security without formal prioritization of requirements or definition of KPIs, reflecting a less structured approach. Privacy practices are similarly inconsistent, with a significant proportion of companies using ad hoc or project-based methods, although large companies use slightly more formal processes for privacy than for security. It appears that while quality assurance practices are somewhat more formalized, security and privacy tend to be managed more inconsistently, particularly in smaller organizations, indicating potential vulnerability in these critical areas.

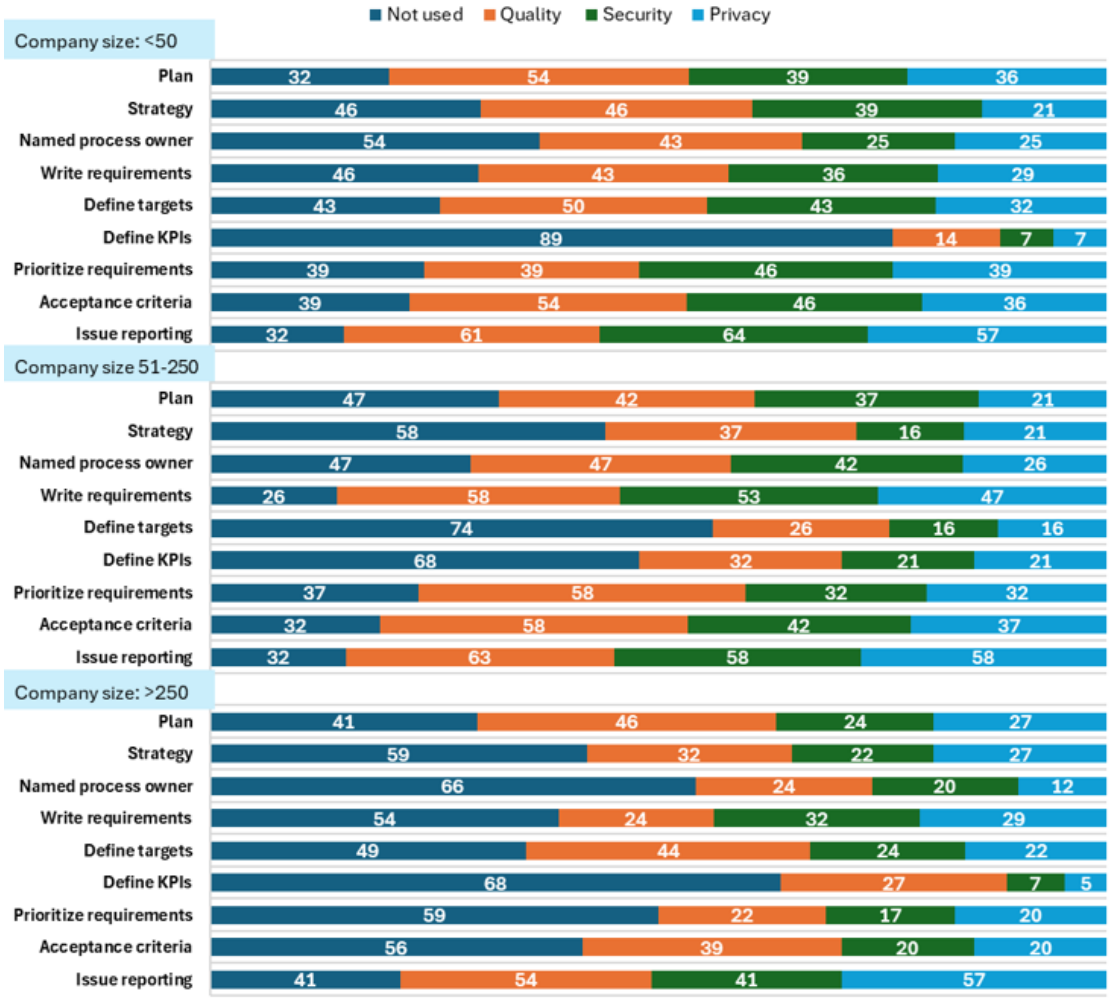

Fig. 2. Implemented quality, security and privacy assurance practices per company size (%) to daily operation

Success of assurance practices. In Table 3 and Fig. 3, we have selected common assurance practices and analyzed how well they are implemented in daily operations. The analysis (Table 3) shows that CAPA practices in particular are underrepresented among assurance practices across all company sizes. This is somewhat surprising, given that the purpose of CAPA is to gather and analyze information, identify and investigate product and quality issues, and implement effective corrective and preventive actions to prevent recurrence. In addition, the central focus of CAPA is to verify and validate the actions taken, communicate the CAPA action activities to responsible parties, provide relevant information for management review, and document these activities to prevent their recurrence and prevent or minimize failures (Majanoja et al., 2017).

The other underrepresented practice is defining and reporting KPIs (Table 3). This is particularly evident in small companies (57 % do not use), but larger companies also report challenges in defining and measuring KPIs. The use of dedicated team to solve production problems is intentionally ambiguous. Smaller companies do not use it, which is understandable given resource constraints, but at the same time it is good that developers are also responsible for production issues. Larger companies have the resources to separate development and maintenance. However, in this case, production problems and challenges may not reach the development team, and as a result, CAPA practices may not work efficiently. Surprisingly, companies of all sizes report varying degrees of success in managing risk and defects. This may indicate that companies have challenges in implementing well-managed risk management practices.

Table 3. Success of assurance practices in use per company size (%)

Level of use	Documented processes	CAPA	Roles and responsibilities	Quality acceptance criteria	KPI	Common rules	List of unsolved risks/defects	Separate team solving customer defects
Small-size companies: <50								
Not used	14	43	18	21	57	11	14	61
Poorly	21	18	18	18	25	11	11	11
Fairly	29	25	29	25	4	14	14	21
Well	21	7	25	18	4	21	18	4
Very well	11	4	11	18	11	18	18	4
Excellently	4	4	0	0	0	25	25	0
Medium-size companies: 51–250								
Not used	5	32	21	21	32	0	5	32
Poorly	11	16	11	26	32	11	21	5
Fairly	21	16	16	21	26	16	26	11
Well	26	26	21	11	0	37	21	26
Very well	32	5	26	11	11	21	26	26
Excellently	5	5	5	11	0	16	0	0
Large-size companies: >250								
Not used	5	24	12	15	20	7	10	29
Poorly	15	20	2	15	12	12	12	10
Fairly	29	17	29	24	32	22	32	22
Well	32	24	29	37	17	24	22	17
Very well	12	7	20	5	12	27	17	15
Excellently	7	7	7	5	7	7	7	7

Defined and documented roles and responsibilities (Table 3) are critical to CAPA and assurance practices, with clear differences based on company size. Small companies often lack well-defined roles, while larger companies tend to document them more thoroughly. Similarly, documented processes follow this trend: as companies grow, the need for clear documentation increases. In addition to serving as a learning and communication tool, well-documented processes are essential for quality assurance audits and validation procedures, which require named process owners.

While companies of all sizes report that they perform quality assurance well, the results show varying degrees of success in defining quality assurance criteria (Table 3). Interestingly, even large companies report similar challenges to smaller ones, suggesting that defining and managing quality requirements remains difficult across the IT industry. However, all companies report strong adherence to common rules and practices in their daily software development operations.

When analyzing how organizations perform assurance practices on a daily basis, in small companies (Fig. 3) practices such as risk management, technical debt management, and software development practices tend to be in the "well" to "excellent" range, but still have a noticeable portion of respondents indicating low application. Clear DoD is a well-implemented practice, while privacy assurance and security assurance are moderately practiced. Testing and quality assurance show room for improvement, with more developers reporting moderate compliance. For medium-size companies, software development practices, security and quality assurance show strong performance, with many IT professionals rating them highly, indicating focus on these areas. Risk management and technical debt management also perform well but show much more variability. Privacy assurance and testing practices have inconsistent results, suggesting that these areas are not consistently prioritized. Use of standards & best practices and clear DoD show moderate performance.

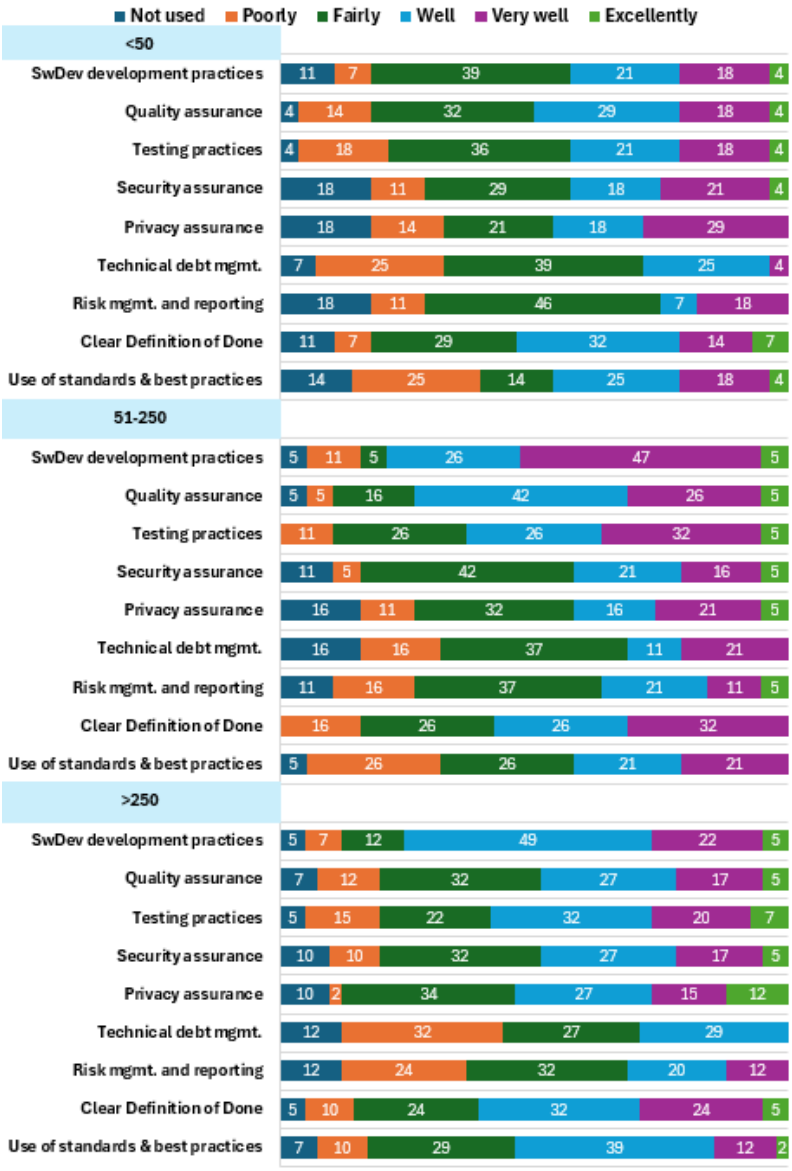

Fig. 3. Performing assurance practices on a daily operation

In large companies (Fig. 3), software development practices, clear DoD, and quality assurance are well practiced. Security and privacy assurance are moderately well practiced, as are testing and the use of standards and best practices, but there is some variability. Technical debt management and risk management show weaker performance. This suggests that these practices are less of a priority or are not implemented or managed efficiently.

Verification practices. Verification is one of the most important assurance practices (Fig. 4). When analyzing the "Not used" practices, small organizations rarely use threat assessments and external audits, and have limited use of internal audits, observations, and interviews. Medium-sized organizations also avoid threat assessments and architectural reviews, with moderate avoidance of external audits and observations. Large organizations rarely use external audits, which is a surprising result. They also rarely use observations and threat assessments, architectural reviews and interviews.

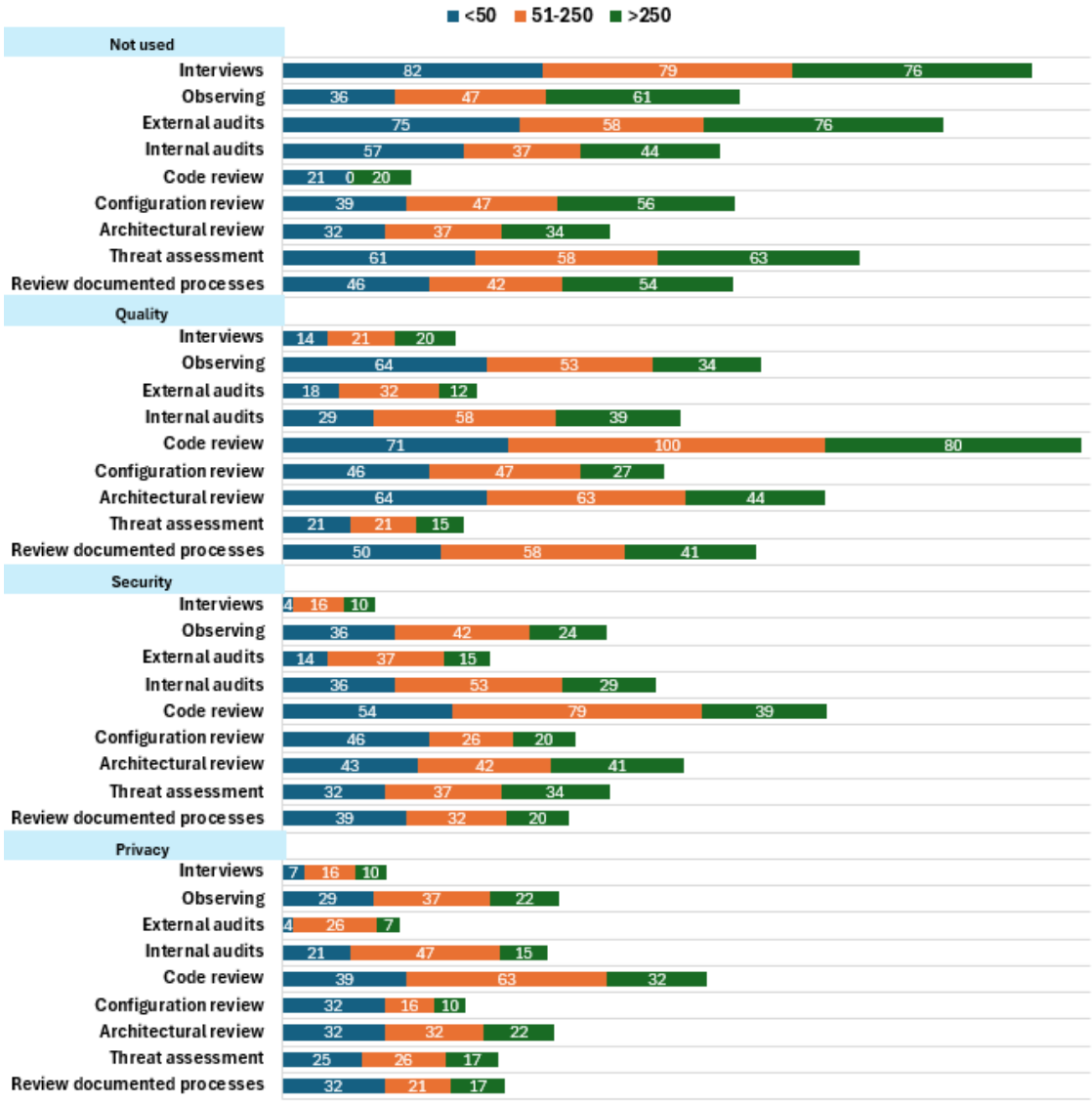

Fig. 4. Verification techniques used during a project for quality, security and privacy (%)

Small companies (Fig. 4) focus primarily on code and architecture reviews and internal audits. For quality assurance, internal audits and observations are common, while architecture reviews and review documented processes are also high priorities. Configuration reviews and code reviews are fairly common, with moderate use of external audits and threat assessments. In security, documented process reviews, threat assessments, architectural reviews, and configuration reviews are well used, showing a strong focus on security design. Internal audits and code reviews are common, but external audits and observations are less common. For privacy, review documented processes and architectural reviews are widely used, with internal audits and code reviews indicating strong internal controls. Threat assessments and configuration reviews are used moderately, while external audits and observations are less common. Interviews are rarely used as a verification practice across all three assurance areas (quality, security, and privacy).

Medium-sized companies (Fig. 4) prioritize internal audits and architectural reviews for quality assurance. Documented process reviews, external audits, observations, and configuration reviews are also common. For security, internal audits and threat assessments are highly prioritized, and external audits and observations are common. Code reviews are less emphasized, and interviews are rarely used. For privacy, internal audits and code reviews are main practices, while observations, review of documented processes, and threat assessments are less emphasized. External audits are used moderately, and interviews are a lesser verification practice for quality, security, and privacy.

For quality, large companies (Fig. 4) consistently use internal audits, architectural reviews, and review of documented processes, while surprisingly code reviews, threat assessments, and observations are less emphasized. For security, internal audits and threat assessments are commonly used. External audits, configuration reviews and observations are used moderately. For privacy, internal audits and architectural reviews are used, but less consistently than in smaller organizations, while review documented processes and threat assessments show limited use. External audits and observations are infrequent, indicating a lower priority, and interviews are used the least, similar to smaller organizations.

Post-delivery challenges. Analysis of the post-delivery results for all company sizes (Fig. 5) shows that many respondents chose "no answer," likely due to lack of knowledge (large companies represented over 40 % of respondents), especially in larger companies where maintenance teams handle post-delivery issues.

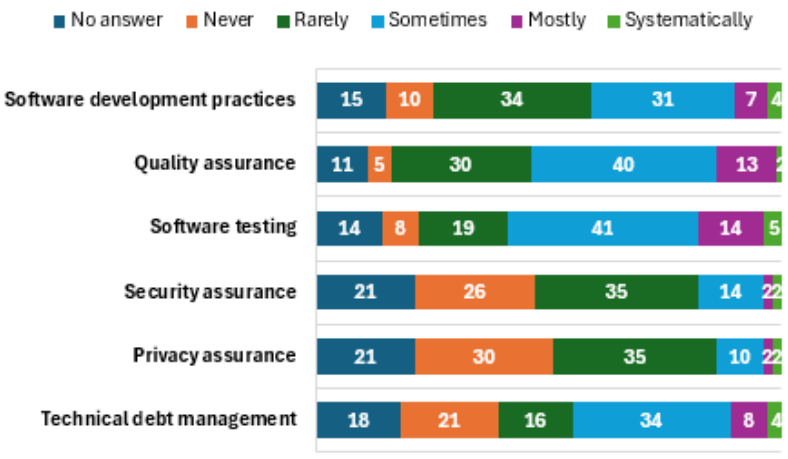

Fig. 5. Post-delivery challenges (%)

The results in Fig. 5 and the open-ended responses show that post-deployment challenges are primarily related to quality assurance and testing (60 %), indicating that gaps in these areas/assurance practices during development lead to problems in production. Software development practices also contribute to post-delivery problems (40 %), indicating that deficiencies in these practices affect production stability. In addition, technical debt is identified as a source of production problems, highlighting the need for proactive management. In particular, security and privacy issues appear post-delivery in over 50 % of cases, suggesting that these areas may lack robust assurance practices during development, which are often addressed reactively ("band-aid" solutions) when problems arise. This could indicate insufficient proactive measures or a lack of prioritization of security and privacy during the development phase. Overall, these findings suggest that insufficient focus on quality, security, and privacy assurance during development leads to many post-delivery challenges.

5. Discussion

This section discusses the results of the study and is divided into five sections. First, Section 5.1 discusses assurance practices in day-to-day operations. Section 5.2 focuses on the challenge of using fixed practices in a continuous improvement environment. Section 5.3 examines the role of education and universities in integrating assurance practices into educational content and shaping culture. Section 5.4 provides theoretical and managerial perspectives on this study, and Section 5.5 outlines the limitations of the study.

5.1 Assurance practices in daily operations

To deliver business value and retain customers, organizations must be able to produce high-quality, secure, and privacy-compliant products. Assurance practices play an important role in ensuring that these requirements are successfully addressed as part of the software development lifecycle. However, the assurance practice survey shows that companies have not fully implemented assurance practices to realize the full benefits of well-implemented assurance management. The survey results show that companies are primarily focused on software development and ensuring that the necessary requirements are in place. This focus is critical because the business demands results and products that create value for both the business and its customers. Achieving this value is only possible through effective software development, namely delivering high quality products. This is clearly reflected in the results, where the most important practice identified is adherence to software development practices.

When analyzing the success of companies in implementing assurance practices, most of the focus has been on implementing quality and testing practices as part of daily operations. Our findings on the importance of quality assurance are consistent with Deshpande et al. (2023), who also emphasize its critical role in daily operations. In contrast, security and privacy practices were quite inconsistent, giving the impression that these aspects are addressed when something negative happens. Thus, unlike Wong et al. (2022) who found that security practices are well integrated, our study reveals inconsistencies in the implementation of security practices. Our findings suggest that privacy assurance practices are still in their infancy, which is consistent with the observations of Presthus & Sørum (2024). This may indicate that security and privacy, in particular, are considered separate from software development, and that implementing these aspects in daily operations is more challenging than quality practices. However, many security-related tests rely heavily on manual testing, and therefore security-related automation does not provide benefits as quickly as quality-related test automation. A very typical solution for companies is to always use the same set of predefined practices, and these same practices are used regardless of the customer, the project, or the situation. Companies also tend to leave compliance as a project responsibility.

However, if you stick to one method for a long time, you will not improve. And this goes against the current de facto approach in software development, which is to use agile practices. Our study extends the work of Haider & Bhatti (2022) by highlighting the challenges of integrating security practices in agile environments. Agility thrives on change, uses iterative and flexible processes, and focuses on continuous improvement that evolves as the project progresses. The core

of agile practices is to reflect on the work, practices, and processes after each iteration and identify ways to improve. If teams always use the same set of practices, this process becomes ineffective because they do not adjust their approach based on lessons learned or evolving project needs. Sticking to a fixed set of practices inhibits the ability to adapt to changing needs and contexts, making it fundamentally incompatible with the agile approach. There is no one-size-fits-all assurance practice. Different industries, regulatory requirements, and project types may require different assurance practices.

An important aspect that is often overlooked is the lack of alignment between organizational priorities and the actual implementation of assurance practices. As noted in this study, while organizations recognize the importance of assurance practices, there is a gap between their recognition and practical integration. This gap may be due to competing priorities, such as the pressure to deliver results quickly or meet tight deadlines, which often take priority over embedding comprehensive assurance practices in day-to-day operations. The organizational mindset may still treat assurance practices as a compliance requirement rather than a strategic enabler of long-term business value. This mindset not only limits the potential benefits of assurance practices, but also hinders innovation, as teams may resist change or fail to adopt more effective practices that could address evolving needs. For organizations, addressing this strategic misalignment is critical to transforming assurance from a peripheral activity to a core element of software development.

5.2 The challenge of fixed practices vs. the need for continuous improvement

The study shows that the importance of effective and well-managed assurance practices in software development is well recognized and understood. So why, despite the best efforts of IT professionals and companies, do assurance practices not become an integral part of daily operations? Why do they remain as separate activities? Several factors are likely contributing to this challenge.

There may be resistance to implementing comprehensive assurance practices because they are not seen as critical by senior management. Without management support, it is difficult to implement assurance practices consistently. This is also often justified by the inflexibility of assurance practices, which are seen as development bottlenecks that limit product implementation by introducing external constraints and requirements (such as various compliance requirements) through often structured, process-heavy approaches. This tension between agile iteration and formal assurance can make it difficult to consistently embed comprehensive quality, security, and privacy assurance practices throughout the development process. Documenting processes for assurance practices can be a pain point, especially in large organizations. The need for detailed process documentation and compliance checks can feel overwhelming, and some teams may neglect it in favor of shorter, more agile workflows.

Assurance practices are often seen as a separate task that adds cost, and paying attention to assurance issues takes resources and prevents things from being done "easily" or efficiently because of (perceived) marginal problems. As a result, assurance practices are seen as an added cost rather than a value-added activity, often deprioritized by developers to meet tight deadlines or budgets. Development teams focus on adding new features and meeting deadlines, while assurance teams focus on risk mitigation, defect detection, and regulatory compliance. These priorities can conflict, especially when development teams are under pressure to release quickly.

Quality assurance practices such as comprehensive testing, security testing, and compliance checks require significant time and resources, which can slow down the development process. While Nakahara et al. (2021) suggest that code review can reduce testing effort, our findings indicate that comprehensive testing remains essential for quality assurance. Effective quality and security assurance requires skilled professionals, which can be a challenge, especially in smaller organizations. For example, new vulnerabilities and threats emerge regularly, making it challenging to maintain security throughout the software lifecycle. With the rise of cloud computing, microservices, and third-party integrations (e.g., leveraging digital supply chains), ensuring quality and security has become more complex. Assurance practices must account for multiple platforms, technologies, and environments, making it difficult to maintain consistency and coverage.

The growing importance of security assurance is consistent with Haider & Bhatti (2022), who emphasize the integration of cybersecurity into quality assurance testing. Unlike Khan et al. (2022), who examine various security testing methodologies, this study highlights inconsistencies in implementation of security practices, suggesting that companies are taking a reactive rather than proactive approach.

This study identified quality assurance as the most emphasized practice, consistent with Deshpande et al. (2023), who emphasize its critical role in day-to-day operations. While Wong et al. (2022) emphasize the importance of testing techniques, our findings suggest that companies often adhere to predefined practices rather than tailoring them to specific project needs. Using the same practices helps companies standardize processes across the organization, making it easier to manage and maintain assurance requirements. It is also easier to sell and convince customers of the company's assurance practices. This approach creates a predictable and repeatable workflow to follow. It also saves time because customization requires additional effort in planning, training, and adapting practices. Using familiar practices can also be seen as a form of risk management. The practices may not be perfect, but they have been tested and refined over time. These familiar practices provide a safety net. When an organization has experienced success with a set of practices, it is less likely to experiment with new methods that could lead to errors or disruptions in workflow. And when a set of practices is seen as a working approach, there is no rush to add or improve it, even if companies are not able to reap all the benefits of well-implemented assurance, especially if they do not see an immediate benefit from changing or expanding current practices. For example, customers are unlikely to pay for reporting on supplier quality metrics.

5.3 Educational shift for assurance practices

Given the challenges in successfully implementing assurance practices, what can be done to address them? Often it is a cultural aspect that needs to be changed, and that change starts with IT education. Different IT practices and viewpoints are indoctrinated during IT studies, and if assurance practices are seen as separate silos rather than core software development skills and practices, change is almost impossible. Improving software development practices requires early and integrated approaches to assurance that are progressively introduced in courses.

Seamless integration of quality, security and privacy into the software development lifecycle means incorporating these aspects from the beginning and using automated tools for continuous testing and monitoring. An example is the "Shift Left" approach, where the goal is to integrate security early in the development lifecycle (security reviews, threat modeling and vulnerability scanning during design; TDD and CI/CD approaches to ensure quality is built in). This study's identification of the role of education in integrating security practices is consistent with the Shift Left approach, which emphasizes early and proactive security measures. Essential skills include automated testing and CI (TDD, automated security scanning: Static Application Security Testing, Dynamic Application Security Testing, and Software Composition Analysis), and embedding security in agile practices, such as learning how to embed security practices in agile development practices like DevSecOps. Or learn that effective assurance involves Sprint planning with quality, security, and privacy in mind, using a DoD that requires unit testing, code reviews, security checks, and privacy compliance before completion. It is important to learn to use metrics to track the effectiveness of assurance practices. If metrics are not introduced and practiced during studies, it is difficult to understand their value.

While universities play a key role in preparing students for software development, the curriculum often lags behind industry needs, especially in areas such as security and privacy. This mismatch between academic preparation and real-world demands leaves graduates unprepared to address security or privacy challenges in real-world scenarios. The lack of a consistent framework that incorporates both theory and industry needs for evaluating the success of integrating quality, security, and privacy into educational content limits the ability to identify gaps and improve integration. Although integration of these topics ensures that graduates are better prepared for real-world demands, many IT curricula still treat privacy and security as peripheral or advanced topics rather than core competencies. This separation reinforces the misconception that quality, security and privacy practices are optional or secondary to software development. However, the rapid evolution

of threats, technologies, and regulations is creating a gap between academic knowledge and practical requirements, making it difficult for universities to keep their programs aligned with real-world needs. Another challenge is balancing theoretical knowledge with practical experience. While students may learn about general security or privacy principles or compliance frameworks, they often lack exposure to practical tools such as automated testing, threat modeling, or CI/CD pipelines. This study highlights the importance of practical tools and metrics, yet the literature suggests that universities often lag behind industry needs in these areas. The interdisciplinary nature of security and privacy at the intersection of law, ethics, and business is often overlooked in IT programs, limiting students' ability to address complex, multifaceted challenges.

This raises the question of how well university programming and software development courses incorporate assurance practices. It would be valuable to examine the extent to which these courses prepare students to view quality, security, and privacy as integral to the development process, rather than as isolated silos or checklists. In addition, exploring faculty perspectives could reveal whether certain attitudes or traditional approaches may unintentionally hinder the shift toward viewing assurance as a core component of software development. Understanding these dynamics could provide insights into how educational programs might evolve to produce graduates who are ready to seamlessly integrate assurance into their work and thereby drive change in organizational practices.

5.4 Theoretical and Managerial perspectives

Theoretical perspective. This study contributes to the theoretical understanding of assurance practices in the context of software development by exploring how quality, security, and privacy frameworks are adopted, applied, and performed throughout the software development lifecycle. By surveying 88 software professionals in Finland, the research provides empirical insights into the challenges and inconsistencies developers face in integrating assurance practices. The findings highlight the tension between the flexibility of agile methodologies and the rigidity of traditional assurance practices, emphasizing the need for a more integrated and adaptive approach. The study also draws attention to the underemphasis on security and privacy assurance compared to quality practices, suggesting that these areas are often addressed reactively rather than proactively. This research extends the theoretical discourse on the role of assurance in software engineering by advocating a seamless approach that incorporates quality, security, and privacy assurance from the early stages of development, rather than as an isolated or final activity. It also contributes to the conversation about the educational gaps in preparing software developers to integrate assurance practices as a core competency.

This study revealed practices that were consistent with the literature review. As shown in Table 1, quality remains the highest priority, with 80 % of the reviewed papers and the survey results of this paper emphasizing its importance in daily operations. Security follows, addressed in 50 % of the literature and reflected in operational practices in the survey, although it has not yet reached the same level of maturity as quality. Privacy practices were still in the early stages, with efforts focused on defining and establishing daily practices, consistent with findings in the literature. However, privacy practices are expected to evolve rapidly and reach the same level of importance as security.

According to Alvesson & Sandberg (2011), Avison & Malaurent (2014) and Gregor (2006), theory building involves a systematic approach that integrates problematization, practical relevance, and structured frameworks. Alvesson and Sandberg (2011) emphasize the problematization of assumptions in existing research in order to generate innovative questions. Avison & Malaurent (2014) argue for balancing theoretical contributions with real-world applicability to ensure practical impact. Gregor's (2006) taxonomy provides a framework for theory building, whether for analysis, explanation, prediction, or design and action. The findings of this study problematize the reactive nature of quality, security, and privacy assurance and highlight the tension between the need for the flexibility of agile methodologies and the rigidity of traditional assurance frameworks. This research challenges established assumptions and supports an integrated, proactive approach to assurance that is consistent with agile principles and modern development needs. Balancing theoretical rigor with practical relevance, this research extends the theoretical discourse by providing actionable insights, such as embedding

assurance practices from the earliest stages of development and addressing educational gaps that prevent the seamless integration of these practices as core competencies. This study contributes by proposing adaptive and integrated assurance practices that enable software professionals to meet current challenges. By bridging theoretical concepts with practical implications, this research provides both a deeper understanding of assurance practices and concrete ways to improve them in agile and iterative development contexts.

Managerial perspective. From a managerial perspective, this study provides valuable insights for organizations seeking to improve their software development practices by embedding assurance methodologies into daily operations. The findings suggest that while organizations recognize the importance of quality assurance, many still struggle to implement security and privacy practices effectively, often treating them as secondary concerns or addressing them reactively when problems arise. These findings aim to bridge the gap between theoretical knowledge and practical application by fostering a culture of continuous improvement where assurance practices are seamlessly integrated into the development process. The study highlights the need for a balanced approach that combines the flexibility of agile methodologies with the structured, process-driven requirements of assurance. For organizations struggling to adopt comprehensive assurance practices, the research suggests the use of automated tools and continuous monitoring to reduce manual effort and streamline testing procedures. In addition, the study underscores the importance of fostering strong management support for assurance practices and aligning these practices with business goals to ensure that they are not perceived as obstacles, but as value-added activities that are essential to producing secure, high-quality products. The study also highlights the need to include assurance practices as part of educational content to make quality, security and assurance practices core IT competencies.

Organizations need to view assurance practices as integral components of business value creation, rather than as isolated tasks. This study underscores the importance of embedding assurance into daily operations by aligning it with strategic objectives to ensure that it supports both organizational goals and customer expectations. Organizations should take a proactive approach to security and privacy, integrating it as a core element of quality assurance rather than a reactive afterthought. By leveraging automated tools, continuous monitoring, and agile-compatible assurance practices, organizations can reduce manual effort and increase efficiency. Strong leadership support is critical to fostering a culture where assurance is seen as a strategic enabler, not a hindrance. Integrating assurance practices into education ensures that future IT professionals are equipped with the skills necessary to seamlessly integrate quality, security, and privacy into software development workflows, bridging the gap between academia and industry needs.

5.5 Limitations

Naturally, it is recognized that this research has limitations. We recognize that the typical literature review is conducted prior to actual research, such as a survey (as in Webster and Watson, 2002). Because quality assurance practices are well known and well researched, we conducted the literature review using the most recent research available at the time of the survey data analysis, rather than limiting the timeframe to before 2020. Thus, the findings and categorizations from the most recent study were used to structure the review of the survey results and the analysis. The data and examples are drawn from surveys with limited response. All results are based on data from Finland, and country-specific differences may affect applicability. Several measures were taken in the preparation, design, and implementation of the survey and in the analysis of the data to reduce or avoid threats to the validity of the study. The survey instrument was an open and anonymous questionnaire, which posed challenges such as the inability to identify the respondents' place of residence, institution, company or project. Recognizing that the respondents have the same or very similar educational background, possibly even having graduated from the same institutions; the respondents work in the software development industry, share their experiences, and draw from the same practical influences. The target group to which the survey was sent and advertised was selected from an extensive list of Finnish software companies and well-known actors in the industry. The respondents were involved in different activities in software development projects. Thus, generalizing the results from the observed sample to all IT professionals would lead to potentially misleading results, and therefore descriptive statistics

were used to present the results in order to avoid issues in external validity. The results provide valuable insights that can be generalized to similar contexts and inform improvements in assurance practices in both industry and academia.

6. Conclusion

The purpose of this paper was to examine the order in which software developers follow assurance practices, how systematically they use the same set of practices, the compliance with regulations, and the success of implementing quality, security, and privacy assurance in daily operations. The results show that IT professionals place the greatest emphasis on software development practices, recognizing the role of quality assurance and testing in producing quality products. However, security and privacy practices appear to be less consistently implemented, suggesting that they are often treated as separate from daily practices. This underscores the need to integrate assurance practices more closely into routine operations, rather than as isolated tasks that are addressed only reactively. Addressing this gap starts with IT education, where curricula could evolve to introduce frameworks that embed quality, security, and privacy at the core of software development. This could prompt universities to evaluate their course content and provide students with a stronger foundation in assurance practices.

References

Agile Alliance. (2024). What is Agile? | Agile 101 | Agile Alliance. https://www.agilealliance.org/agile101/. Accessed 11.11.2024.

Alvesson, M., & Sandberg, J. (2011). Generating Research Questions Through Problematization. Academy of Management Review, 36(2), 247–271. https://doi.org/10.5465/AMR.2009.0188

Arkin, B., Stender, S., & McGraw, G. (2005). Software penetration testing. IEEE Security and Privacy, 3(1), 84–87. https://doi.org/10.1109/MSP.2005.23

Arnold, B., & Qu, Y. (2020). Detecting Software Security Vulnerability during an Agile Development by Testing the Changes to the Security Posture of Software Systems. Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020, 1743–1748. https://doi.org/10.1109/CSCI51800.2020.00323

Atoum, I., Baklizi, M. K., Alsmadi, I., Otoom, A. A., Alhersh, T., Ababneh, J., Almalki, J., & Alshahrani, S. M. (2021). Challenges of Software Requirements Quality Assurance and Validation: A Systematic Literature Review. IEEE Access, 9, 137613–137634. https://doi.org/10.1109/ACCESS.2021.3117989

Avison, D., & Malaurent, J. (2014). Is Theory King?: Questioning the Theory Fetish in Information Systems. Journal of Information Technology, 29(4), 327–336. https://doi.org/10.1057/JIT.2014.8

BBC News. (2021). Three years of GDPR: the biggest fines so far. Bbc.Com, May 2021, 1–10. https://www.bbc.com/news/technology-57011639. Accessed 11.11.2024.

Beck, K. (2022). Test Driven Development: By Example. Addison-Wesley Professional.

Cavoukian, A. (2009). Privacy by design: The 7 foundational principles. In Information and privacy commissioner of Ontario, 5(12).

CCPA. (2024). California Consumer Privacy Act (CCPA) | State of California - Department of Justice - Office of the Attorney General. California Consumer Privacy Act. https://oag.ca.gov/privacy/ccpa. Accessed 11.11.2024.

CMMI Institute. (2018). CMMI Institute - CMMI® Development V2.0. CMMI Institute. https://cmmiinstitute.com/resource-files/public/marketing/white-papers/cmmi%C2%AE-development-v2-0. Accessed 11.11.2024.

Cohen, S., Hudak, J. J., & McGregor, J. (2021). An Architecture Centric Approach to Safety and Security Assurance. AIAA/IEEE Digital Avionics Systems Conference - Proceedings, 2021-October. https://doi.org/10.1109/DASC52595.2021.9594336

Deshpande, M. V., Soitkar, P. A., Tripathi, D. R., & Agarmore, S. B. (2023). Ensuring Web Application Quality: The Role of Software Testing as a Form of Quality Assurance. 3rd IEEE International Conference on ICT in Business Industry and Government, ICTBIG 2023. https://doi.org/10.1109/ICTBIG59752.2023.10456277

Filipovikj, P., Causevic, A., & Lisova, E. (2020). Service realizability check as a technique to support a service security assurance case. Proceedings of the IEEE International Conference on Industrial Technology, 2020-February, 973–980. https://doi.org/10.1109/ICIT45562.2020.9067250

Galindo-Francia, J., & Auccahuasi, W. (2024). Implementation of Best Practices based on the MAC Methodology for Software Development Quality Assurance. 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things, IDCIoT 2024, 1727–1733. https://doi.org/10.1109/IDCIOT59759.2024.10467386

Gallaba, K. (2019). Improving the Robustness and Efficiency of Continuous Integration and Deployment. Proceedings - 2019 IEEE International Conference on Software Maintenance and Evolution, ICSME 2019, 619–623. https://doi.org/10.1109/ICSME.2019.00099

GDPR. (2018). General Data Protection Regulation (GDPR) – Legal Text. EU Regulation. https://gdpr-info.eu/. Accessed 11.11.2024.

Gonen, B., & Sawant, D. (2020). Significance of agile software development and SQA powered by automation. Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020, 7–11. https://doi.org/10.1109/ICICT50521.2020.00009

Gregor, S. (2006). The nature of theory in information systems. MIS Quarterly. https://doi.org/10.5555/2017296.2017300

Haider, A., & Bhatti, W. (2022). Importance of Cyber Security in Software Quality Assurance. 2022 17th International Conference on Emerging Technologies, ICET 2022, 6–11. https://doi.org/10.1109/ICET56601.2022.10004656

Holjevac, M., & Jakopec, T. (2022). Quality Assurance Procedures in Croatian IT Companies: Study into Employer's Perceptions and Experiences in Software Solution Testing. 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology, MIPRO 2022 - Proceedings, 1132–1137. https://doi.org/10.23919/MIPRO55190.2022.9803346

Hutton, D. M. (2009). Clean Code: A Handbook of Agile Software Craftsmanship20092Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice-Hall, 2008. £27.99, ISBN: 9-780-13235-088-4. Kybernetes, 38(6), 1035–1035.

Hynninen, T., & Jantunen, S. (2022). Questionnaire Approach for Assessing Software Engineering and Quality Assurance Practices. 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology, MIPRO 2022 - Proceedings, 1301–1306. https://doi.org/10.23919/MIPRO55190.2022.9803658

IEEE. (2014). IEEE Standard for Software Quality Assurance Processes. https://doi.org/10.1109/IEEESTD.2014.6835311. Accessed 11.11.2024.

ISO15408. (2022). ISO/IEC 15408-1:2022 - Information security, cybersecurity and privacy protection - Evaluation criteria for IT security - Part 1: Introduction and general model. https://www.iso.org/standard/72891.html. Accessed 11.11.2024.

ISO25002. (2024). ISO/IEC 25002:2024 - Systems and software engineering- Systems and software Quality Requirements and Evaluation (SQuaRE)- Quality model overview and usage. https://www.iso.org/standard/78175.html. Accessed 11.11.2024.

ISO27001. (2022). ISO/IEC 27001:2022 - Information security management systems. https://www.iso.org/standard/27001. Accessed 11.11.2024.

ISO29100. (2024). ISO/IEC 29100:2024 - Information technology — Security techniques — Privacy framework. https://www.iso.org/standard/85938.html. Accessed 11.11.2024.

Itkonen, J., & Rautiainen, K. (2005). Exploratory testing: A multiple case study. 2005 International Symposium on Empirical Software Engineering, ISESE 2005, 84–93. https://doi.org/10.1109/ISESE.2005.1541817

Janisar, A. A., Kalid, K. S. Bin, Sarlan, A. B., & Gilal, A. R. (2023). Security Requirements Assurance: An Assurance Case Perspective. 8th International Conference on Software Engineering and Computer Systems, ICSECS 2023, 78–83. https://doi.org/10.1109/ICSECS58457.2023.10256374

Jharko, E. (2021a). Life cycle and quality assurance of software for systems of critical information infrastructure facilities. Proceedings - 2021 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2021, 440–444. https://doi.org/10.1109/ICIEAM51226.2021.9446409

Jharko, E. (2021b). Some Aspects of Quality Assurance in the Development of Digital Systems. Proceedings of 2021 14th International Conference Management of Large-Scale System Development, MLSD 2021. https://doi.org/10.1109/MLSD52249.2021.9600164

Jonathan, Lim, A. P., Thenuardi, D. S., Soewito, B., & Antonyova, A. (2020). Survey on quality assurance testing on service-oriented architecture. Proceedings of 2020 International Conference on Information Management and Technology, ICIMTech 2020, 443–447. https://doi.org/10.1109/ICIMTECH50083.2020.9211258

Khan, R. A., Khan, S. U., Alzahrani, M., & Ilyas, M. (2022). Security Assurance Model of Software Development for Global Software Development Vendors. IEEE Access, 10, 58458–58487. https://doi.org/10.1109/ACCESS.2022.3178301

Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2022). Systematic Literature Review on Security Risks and its Practices in Secure Software Development. IEEE Access, 10, 5456–5481. https://doi.org/10.1109/ACCESS.2022.3140181

Kharchenko, A., Raichev, I., Bodnarchuk, I., & Matsiuk, O. (2021). The Survey of Global Software Design Processes. 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology, PIC S and T 2021 - Proceedings, 291–294. https://doi.org/10.1109/PICST54195.2021.9772205

Khurana, S. K., & Wassay, M. A. (2023). Towards Challenges Faced in Agile Risk Management Practices. 6th International Conference on Inventive Computation Technologies, ICICT 2023 - Proceedings, 937–942. https://doi.org/10.1109/ICICT57646.2023.10134188

Kruchten, P., Nord, R. L., & Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. IEEE Software, 29(6), 18–21. https://doi.org/10.1109/MS.2012.167

Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. Journal of Systems and Software, 101, 193–220. https://doi.org/10.1016/J.JSS.2014.12.027

Majanoja, A. M., & Hakkala, A. (2023). Enhancing a cybersecurity curriculum development tool with a competence framework to meet industry needs for cybersecurity. ACM International Conference Proceeding Series, 23, 123–128. https://doi.org/10.1145/3606305.3606325

Majanoja, A. M., Linko, L., & Leppänen, V. (2017). Global corrective action preventive action process and solution: Insights at the Nokia Devices operation unit. International Journal of Productivity and Quality Management, 20(1), 29–47. https://doi.org/10.1504/IJPQM.2017.080691

Majanoja, A.-M., Hakkala, A., Virtanen, S., & Leppänen, V. (2023). Motivation for continuous software engineering expertise development through lifelong learning. 19th International CDIO Conference, NTNU, Norway, 845–856.

Medium.com. (2020). Some of the most famous bugs in software history | by Kesk -*- | The Startup | Medium. https://medium.com/swlh/some-of-the-most-famous-bugs-in-software-history-bb16a2ee3f8e

Menascé, D. A. (2002). Load testing of Web sites. IEEE Internet Computing, 6(4), 70–74. https://doi.org/10.1109/MIC.2002.1020328

Mishra, A. D., & Mustafa, K. (2020). Security requirements specification: A formal method perspective. Proceedings of the 7th International Conference on Computing for Sustainable Global Development, INDIACom 2020, 113–117. https://doi.org/10.23919/INDIACOM49435.2020.9083691

Mishra, R. (2023). Comparative Analysis of Quality Assurance Models for Business Excellence: With special reference to ISO 9001 and EFQM model. Proceedings of 3rd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2023, 450–454. https://doi.org/10.1109/ICCIKE58312.2023.10131752

Nägele, S., Schenk, N., & Matthes, F. (2023). The Current State of Security Governance and Compliance in Large-Scale Agile Development: A Systematic Literature Review and Interview Study. Proceedings - 2023 IEEE 25th Conference on Business Informatics, CBI 2023. https://doi.org/10.1109/CBI58679.2023.10187439

Nakahara, H., Monden, A., & Yucel, Z. (2021). A Simulation Model of Software Quality Assurance in the Software Lifecycle. Proceedings - 22nd IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2021-Fall, 236–241. https://doi.org/10.1109/SNPD51163.2021.9704927

Neely, A., Mills, J., Platts, K., Richards, H., Gregory, M., Bourne, M., & Kennerley, M. (2000). Performance measurement system design: Developing and testing a process-based approach. International Journal of Operations and Production Management, 20(10), 1119–1145. https://doi.org/10.1108/01443570010343708/FULL/PDF

NIST. (2020a). NIST Special Publication 800-181 Revision 1: Workforce Frame-work for Cybersecurity (NICE Framework). ht. Https://Nlpubs.Nist.Gov/Nistpubs/SpecialPublications/NIST.SP.800-181rl.Pdf. https://cybersecurity.att.com/resource-center/solution-briefs/nist-compliance-usm-anywhere?utm_source=google&utm_medium=cpc&utm_term=kwd-337531935114&utm_campaign=10688982858&source=EBPS0000000PSM00P&WT.srch=1&wtExtndSource=ACS&wtpdsrchprg=AT%2526T%2520AB. Accessed 11.11.2024.

NIST. (2020b). Security and Privacy Controls for Information Systems and Organizations. https://doi.org/10.6028/NIST.SP.800-53r5. Accessed 11.11.2024.

NIST. (2024a). National Institute of Standards and Technology. https://www.nist.gov/. Accessed 11.11.2024.

NIST. (2024b). The NIST Cybersecurity Framework (CSF) 2.0. https://doi.org/10.6028/NIST.CSWP.29. Accessed 11.11.2024.

Niu, X., Yang, L., Liu, K., & Liu, Z. (2024). Research on the Transformation Path of DevOps in the Digital Era. International Conference on Advanced Communication Technology, ICACT, 248–251. https://doi.org/10.23919/ICACT60172.2024.10471935

OWASP. (2024). OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. Explore the World of Cyber Security. https://owasp.org/. Accessed 11.11.2024.

Presthus, W., & Sønslien, K. F. (2021). An analysis of violations and sanctions following the GDPR. International Journal of Information Systems and Project Management, 9(1). https://aisel.aisnet.org/ijispm/vol9/iss1/3

Presthus, W., & Sørum, H. (2019). Consumer perspectives on information privacy following the implementation of the GDPR. International Journal of Information Systems and Project Management, 7(3). https://aisel.aisnet.org/ijispm/vol7/iss3/3

Presthus, W., & Sørum, H. (2024). Five years with the GDPR: an empirical study emphasising information privacy and the consumer. International Journal of Information Systems and Project Management, 12(3). https://aisel.aisnet.org/ijispm/vol12/iss3/2

Ramirez, A., Aiello, A., & Lincke, S. J. (2020). A survey and comparison of secure software development standards. 13th CMI Conference on Cybersecurity and Privacy - Digital Transformation - Potentials and Challenges, CMI 2020. https://doi.org/10.1109/CMI51275.2020.9322704

Raygun.com. (2022). 10 Famous Bugs in The Computer Science World - GeeksforGeeks. https://www.geeksforgeeks.org/10-famous-bugs-in-the-computer-science-world/. Accessed 11.11.2024.

Ruparelia, N. B. (2010). Software development lifecycle models. ACM SIGSOFT Software Engineering Notes, 35(3), 8–13. https://doi.org/10.1145/1764810.1764814

Shikta, S., Mahir Shahriyar, H. M., Das, S. K., Nur Mahal, S., Al Jannat, K. B., & Alam, S. (2021). Quality assurance guidelines for successful startups. 2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications, SERA 2021, 81–85. https://doi.org/10.1109/SERA51205.2021.9509046

Siang, L. Y., & Selvarajah, V. (2022). Security Assurance through Penetration Testing. 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications, ICMNWC 2022. https://doi.org/10.1109/ICMNWC56175.2022.10031663

Sinha, A., & Das, P. (2021). Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry. 2021 5th International Conference on Electronics, Materials Engineering and Nano-Technology, IEMENTech 2021. https://doi.org/10.1109/IEMENTECH53263.2021.9614779

Tom, E., Aurum, A., & Vidgen, R. (2013). An exploration of technical debt. Journal of Systems and Software, 86(6), 1498–1516. https://doi.org/10.1016/J.JSS.2012.12.052

Tona, C., Juarez-Ramirez, R., Jimenez, S., Quezada, A., Guerra-Garcia, C., & Pacheco Lopez, R. G. (2021). Scrumlity: An Agile Framework Based on Quality Assurance. Proceedings - 2021 9th International Conference in Software Engineering Research and Innovation, CONISOFT 2021, 88–96. https://doi.org/10.1109/CONISOFT52520.2021.00023

Wagner, T. J., & Ford, T. C. (2020). Metrics to Meet Security Privacy Requirements with Agile Software Development Methods in a Regulated Environment. 2020 International Conference on Computing, Networking and Communications, ICNC 2020, 17–23. https://doi.org/10.1109/ICNC47757.2020.9049681

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly, 26(2), xiii–xxiii.

Wikantyasa, I. M. A., Kurniawan, A. P., & Rochimah, S. (2023). C K Metric and Architecture Smells Relations: Towards Software Quality Assurance. 2023 14th International Conference on Information and Communication Technology and System, ICTS 2023, 13–17. https://doi.org/10.1109/ICTS58770.2023.10330874

Wired.com. (2022). History's Worst Software Bugs | WIRED. https://www.wired.com/2005/11/historys-worst-software-bugs/. Accessed 11.11.2024.

Wong, W. Y., Hai Sam, T., Too, C. W., & Fong Pok, W. (2022). Software Quality Assurance Plan: Setting Quality Assurance Checkpoints within the Project Life Cycle and System Development Life Cycle. 2022 IEEE 18th International Colloquium on Signal Processing and Applications, CSPA 2022 - Proceeding, 214–219. https://doi.org/10.1109/CSPA55076.2022.9782044

Zhao, Y., Hu, Y., & Gong, J. (2021). Research on International Standardization of Software Quality and Software Testing. Proceedings - 2021 IEEE/ACIS 21st International Fall Conference on Computer and Information Science, ICIS 2021-Fall, 56–62. https://doi.org/10.1109/ICISFALL51598.2021.9627426

Biographical notes

Anne-Maarit Majanoja is a university teacher (PhD, MA. Ed.) in Software Engineering at the Department of Computing, University of Turku, Finland. With over a decade of industry experience, she has gained in-depth expertise in global IT development, quality management, leadership, and IT outsourcing environments. In addition, she has worked extensively in developing course content and educational solutions in the areas of software engineering and software security, and has also implemented several continuous education courses at the University of Turku. Her current research interests include quality, security and privacy practices in software engineering.

ORCID: 0000-0002-0340-775X

Ville Leppänen is a full professor in software engineering and software security (since 2012). At the moment, he is also vice dean of Faculty of Technology. He received his PhD in 1996 (Computer Science) and has now more than 250 international conference and journal publications. His research interests are related broadly to software engineering and security, ranging from software engineering methodologies, practices, and tools to security and quality issues, as well as to programming languages, parallelism, and architectural design topics. Leppänen is a member in several boards and working groups in University of Turku and outside the university.

ORCID: 0000-0001-5296-677X