11124

Secure Agile Development: Assessing the Integration of Security into Agile Development

Ryan Sajulga
sajul004@morris.umn.edu
Division of Computer Science
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

Agile methodologies have revolutionized software development practices, emphasizing flexibility, collaboration, and focus on customer satisfaction. However, applying security measures to Agile environments presents significant challenges. To ensure security, it's important to include *security activities* into the software development process to identify and prevent vulnerabilities. However, these practices often clash with the Agile principles since they tend to be more complex and require additional documentation and tools. This research paper dives into how security practices influence various aspects of Agile development—exploring the views of software engineers on integrating security practices in Agile environments with an online survey. The results offer insights into the impact of security practices on Agile development and provide recommendations for improving the integration of security measures in Agile processes.

Keywords: Agile, Software Engineering, Security Activities

1 Introduction

Agile methodologies, known for their flexibility, iterative delivery, and customer focus, have become increasingly popular as they promote rapid releases and adapt to changing requirements, making them a natural fit for organizations striving for efficiency and responsiveness. However, the speed and fluidity of Agile workflows introduce challenges, particularly in integrating security practices.

Secure software is essential for keeping user's information and data safe. Security prevents cyberattacks and keeps the system's integrity. Software that is not secure can lead to negative consequences. An example would be the Equifax data breach in 2017 [9]. This breach included one of the largest credit reporting agencies. The hackers were able to exploit the software's weakness and compromise about 147 million users' personal identifiable information (PII). This resulted in a multimillion-dollar settlement that addressed the organization's lawsuits and fines. This was a strong motivation for organizations to pay more attention to security and invest in security during the development phase.

Software security is crucial for these effects, however, the fast-paced nature of Agile can lead to security being overlooked, especially when the frequent changes and compressed timelines complicate the inclusion of security measures. To address this, various security activities have been implemented in development. Agile's emphasis on flexibility often contrasts with the more structured security measures required to detect vulnerabilities early, as tools for security-based code analysis typically generate more documentation than Agile processes prefer.

Despite these challenges, methods like DevSecOps [7], which integrates security into the Agile development pipeline, show promise in merging security activities with Agile practices. This paper explores how the adoption of security practices within Agile environments affects software developers' confidence in the security of their products, using survey data to assess how developers feel about the security practices and their confidence when these measures are integrated into their workflows. The author's goal was to provide insights that improve the adoption of security techniques for protecting complex software systems. This paper will cover background information on security and Agile, the research method used, the results, and an analysis of those results, and will conclude with a summary.

2 Background

2.1 Security

Security practices are implemented to ensure that the products developed are both secure and dependable. However, securing software has become an increasingly challenging task for software developers [10]. To address this challenge, a variety of tools have been utilized. Tools such as automated static analysis tools like FindSecurityBugs [11] and Bandit [12], have been introduced to streamline and simplify security tasks during development. FindSecurityBugs is designed to identify security vulnerabilities in Java applications. Similarly, Bandit is intended for identifying common security issues specifically in Python code. In research conducted by Thool and Brown, eight security practices were identified as being particularly valuable in improving the security of Agile software development processes [18]. These practices are outlined in Table 1. The focus will be on these eight practices, which will be explored in a survey to assess how Agile practitioners perceive them.

Table 1. Security Activities for Agile Software Development

Security Activity	Definition
Addressing security in early iterations with requirements and testing	This security activity emphasizes the importance of development teams addressing security issues and concerns early in the project before deploying the software.
Stating security requirements that are expected in the production software	This requires incorporating security expectations in project requirements when describing the responsibilities and behavior of the software.
Adding a security specialist to your team	Security specialists, such as a Security Master, are members of a development team that focus on security aspects of the project to address concerns and ensure the security of the system.
Additional points or weights to issues with an impact on security	This activity involves increasing the weights, such as story points in an Agile development environment, of issues that will have a higher impact on the security of the product to prioritize security-related tasks and encourage more secure development and testing.
Iterative and incremental vulnerability and penetration testing	This security activity suggests incorporating recurring security scanning, such as Dynamic Application Security Testing (DAST), to test for security flaws in the working software automatically.
Iterative and incremental security static analysis	Similar to DAST, Static Application Security Testing (SAST) involves using security-related static analysis tools to detect potential security vulnerabilities by scanning the source code.
Iterative and incremental risk analysis, countermeasure graphs	This security activity consists of using tools to monitor networks, applications, and infrastructure and perform risk analysis to identify vulnerabilities. These tools can evaluate the system's security and suggest methods to prevent attacks.
Automatic testing	This security activity involves incorporating secure coding practices, such as vulnerability analysis and risk assessment, into the deployment pipeline for software projects. This allows security checks to be automatically triggered with code changes and issues to be addressed before the software is deployed to users.

2.2 Agile Development

The software engineer's process helps organize activities within the Software Development Cycle (SDLC), which typically starts with requirements analysis and proceeds through design, implementation, testing, maintenance, and deployment. Agile is a prominent framework within software engineers' processes that emphasizes iterative and incremental development, continuous delivery, and customer satisfaction through collaboration. Introduced in 2001, the Agile Software Development Manifesto [2] outlines values and principles focusing on flexibility and responsiveness to change, aiming to reduce development costs and avoid unnecessary rework inherent in traditional software engineers processes [4]. Various Agile methodologies, such as Scrum, Kanban, and Feature-Driven Development [15], are used in software development due to their ability to adapt to changing requirements efficiently. They promote delivering potentially shippable software in iterations, allowing teams to provide high-value features to customers quickly. Books like "The Agile Samurai" [13], used in courses like the UMN Morris Software Design and Development class, further emphasize Agile's core principles of collaboration, flexibility, and frequent delivery of working software, typically every few weeks to a couple of months, prioritizing shorter timescales to maximize customer satisfaction.

While Agile methods bring rapid software delivery to the table, they also present the challenge of integrating security practices. The authors state that the Agile methodologies do not decrease software security, however, these challenges arise in properly applying security measures, tracking requirement changes, and ensuring security requirements are not overlooked during the development process [1] When it comes to aligning security objectives with Agile software development it is essential to ensure that the security measures are effectively integrated into the development press.

Security engineering processes and activities need to be carefully incorporated into the iterative and incremental software engineering process to achieve security objectives successfully [14].

3 Methodology

The data collection process for the author's study includes the creation and distribution of an online survey using QuestionPro [18]. The survey was designed to investigate the impact of security practices on various aspects of Agile development. Specifically, the study aimed to understand the challenges and implications of integrating security measures into Agile processes, the influence of security practices on software development, and the perception of software practitioners regarding the impact of security practices on Agile development [18].

The structure of the survey was comprised of nine questions with multiple parts, made to gather insights from participants. These questions focused on security practices, experiences in Agile development, and the perceived impact of security practices on software development. The survey also collected background details on participants' experiences with implementing security practices into Agile development to ensure a diverse perspective. Using a mix of closed-end questions, Likert scale, and open-ended questions allowed for a comprehensive assessment of participants' perspectives on security practices within Agile processes. The study explores the following research questions (RQs)

RQ1 How do software practitioners perceive the effectiveness of adopted and state-of-the-art security practices, and what is their level of willingness to incorporate them into the Agile software development process?
RQ2 How are the team velocity and productivity, as perceived by the software practitioners, affected by the inclusion of security activities?
RQ3 What is the impact of integrating security activities into Agile development on software practitioners' confidence in their software product and organization?

3.1 Participants

The goal of the survey was to find a diverse pool of participants to help ensure a comprehensive perspective on the RQs. The recruitment strategy consisted of reaching out with personalized invites, posts on LinkedIn [1], reaching out to IT teams through Slack [2]. The researchers also reached out to Virginia Tech graduate students with relevant technical work experience. [18]. Overall, the survey was completed by 34 individuals. Information about the participants with their self-reported answers can be found in Table 2. Of the participants, 67% (n = 23) were working professionals with an average of eight years of technical work experience in organizations such as Acquia, Flexcar, GlobalLogic, Decisions.com, Cvent, Lutron Electronics, Palo Alto Networks, and Microsoft, holding roles like software engineer, infrastructure engineer, senior product manager, technical consultant, systems architect, and database administrator [18]. The remaining 33% (n = 11) were graduate students from Virginia Tech, pursuing studies in software engineering and bringing an average of two years of relevant work experience. Most participants viewed security as either extremely (n = 25, 76%) or very important (n = 7, 21%) for their software teams, providing a broad range of insights into security practices within Agile processes.

4 Results

In this section, we will go over the results of the survey and talk about each of the research questions mentioned in the introduction section. Analyzing and understanding the survey responses to find how the software engineers value the security practices in Agile development.

4.1 RQ1: Security Practices in Agile, Effectiveness, and Willingness to Adopt

4.1.1 Adoption: The study revealed that 97% (n = 33) of participants reported using Agile software development methodologies. However, only 72% (n = 23) had security-related activities incorporated into their Agile processes. When excluding participants who reported being students, 77% (n = 27) of software professionals adopt security activities in their development process. These participants work in Agile development, but, not specifically in Agile security. An example of security activities that were used in the participant's Agile teams includes security training, security scanning & monitoring systems, security static analysis tools, code reviews, and integrating standards such as Open Worldwide Application Security Project (OWASP) [6], Identity Access Management, Multi-Factor Authentication, and zero trust policies [16], and separate security teams.

4.1.2 Perspective on Security Practices: The next section of the survey contained questions structured to understand the respondents' perspectives when it came to security practices. An example question from the survey would be...

> *What is your take on these security practices used in your team? (Optional)*

The responses had diverse opinions on the topic. A majority of the responses were expressed as "good" (n = 8), "informative" (n = 2), "necessary" (n = 2) and something that had to be complied with (n = 1). However, some of the participants expressed the security practices were "time-consuming" (n = 1) and "disliked" (n = 1). Finally, respondents acknowledged the need for improvement (n = 4), emphasizing the importance of enhancing security measures. [18].

4.1.3 Effectiveness of software security practices and willingness to include them in Agile. The survey includes the security practices 1 and asks participants how effective the practices would be in increasing the security and robustness of the software.

> *How effective would each security practice be in increasing the security and robustness of the software, if your team would include it in the Agile software development process.*
>
> *How willing are you to include each security practice in your Agile software development process?*

About 30 participants responded and the results can be seen in Table 3 and Table 4. The table highlights that activities involving early integration of security and automation are perceived as the most effective with "Iterative vulnerability testing" scoring a 60% and "Automatic testing" scoring a 66.67%. In contrast, manual or less structured activities (like static analysis and risk assessment) are seen as moderately effective.

When it comes to willingness to include these security practices, The table highlighted a strong willingness among practitioners to embrace security practices within agile processes. This support is particularly evident for activities such as automated testing and vulnerability assessment. "Iterative vulnerability testing" scoring a 64.52% and "Automatic testing" scoring a 67.74%.. The data reflects a broader trend

[1] https://www.linkedin.com/
[2] https://www.slack.com/

Table 2. Survey Participants

Participant	Role	Industry Exp. (years)	Agile?	Sec?	Participant	Role	Industry Exp. (years)	Agile?	Sec?
P1	Associate Software Engineer	1	Yes	Yes	P18	Chief Test Monkey	41	Yes	Yes
P2	Software Engineer	2.2	Yes	Yes	P19	Cloud Engineer	7	Yes	Yes
P3	Software Engineer	0.5	Yes	Yes	P20	Systems Architect	8	Yes	Yes
P4	Engineering Manager	11	Yes	Yes	P21	Department Head	23	Yes	Yes
P5	Software Engineer	6	Yes	No	P22	Associate Director of Systems Development	25	Yes	Yes
P6	Student	0	Yes	Yes	P23	Director, DBAA	22	Yes	Yes
P7	Quality Engineer	1.5	Yes	Yes	P24	Software Developer	0	No	No
P8	Graduate Teaching Assistant	1	Yes	Yes	P25	Software Engineering Co-Op	1	Yes	Yes
P9	Student	4	Yes	No	P26	Senior Product Manager	10	Yes	Yes
P10	Consultant	3	Yes	Yes	P27	Software Engineer	12	Yes	Yes
P11	Senior Software Engineer	5	Yes	Yes	P28	Student	2	Yes	Yes
P12	Student	3	Yes	Yes	P29	Student	2	Yes	Yes
P13	Student	0	Yes	Yes	P30	Technical Consultant	13	Yes	Yes
P14	Automation Test Engineer	4.2	Yes	Yes	P31	Software Engineer	2.5	Yes	Yes
P15	Graduate Student	2.8	Yes	Yes	P32	Security Co-Op	0.5	Yes	Yes
P16	Student	0	Yes	No	P33	Senior Staff Machine Learning Engineer	16	Yes	Yes
P17	Senior Software Engineer	6	Yes	No	P34	Infrastructure Engineer	1.5	Yes	Yes

toward prioritizing proactive and integrated security measures and helps emphasize the importance of embedding security early and using automated tools in development practices.

4.2 RQ2: Productivity

4.2.1 Team Velocity. This question asked about what the effects the security practices had on the sprint velocity. Spring velocity refers to the amount of work a team can complete during a sprint, which is a key metric used in Agile methodology.

How was the sprint velocity affected? (Optional)

About 14 participants responded. The majority of participants reported were not affected by adopting the security practices. However, the impact was dependent of team-specific policies, such as planning security activities before the sprint. Some participants commented about the productivity being affected by “10-20%” (P17) and another comment said “one to two days” (P16, P28). The last notable comment is from one participant (P14) who responded by saying that incorporating new initiatives for security engineering “will always affect the sprint velocity drastically” leading to a slower rollout of features, but concluded, “such changes are fruitful” [18].

4.2.2 Day-to-day Activities. When it came to day-to-day activities, the majority (n = 16) concluded that they have little to no effect on daily development tasks. Specific security activities that were mentioned to not interrupt development processes include integrating tools to do “periodic security checking” (P20) and having an external team, referring to a group of specialist on the security aspect. However, about two participants mentioned that the security activities lead to "more time spent authenticating to access different environments and projects”. Overall, integrating security activities in Agile development processes did not have a major impact on the sprint velocity of Agile development teams [18].

4.3 RQ3: Impact

4.3.1 Software Products. This part of the survey was optional and asked how incorporating security activities into Agile affected their software products as a whole.

How has the involvement of these security practices affected the software product? (Optional)

14 participants responded and shared how these security practices influenced their software products. Many participants (n = 10) noted an increase in overall security in their software products. This suggests that implementing strong security measures has a positive effect on software, enhancing its security and resilience to potential threats. Additionally, some participants highlighted that these practices boosted customer trust (n = 1), emphasizing the value of fostering end-user confidence. Other reported benefits included increased team confidence (n = 1), adherence to higher standards (n = 1), and a reduction in bugs (n = 1).

However, there were noted downsides. For instance, one participant mentioned that security activities delayed delivery timelines, as code often awaited approval (P30). Another participant, P19, reported that their team’s security practices rarely impacted the product’s security. Overall, our findings indicate that security practices have a beneficial impact on software products—improving security posture, customer

Table 3. Security Activities and Practitioners' Perceived Effectiveness

Security Activity	Not at all	Slightly	Moderately	Very	Extremely
Addressing security in early iterations with requirements and testing	0%	0%	13.33%	73.33%	13.33%
Stating security requirements that are expected in the production software	0%	3.33%	20%	46.67%	30%
Adding a security specialist to your team	0%	6.67%	20%	40%	33.33%
Additional points or weights to issues with an impact on security	0%	0%	20%	46.67%	33.33%
Iterative and incremental vulnerability and penetration testing	0%	0%	10%	30%	60%
Iterative and incremental security static analysis	0%	3.33%	6.67%	53.33%	36.67%
Iterative and incremental risk analysis, countermeasure graphs	0%	6.67%	30%	43.33%	20%
Automatic testing	0%	3.33%	0%	30%	66.67%

Table 4. Security Activities and Practitioners' Willingness to Include Them in Agile Processes

Security Activity	Not at all	Slightly	Moderately	Very	Extremely
Addressing security in early iterations with requirements and testing	0%	0%	29.03%	45.16%	25.81%
Stating security requirements that are expected in the production software	0%	0%	29.03%	41.94%	29.03%
Adding a security specialist to your team	0%	6.45%	19.35%	48.39%	25.81%
Additional points or weights to issues with an impact on security	0%	0%	12.9%	38.71%	48.39%
Iterative and incremental vulnerability and penetration testing	0%	0%	16.13%	19.35%	64.52%
Iterative and incremental security static analysis	0%	0%	3.23%	45.16%	51.61%
Iterative and incremental risk analysis, countermeasure graphs	3.23%	0%	22.58%	48.39%	25.81%
Automatic testing	0%	0%	3.23%	29.03%	67.74%

trust, team morale, and quality. However, careful planning is needed to avoid delays in feature deployment.

4.3.2 Organization. The authors also investigated how security activities influenced the broader organization. About 12 participants responded to this question. While some participants reported positive effects, such as an increase in security practices (n = 1), improved organizational culture (n = 1), enhanced company reputation (n = 1), and greater customer confidence (n = 1), most respondents saw either no effect (n = 4) or minimal impact (n = 4). This suggests that while some organizations perceive clear benefits, others may not see significant changes from implementing these practices. Further analysis is required to identify factors driving these variations and strategies for optimizing security integration in organizational contexts. Additionally, benefits may be less visible as it's difficult to quantify how many security threats were prevented due to these practices.

4.3.3 Confidence. The study explored how security practices influenced participants' confidence in the security of their software. The results showed that 50% of respondents felt "fairly confident" in their software's security, 25% were "somewhat confident," and another 25% were "completely confident." These findings suggest that security practices positively impact the confidence and trust of software practitioners. While the study underscores the importance of integrating robust security measures into the development process, it also highlights the need for further efforts to enhance security within Agile to help instill software engineers' confidence in their products' security.

5 Analysis of Results

The study provides valuable insights into integrating security activities within Agile development, focusing on software practitioners' perceptions and their impact on the Agile process. Participants generally had a positive view of incorporating security practices into Agile, despite potential conflicts between the domains. This alignment with the security practices shows a growing awareness of the importance of software security, with security activities having minimal impact on productivity while generally improving software security. Although some participants noted occasional delays and increased time in feature deployment due to security activities, they believed the benefits of enhanced security

outweighed these concerns. The study suggests that successful integration of security practices is possible with careful planning and efficient implementation, allowing Agile teams to maintain productivity. However, some participants expressed only moderate confidence that these security activities were effectively improving software protection. The study recommends increasing automation in security processes and improving feedback mechanisms to enhance the effectiveness of security practices and boost confidence in system security.

5.1 Automation

Participants in the study showed a positive perception and willingness to adopt Agile processes that involve automation and security tools. The authors state that there was no significant difference in the effectiveness of the eight security activities surveyed; however, automated approaches, such as automatic testing, were viewed more favorably and had higher adoption rates, from Table 3 and Table 4 compared to manual practices, like adding a security specialist. This preference for automation aligns with prior research, which has shown that automated penetration testing is more efficient than manual methods [17]. While automated security techniques have some potential drawbacks, including inaccurate output, failure to meet stakeholder requirements and information overload, they still offer significant advantages [18]. There is an opportunity to further streamline security activities through automation, and teams are encouraged to implement automated tools for testing, vulnerability assessments, penetration testing, and static security analysis to enhance software security. Automation was also found to be highly effective and conducive to supporting security activities within iterative development, such as continuous security testing and integrating automated security tools into development pipelines. Additionally, the study highlights the growing adoption of DevSecOps[7], an extension of the DevOps methodology that incorporates security tools and concepts directly into development pipelines.

5.2 Enhanced Feedback

The study emphasizes caution against blindly automating security tasks, as doing so can lead to ineffective or incomprehensible results. Previous research suggests that users are unlikely to trust and use automated tools that provide incorrect or unclear feedback [8]. In this study, some participants noted that the security activities adopted by their teams had minimal impact on product security, which contributed to a lack of confidence in these processes. Additionally, information overload—providing too much data without context—can overwhelm developers, making the feedback from security tools less useful [5]. To improve tool adoption and effectiveness, security activities must include actionable feedback. Collaboration between security experts and developers is crucial for enhancing feedback and increasing trust in automated tools. Automated tools should not only report vulnerabilities but also provide context and actionable insights to help developers understand and address security issues. Furthermore, developers often lack an understanding of the consequences of security issues, making it difficult to respond effectively. To improve understanding, prior research suggests using vignettes or brief stories to explain security concepts and influence developers' behavior [3]. Automated tools could also suggest fixes for reported vulnerabilities, addressing developers' lack of knowledge on how to resolve security issues. Research on automated program repair has shown its potential to improve debugging, and similar tools for suggesting security fixes could significantly enhance software security.

6 Conclusion

Overall, integrating security practices into Agile software development introduces both challenges and opportunities. This study reveals that while most software practitioners acknowledge the importance of security within Agile processes, a substantial gap still exists between acknowledgment and effective implementation. The findings indicate that embedding security practices not only enhances software quality but also strengthens customer trust, which in turn boosts practitioners' confidence in their products' security. However, the increased complexity and potential delays in development timelines highlight the necessity for more effective planning and optimization of these practices.

Organizations stand to gain by incorporating security from the earliest stages of development, utilizing automation tools, and fostering collaboration between security and development teams. Making security a continuous and integral aspect of the Agile workflow is essential to balancing the dual demands of rapid delivery and solid security. Although challenges remain—such as managing shifting requirements and maintaining sprint velocity—the insights from this research offer a roadmap for better aligning security goals with Agile methodologies.

References

[1] Reem Alshareef, Esra'a Alshabeeb, Noor Alakkas, and Mahmood Niazi. 2024. Challenges in Developing Secure Software within Agile Environments. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering* (Salerno, Italy) *(EASE '24)*. Association for Computing Machinery, New York, NY, USA, 652–661. https://doi.org/10.1145/3661167.3661284

[2] Kent Beck and et al. 2001. The Agile Manifesto. http://agilemanifesto.org/ Accessed: 2024-12-02.

[3] John Blythe. 2013. Cyber Security in the Workplace: Understanding and Promoting Behaviour Change. In *Proceedings of CHItaly 2013 Doctoral Consortium*, Vol. 1065. 92–101.

[4] Kieran Conboy. 2009. Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research* 20, 3 (2009), 329–354.

[5] W. Keith Edwards, Erika Shehan Poole, and Jennifer Stoll. 2008. Security automation considered harmful?. In *Proceedings of the 2007*

Workshop on New Security Paradigms (New Hampshire) *(NSPW '07)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/1600176.1600182

[6] OWASP Foundation. n.d.. OWASP Application Security Verification Standard. https://github.com/OWASP/ASVS Accessed: 2024-12-02.

[7] Henry Haverinen, Tomi Janhunen, Tero Päivärinta, Sami Lempinen, Suvi Kaartinen, and Sami Merilä. 2024. Automating Cybersecurity Compliance in DevSecOps with Open Information Model for Security as Code. In *Proceedings of the 4th Eclipse Security, AI, Architecture and Modelling Conference on Data Space* (Mainz, Germany) *(eSAAM '24)*. Association for Computing Machinery, New York, NY, USA, 93–102. https://doi.org/10.1145/3685651.3686700

[8] Brittany Johnson, Christian Bird, Denae Ford, Nicole Forsgren, and Thomas Zimmermann. 2023. Make Your Tools Sparkle with Trust: The PICSE Framework for Trust in Software Tools. In *Proceedings of the 45th International Conference on Software Engineering: Software Engineering in Practice* (Melbourne, Australia) *(ICSE-SEIP '23)*. IEEE Press, 409–419. https://doi.org/10.1109/ICSE-SEIP58684.2023.00043

[9] McKenzie L. Kuhn. 2018. 147 Million Social Security Numbers for Sale: Developing Data Protection Legislation After Mass Cybersecurity Breaches. *Iowa Law Review* 104 (2018), 417.

[10] Fabiola Moyón, Florian Angermeir, and Daniel Mendez. 2024. Industrial Challenges in Secure Continuous Development. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice* (Lisbon, Portugal) *(ICSE-SEIP '24)*. Association for Computing Machinery, New York, NY, USA, 309–311. https://doi.org/10.1145/3639477.3639736

[11] Joseph P. Near and Daniel Jackson. 2016. Finding Security Bugs in Web Applications using a Catalog of Access Control Patterns. In *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016)*. http://hdl.handle.net/1721.1/102281

[12] PyCQA. 2024. Bandit: A Security Linter for Python Source Code. GitHub repository. https://github.com/PyCQA/bandit Accessed: 2024-12-02.

[13] Jonathan Rasmusson. 2010. *The Agile Samurai: How Agile Masters Deliver Great Software*. Pragmatic Bookshelf.

[14] Kalle Rindell, Sami Hyrynsalmi, and Ville Leppänen. 2018. Aligning security objectives with agile software development. In *Proceedings of the 19th International Conference on Agile Software Development: Companion* (Porto, Portugal) *(XP '18)*. Association for Computing Machinery, New York, NY, USA, Article 3, 9 pages. https://doi.org/10.1145/3234152.3234187

[15] Sabbir M. Saleh, Syed Maruful Huq, and M. Ashikur Rahman. 2019. Comparative Study within Scrum, Kanban, XP Focused on Their Practices. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 1–6.

[16] Malcolm Shore, Sherali Zeadally, and Astha Keshariya. 2021. Zero Trust: The What, How, Why, and When. *Computer* 54, 11 (2021), 26–35.

[17] Yaroslav Stefinko, Andrian Piskozub, and Roman Banakh. 2016. Manual and Automated Penetration Testing: Benefits and Drawbacks. Modern Tendency. In *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*. IEEE, 488–491.

[18] Arpit Thool and Chris Brown. 2024. Securing Agile: Assessing the Impact of Security Activities on Agile Development. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering* (Salerno, Italy) *(EASE '24)*. Association for Computing Machinery, New York, NY, USA, 668–678. https://doi.org/10.1145/3661167.3661280