# ENHANCING SOFTWARE DEVELOPMENT THROUGH CYBERSECURITY INTEGRATION AT EACH PHASE OF THE LIFECYCLE

Serhii Hlushko
Simon Kuznets Kharkiv National University of Economics

DOI 10.24917/9788368020861.7

**Abstract**

This research delves into the critical examination of cybersecurity integration within the Software Development Life Cycle (SDLC), emphasizing its necessity from the initial stages of development to safeguard against potential vulnerabilities. The study underscores the importance of preemptive cybersecurity measures in mitigating risks, enhancing software reliability, and ensuring cost-effective development processes. By integrating cybersecurity practices early in the SDLC, organizations can avoid the compounded costs and efforts associated with rectifying security flaws in later stages. The research aims to provide a comprehensive analysis of how cybersecurity integration influences the various phases of software development, including requirements analysis, design, implementation, and quality assurance, and its impact on the overall project cost and timeline. Through a combination of theoretical exploration and empirical investigation, this study seeks to establish a framework for effective cybersecurity practices within the SDLC, thereby contributing to the development of more secure, reliable, and efficient software systems.

**Keywords:** cybersecurity integration, software development life cycle (SDLC), early-stage security, cost efficiency, development timelines, software reliability.

Cybersecurity significantly influences various phases of the Software Development Life Cycle (SDLC), requiring a holistic and integrated approach to ensure the development of secure and resilient software systems. The integration of cybersecurity principles and practices at each phase of the SDLC not only mitigates security risks but also enhances the overall quality and reliability of software products.

The integration of cybersecurity within the SDLC serves a multifaceted purpose, directly correlating with the operational, financial, and reputational dimensions of organizations engaged in or reliant on software development. This strategic inclusion aims to instill a comprehensive security posture from the inception to the decommissioning of software products, addressing the evolving landscape of cyber threats and vulnerabilities. The rationale and implications of this integration for modern practical tasks can be elucidated through the lenses of monetary impact, time efficiency, and reputation management.

The digital transformation of businesses, the proliferation of cloud computing, and the increasing reliance on software for critical operations underscore the importance of security as a foundational element of software development. As organizations navigate the complexities of regulatory compliance, data protection laws, and the escalating sophistication of cyber threats, the strategic incorporation of cybersecurity practices within the SDLC aligns with operational resilience, risk management, and sustainable growth objectives.

So, it's clear that understanding how Cybersecurity integration affects the final result of the Software Development process in terms of money, time, reputation and other aspects depending on which SDLC phase such integration will take place is important for the business and Project Management theory in general.

Existing publications illuminate various dimensions of integrating cybersecurity within the Software Development Life Cycle (SDLC) and the broader digital ecosystem. Each publication, through its unique lens, underscores the criticality of cybersecurity measures and offers insights into effective implementation strategies, albeit with certain limitations or areas left unexplored.

Article [1] emphasizes the importance of incorporating security practices within the Agile development framework, highlighting the need for roles, ceremonies, and artifacts tailored to enhance security throughout the SDLC. While it provides a novel taxonomy of security practices, a potential limitation is its reliance on data from a limited geographical and organizational scope, potentially restricting the applicability of its findings across diverse contexts.

In [2] study, the author(s) seeks to establish a comprehensive guide for secure software development, driven by industry feedback and the necessity for technical and organizational measures. It addresses the crucial aspect of equipping software architects and developers with practical tools and measures to enhance security. However, the research might overlook the dynamic nature of cybersecurity threats, necessitating continuous updates to the guide to remain effective.

The [3] investigation explores the impact of cybersecurity knowledge, attitudes, and self-perception on cybersecurity awareness among university students. Its significant contribution lies in identifying the attitude as a mediator in cybersecurity awareness. Nonetheless, the study's focus on a specific demographic and region may limit its generalizability, suggesting a need for broader research to validate these findings across different populations.

The author(s) of [4] paper discusses the development of cybersecurity frameworks (CSFs) by various entities to combat cybersecurity threats. It offers a comparative analysis of existing frameworks, contributing to an understanding of their features and applicability. The limitation here is the lack of a unified approach or consensus on the optimal framework, indicating an area for further research and development.

Publication [5] addresses the intersection of web accessibility standards and cybersecurity, highlighting the W3C's efforts through the WCAG. It underscores the necessity of ensuring equivalent user experiences while maintaining cybersecurity standards. A possible shortcoming is the abstract's narrow focus on accessibility without a detailed exploration of how cybersecurity measures can be harmonized with accessibility guidelines.

These publications collectively underscore the multifaceted nature of cybersecurity integration within software development and digital operations. They highlight the importance of adopting a holistic approach that encompasses technical, organizational, and educational measures. However, each study presents limitations that suggest avenues for further research, including the need for updated guides to counter evolving threats, broader studies to enhance the generalizability of findings, and unified cybersecurity frameworks that balance security with usability and accessibility.

It is worth analyzing cybersecurity integration across different stages of the SDLC and think about underscores the necessity of a proactive approach to cybersecurity, advocating for ongoing research to address identified gaps, such as the dynamic nature of threats, the development of universally applicable frameworks, and the integration of cybersecurity with other critical standards like accessibility.

The primary objective of this work is to methodically examine the integration of cybersecurity measures at various stages of the Software Development Life Cycle (SDLC) and to elucidate its consequential impact on the cost and time expenditures associated with software development projects. This investigation seeks to articulate a comprehensive understanding of how preemptive cybersecurity practices can optimize the allocation of resources and enhance the efficiency of development processes, thereby providing empirical insights and theoretical underpinnings for the strategic planning and execution of software projects.

To establish a theoretical framework that delineates the role of cybersecurity within the SDLC, highlighting its significance in preempting potential vulnerabilities and threats at each phase – from requirements analysis and design to implementation, testing, deployment and other phases [6].

Also, conduct a rigorous cost-benefit analysis that quantifies the financial implications of integrating cybersecurity practices into the SDLC. This analysis will explore how the integration at different phases of SDLC the security measures affects the potential costs associated with mitigating security breaches post-deployment, including but not limited to, regulatory fines, legal liabilities, and reputational damage repair.

To evaluate the impact of cybersecurity integration on the time efficiency of software development projects. This includes assessing how proactive security measures can streamline development timelines, minimize the need for extensive rework due to security flaws discovered at later stages, and facilitate a quicker time-to-market for software products.

Based on the findings, to formulate strategic recommendations for software development teams and organizational leaders. These recommendations will aim to optimize their cybersecurity strategies within the SDLC, balancing the trade-offs between upfront investments in security and the long-term benefits of reduced vulnerabilities and enhanced software quality.

In the scholarly work on iterative software development methodologies, where the mathematical model of the Software Development process was described [7], the Scrum framework is analyzed for its efficacy in facilitating the software development process from initial conception to final delivery. The model delineated for a Scrum iteration – spanning from the initial Feature Idea to the Iteration phase – serves as a pivotal reference for understanding the progressive stages involved in software development.

The process begins with the Feature Idea phase, a preliminary stage characterized by the conceptualization of new functionalities or enhancements within the software, marking a phase of creative inception where stakeholders identify potential software improvements. Following this, the idea progresses to the Business Logic / Requirements phase, where Business Analysts play a crucial role in formalizing these ideas into concrete functional requirements, thereby setting clear objectives and expectations for the proposed feature.

Subsequently, the Technical Architecture phase sees the involvement of a Tech Lead who is tasked with crafting a technological strategy that encompasses design patterns, software components, interfaces, and other critical technical details essential for the feature's implementation. This phase is critical in laying the groundwork for the subsequent Design/UI/UX phase, wherein designers focus on the user interface and experience, ensuring the software's usability, accessibility, and aesthetic appeal.

The development journey continues as the Written Source Code phase transitions ideas into tangible outputs. Developers undertake the crucial task of coding, transforming the established requirements, architecture, and design into functional software. This development phase culminates in the Deployed on Server stage, marking the deployment of the code to a server environment, thereby transitioning the feature from development to a test or production setting, rendering it accessible to end-users or stakeholders.

The nuanced transitions between these states, influenced by the varying involvement of team members, highlight the collaborative and dynamic nature of Scrum. This is exemplified in the referenced scheme (Fig. 1), which provides a visual representation of the varying weights of transitions between stages, underscoring the importance of team collaboration in the iterative development process.

Cybersecurity may significantly influence various phases of the SDLC, requiring a holistic and integrated approach to ensure the development of secure and resilient software systems. The integration of cybersecurity principles and practices at each phase of the SDLC not only mitigates security risks but also enhances the overall quality and reliability of software products. Let's analyze those phases.

Requirements Analysis (BA): In this early phase, integrating cybersecurity involves identifying and specifying security requirements alongside functional requirements. Security needs are assessed based on the anticipated threats, regulatory compliance obligations, and risk management strategies. This early consideration of cybersecurity ensures that security is a foundational component of the software, rather than an afterthought, facilitating the development of a secure by design architecture.

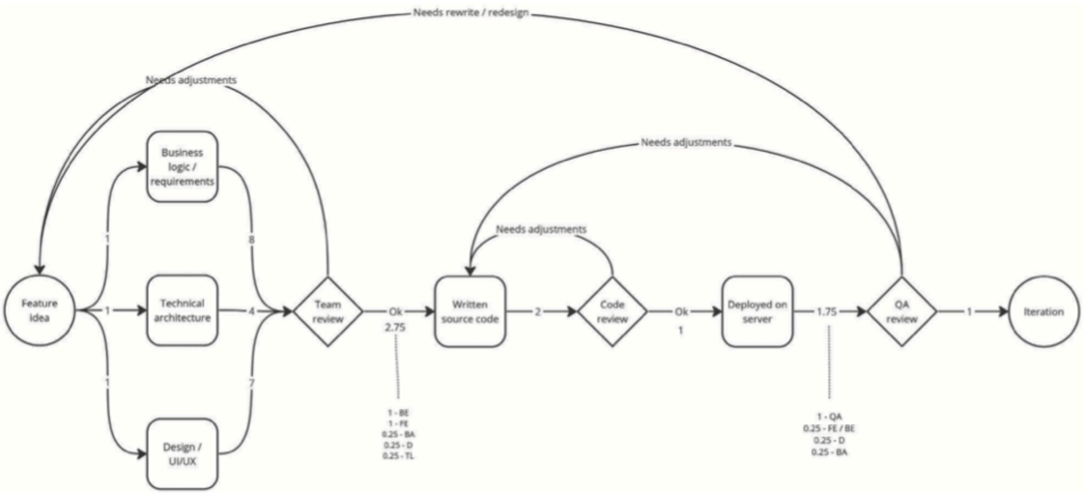

**Figure 1.** Scheme of transitions between states in the Software Development process

Design and Architecture: During the design phase, cybersecurity profoundly impacts the architectural decisions of software systems. Secure design principles, such as least privilege, defense in depth, and segregation of duties, are employed to architect systems that are resilient to attacks. Security patterns and anti-patterns guide the architectural choices, ensuring that the software infrastructure is robust against known vulnerabilities and attack vectors.

Implementation (Write Source Code): Cybersecurity considerations during the implementation phase focus on adhering to secure coding practices to prevent vulnerabilities at the source code level. Developers are encouraged to use coding standards that avoid common security flaws, such as SQL injection, cross-site scripting (XSS), and buffer overflows. Tools such as static and dynamic code analyzers are utilized to detect and rectify security issues early in the development process.

Testing and Quality Assurance (QA): In the QA phase, cybersecurity influences the adoption of security-specific testing methodologies, including vulnerability assessments, penetration testing, and security audits. These practices aim to uncover and address security weaknesses before the software is deployed. Incorporating security testing within the QA process ensures that security vulnerabilities are identified and mitigated, contributing to the reliability and security of the final software product.

Deployment and Maintenance: Cybersecurity continues to play a critical role post-deployment, as software must be maintained and updated to address new security threats and vulnerabilities. Continuous monitoring and regular security updates are essential to protect against emerging threats. The deployment phase also involves ensuring that the software is configured securely and that any deployment scripts or infrastructure as code practices do not introduce security weaknesses.

The integration of cybersecurity into SDLC serves a multifaceted purpose, directly correlating with the operational, financial, and reputational dimensions of organizations engaged in or reliant on software development. This strategic inclusion aims to instill a comprehensive security posture from the inception to the decommissioning of software products, addressing the evolving landscape of cyber threats and vulnerabilities. The rationale and implications of this integration for modern practical tasks can be elucidated through the lenses of monetary impact, time efficiency, and reputation management.

Early detection and mitigation of security vulnerabilities are considerably less expensive than addressing these issues post-deployment. According to industry studies [8, 9, 10], rectifying a security flaw during the maintenance or post-release stage can be up to 100 times more costly than if addressed during the design phase. Thus, a proactive security approach not only minimizes potential financial losses due to security breaches, including regulatory fines, litigation costs, and compensation expenses but also optimizes resource allocation and reduces development overhead.

By identifying and addressing security requirements and vulnerabilities early, organizations can avoid the time-consuming and resource-intensive tasks associated with retrofitting security measures into existing systems. This forward-looking approach reduces the likelihood of project delays caused by security-related rework, enabling a more predictable and efficient project timeline. Consequently, this efficiency contributes to faster time-to-market, a critical factor in today's competitive and rapidly evolving technological landscape.

It is worth mentioning that the reputation of an organization is intricately linked to its ability to safeguard customer data and maintain the integrity and availability of its services. Security breaches not only entail direct financial losses but can also inflict long-lasting damage to a company's reputation, eroding customer trust and loyalty. The integration of cybersecurity within the SDLC demonstrates an organization's commitment to security, potentially enhancing its reputation and competitive advantage.

At this moment we can see the necessity of incorporating cybersecurity measures at various stages of software development, such as requirements analysis, design, implementation, and quality assurance, to mitigate risks and bolster the overall security posture of the developed software.

It is worth highlighting the critical importance of integrating cybersecurity practices at the earliest possible stages of the SDLC. This approach is predicated on the understanding that identifying and addressing vulnerabilities in the later phases of software development not only incurs higher costs but also necessitates revisiting and potentially revising earlier stages, thus amplifying the effort and resources required for remediation. The argument for early integration of cybersecurity is compelling; early detection of security issues prevents the cascade of revising extensive sections of the project, thereby conserving resources and reducing the potential for significant setbacks. The integration of cybersecurity is presented not as a mere compliance requirement but as a strategic investment that can yield substantial dividends in terms of enhancing the reliability, trustworthiness, and marketability of software products. Additionally, leveraging rapidly advancing AI tools [11] can enhance security and productivity outcomes significantly.

This work established a foundational framework for the ongoing research into the integration of cybersecurity within the SDLC, underscoring the significance of early-stage integration and its implications for the software development process. As a next step, it is imperative to conduct empirical studies and develop case studies that can offer deeper insights into the practical applications of these theories. Such research could involve analyzing specific cybersecurity practices at each SDLC phase, assessing their impact on project timelines and budgets, and developing guidelines for effectively balancing security measures with development agility. Additionally, exploring the interplay between cybersecurity integration and emerging software development methodologies could offer fresh perspectives on adapting security practices to fit the evolving landscape of software engineering.

## References

1. A.A. Ardo, J.M. Bass, T. Gaber (2021). *An Empirical Investigation of Agile Information Systems Development for Cybersecurity*. European, Mediterranean, and Middle Eastern Conference on Information Systems, Cham: Springer International Publishing, pp. 567–581.
2. A. Ali, et al. (2021). *Secure Software Development: Infuse Cyber Security to Mitigate Attacks in an Organization*, International Conference on Engineering Software for Modern Challenges, Cham: Springer International Publishing, pp. 154–163.
3. B. Ahamed, et al. (2024). *Empowering Students for Cybersecurity Awareness Management in the Emerging Digital Era: The Role of Cybersecurity Attitude in the 4.0 Industrial Revolution Era*, SAGE Open, 14.1, 21582440241228920
4. A.D. Khaleefah, H.M. Al-Mashhadi (2024). *Methodologies, Requirements, and Challenges of Cybersecurity Frameworks: A Review*, Iraqi Journal of Science.
5. M. Hernández Bejarano, E.T. Páez Cruz, H. Simanca, A. Fredys (2021). *A Look at Usability, Accessibility and Cybersecurity Standards in Software Development*, Inter-

national Conference on Advanced Research in Technologies, Information, Innovation and Sustainability, Cham: Springer International Publishing, pp. 484–496.

6. S. Glushko, L. Zhang (2023). *Analysis of Flexible Software Development Methodologies*, Протидія кіберзлочинності та торгівлі людьми: зб. матеріалів міжнар. наук.-практ. конф. (м. Вінниця, 31 трав. 2023 р.), МВС України, Харків. нац. ун-т внутр. справ, Наук. парк «Наука та безпека». – Вінниця: ХНУВС, pp. 172–173.
7. S. Hlushko, O. Leunenko, I. Mikhieiev (2023). *Mathematical Model of the Software Development Process According to the "Outsource" Scheme*. 13th International Conference on Dependable Systems, Services and Technologies (DESSERT). IEEE, pp. 1–6.
8. R. Rwemalika, et al. (2019). *An Industrial Study on the Differences Between Pre-Release and Post-Release Bugs*, IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, pp. 92–102.
9. A. Anand, S. Das, O. Singh (2016). *Modeling Software Failures and Reliability Growth Based on Pre & Post Release Testing*. 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, pp. 139–144.
10. J. Kang, D. Ryu, J. Baik (2021). *Predicting Just-In-Time Software Defects to Reduce Post-Release Quality Costs in The Maritime Industry*, Software: Practice and Experience, 51.4, pp. 748–771.
11. С.О. Глушко (2024). *Підвищення продуктивності розробки програмного забезпечення засобами штучного інтелекту*, Інформаційні технології та інженерія: Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів 31 січня–2 лютого 2024 р., м. Миколаїв: тези, ЧНУ ім. Петра Могили, Миколаїв, pp. 111–112.