

**Integrating Security in Software Development LifeCycle Using Protection Poker Technique**

**HASSAN SAEED**

College of Electrical and Mechanical Engineering, National University of Sciences & Technology (NUST), Pakistan (e-mail: hassan.saeed31@gmail.com)

**ABSTRACT** Security aspects in a software is a growing and legit concern specially with 5G data availability just at our palms. Research conducted in this field is evolving with new techniques that necessitate periodic comparative analyses. The objective of this study is to present a systematic literature review to investigate the recent developments in the field of security integration in Software Development LifeCycle (SDLC) through articles published in one and a half decades and to propose a way forward. This systematic literature review (SLR) follows the review protocol [Proposed by Kitchenham]. The review has been divided into three main stages including planning, execution, and analysis, as per the protocol. From the selected 23 x articles , it is clear that need of a collaborative approach is necessary for addressing Critical Software Security Risks (CSSRs) through effective risk management/ estimation technique like protection poker. Quantifying risks using a numeric scale enables a comprehensive understanding of their severity, facilitating focused resource allocation and mitigation efforts. Through a comprehensive understanding of potential vulnerabilities and proactive mitigation efforts facilitated by protection poker, organizations can prioritize resources effectively and ensure the successful outcome of projects and initiatives in today's dynamic threat landscape.

**INDEX TERMS** Software Development LifeCycle (SDLC); systematic literature review; Critical Software Security Risks (CSSRs)

1. **INTRODUCTION**

In the current era of 5G revolution where mobile phones are easily accessible to all masses, cyber security has gained paramount importance. Security attacks in the cyber domain have increased significantly and hence calls for increased protective measures. Security in Application Development has gained substantial importance over the recent times owing to hacking and other attacks on computer systems. As a result, security has to be made intrinsic component of all the stages of mobile application development. If sensitive data is hacked or exposed, it may cause un-repairable loss to software companies repute with partners, customers and investors, therefore system managers and users are paying more and more attention to this important aspect of security. In order to have complete secure applications/ softwares, security aspects needs to be considered into every step of Software Development Life Cycle (SDLC) or Application Development Life Cycle (ADLC). To blend security into the software engineering model, it should be considered from the beginning of the SDLC/ ADLC [1]. Most organizations normally see security as a post-development process. Security concerns have a profound impact on the overall quality of software, as organizations grappling with insecure software are compelled to address and rectify unreliable applications, while also impeding the progress of other concurrent projects. Emerging cyber vulnerabilities, both internal and external to the organization, continue to surface persistently, posing formidable financial risks and jeopardizing the integrity of critical company data. The repercussions of such security lapses extend beyond monetary losses, encompassing substantial credibility deficits that can be detrimental to the reputation and standing of the organization [2].

The main purpose of this research is to evaluate the Protection Poker technique's efficacy because of risk assessment tool for CSSRs revealed in Security Assurance Model (SAM) [3]. This type of assessment can help to improve the method used to indicate the areas where under development program needs special consideration/ deliberation. Software companies can now estimate their security parameters and their abilities to create more secure softwares/ applications. Considering this, it is important to identify collaborative techniques to address the risk estimation of the critical security risks involved with software development.

This review aims to provide a comprehensive overview of the current state of research on integrating security aspects in the software development phase and to identify opportunities for future researchers to improve the precision and effectiveness of risk estimation methods in order to accurately prioritize resources to mitigate those risks. To compare the performance of different methods and evaluate their effectiveness, an SLR has been conducted to identify the

* 1. Current trends of research in this field,
  2. Traditional Security Integrating Methods,
  3. Efficacy of Protection Poker technique,
  4. Current gaps and future work.

In this study, the literature on traditional security integrating models in SDLC has been reviewed to discuss the challenges and limitations of these methods, such as the lack of quantifiable parameter to address each risk involved. Moreover, giving security an utmost priority while developing any software/ application is the need of the hour to feel more secure in the cyber world.

In this systematic literature review. Challenges and limitations of this traditional secure software development frameworks like NIST secure SDLC, BSA Framework for Secure Softwarehave also been highlighted. Articles reviewed are published between 2010 and 2023 from prominent digital libraries, including Elsevier, IEEE, ACM, and Web of Science. Inclusion and exclusion criteria based on period, Relevance, Impact & accessibility to select relevant articles for the review have been considered.

This paper is structured in the following way:

Existing literature on practiced secure software development frameworks/ model in Section II. The proposed methodology is discussed in Section III. Section IV presents the results. In the end, Section V concludes this study and provides future work.

1. **LITERATURE REVIEW**

Security consideration while developing a software has been a growing research field specially in the last decade or so. It is being

1. ***SECURITY THROUGH LENS OF VARIOUS PHASES OF SDLC***

* **Initial Planning and Design**: Threat modeling and secure design are the two main cardinals of this phase. Security threats are resolved through threat modeling by devising counter measures for them, while secure design principles calls for such application design which ensures enhanced security.
* **Implementation**: Secure coding strategies/ code reviews are the two main activities performed in this phase to ensure that code is functionally correct and resistant to known vulnerabilities.
* **Testing**: Known secure testing methods, such as dynamic and static analysis, are used to reveal and

identify susceptibilities that may have gone unnoticed in early phases. It also includes code reviews and penetration testing.

* **Deployment:** In order to keep application’s security operational. Secure deployment practices and vulnerability management are key mechanisms that ensures the operation-ability of application with security mechanism intact.
* **Maintenance**: Constant monitoring and incident response are vital for maintaining the application's security post-deployment.

"Integrating security" mentions the systematic addition of security measures and concerns at each phase of the SDLC. This scheme is essential for making applications that are resilient to cyber attacks, protecting users and software producers from any losses and breaches. Integration of security to SDLC is not only a technical requirement but it can also be termed as a key business decision as primarily it serves to reduce weaknesses and post-deployment security fix costs. As a normal practice, historically security was often neglected/ ignored throughout the creation of applications. This reactive scheme has inherent probability of more susceptibilities and exploitation threats. With security-centric methodologies like DevSecOps gaining reputation for integrating security into every stage of the software ***development process, secure*** practices in SDLC have evolved over a short span of time to become more proactive

* **Strategies and Practices**. Coding while keeping security risks in mind helps to end vulnerabilities at the start, this is known as secure coding techniques. Coding standards must be strictly followed in order to prevent security related issues, Moreover, code analysis procedures must also be put to use in order to address security problems.[4]
* **Security Requirements Engineering**. Security requirements must be properly logged, deliberated upon and documented in the initial development stage. It ensures that program is built while keeping security consideration alive during all phases of development [5].

1. ***DEVSECOPS***

DevSecOps automates security integration at all phases of the DevOps life cycle, from initial design to integration, testing, deployment, and delivery. However, integration of security throughout the DevOps process has its own set of challenges [4]. One of such challenges is making security adapt to the DevOps processes, which will mean the security methods need to be highly agile, and the procedures must be accepted and understood by the security, development, and operations teams [6]. Another challenge is how the organization will start the DevSecOps practices by readily adopting the change in skills, tools, standards, and processes to successfully implement security into their culture [7]. Additionally, automation and being conversant with relevant tools and technology is an ongoing challenge for DevSecOps projects. The dynamic nature of the environment implies that

the necessary security functionality should be ready for use in tools that work on the right platforms.

1. ***NIST FRAMEWORK FOR SECURE SOFTWARE DEVELOPMENT***

With the increase in cybercrimes over the last few years, a growing realization for the need for cyber security has begun. Unfortunately, being aware that cyber security is something that needs to be worried and knowing what steps to take are two different things entirely. In the United States, the National Institute of Standards and Technology (NIST) developed the Cyber Security Framework (CSF) to assist critical infrastructures in determining what they need in order to secure their computer systems and networks. While aimed at organizations, much of the guidance provided by the CSF, especially the basic functions it identifies, are also valuable for communities attempting to put together a community cyber security program [8].

The framework core functions i-e Govern, Identify, Protect, Detect, Respond, and Recover organize cyber security outcomes at their highest level.

* **Govern** : Establish and monitor the organization’s cyber security risk management strategy, expectations, and policy. The Govern Function is cross-cutting and provides outcomes to inform how an organization will achieve and prioritize the outcomes of the other five Functions in the context of its mission and stakeholder expectations. Governance activities are critical for incorporating cyber security into an organization’s broader enterprise risk management strategy. Govern directs an understanding of organizational context; the establishment of cyber security strategy and cyber security supply chain risk management; roles, responsibilities, and authorities; policies, processes, and procedures; and the oversight of cyber security strategy.
* **Identify** : Help determine the current cyber security risk to the organization. Understanding its assets (e.g, data, hardware, software, systems, facilities, services, people) and the related cyber security risks enables an organization to focus and prioritize its efforts in a manner consistent with its risk management strategy and the  
  mission needs identified under Govern. This Function also includes the identification of improvements needed for the organization’s policies, processes, procedures, and practices supporting cyber security risk management to inform efforts under all six Functions.
* **Protect** : Use safeguards to prevent or reduce cyber security risk. Once assets and

risks are identified and prioritized, Protect supports the ability to secure those assets to prevent or lower the likelihood and impact of adverse cyber security events. Outcomes covered by this Function include awareness and training; data security; identity management, authentication, and access control platform security (i.e., securing the hardware, software, and services of physical and virtual platforms) and the resilience of technology infrastructure.

* **Detect** : Detect enables timely discovery and analysis of anomalies, indicators of compromise and other potentially adverse cyber security events that may indicate that cyber security attacks and incidents are occurring.
* **Respond** : Take action regarding a detected cyber security incident. Respond supports the ability to contain the impact of cyber security incidents. Outcomes within this function cover incident management, analysis, mitigation, reporting, and communication.
* **Recover** : Restore assets and operations that were impacted by a cyber security incident. Recover supports timely restoration of normal operations to reduce the impact of cyber security incidents and enable appropriate communication during recovery efforts [8].

**Fig 1 : NIST Cyber Security Framework**

1. **METHODOLOGY**

It is imperative to follow a systematic approach in any SLR to minimize biases in the research. This SLR follows the review protocol suggested by Kitchenham [9] as depicted in the flowchart, Figure 2. The review has been divided into three main stages including planning, execution, and analysis, as per the protocol.

1. ***STRUCTURAL OVERVIEW***

As per the Kitchenham’s protocol [9], research methodology has been implemented as second stage to identify related articles based on relevance parameter with research field.

Table 1: Pros and cons of cost estimation techniques discussed in literature.

|  |  |  |
| --- | --- | --- |
| FrameWorks / Models | Pros | Cons |
| Secure SDLC | * By Integrating security into each phase of SDLC ensures vulnerabilities are identified and mitigated early, reducing the likelihood of security breaches * Cost effective by early identification and fixing of security risks. * Regulatory and compliance requirements related to software security are met by adhering to secure SDLC methods | * Complexity is added to the development process while implementing Secure SDLC * Process is resource intensive and demands additional resources in the shape of time, personnel, and financial investment * Additional focus on security can delay the development process * It does not sufficiently satisfies the requirements of volatile and changing requirement |
| DevSecOps | * Continuous Security Integration * Enhanced Collaboration between different teams i-e design, coding and testing etc, yields better performance * DevSecOps practices are highly scalable as per the software requirements demands | * Diversity required in skillset among team members * In order ot adapt DEvSecOps fully, organizations may require to have cultural change, which may leads to resistance with in some organizations |
| NIST | * Consistency and reliability in security practices is achieved through well-composed standards and guidelines * Credibility of NIST guidelines is immense and are highly respected/ trusted in the industry * Highly adaptable and can be used for various organizational needs and sizes, providing huge flexibility | * The wide-range nature of NIST guidelines can be overwhelming and challenging to implement fully * NIST guidelines are prone to overfitting, especially for ML models within the NIST framework * NIST framework can be highly inaccurate with noisy data, hence performance is degraded exponentially |

Table 2: Strengths and limitations of a research articles.

|  |  |  |
| --- | --- | --- |
| Ref. | Strengths | Weaknesses |
| [3] | * Comprehensive analysis of securtiy risks posed to software development * It is a comprehensive guide for a general reader. | * The paper lacks a comparative analysis of the techniques studied * Long tabulated data based on RQs have been at times monotonous |
| [16] | * A good guide on understudy risk estimation method for general readers * Proposed method has been explicitly explained in a detailed manner. | * Complete evaluation of various risk estimation techniques being followed in different organizations not done * Results were dependent on individual personality behaviors and hence may be bias * Numerical results have not been supported with a graphical representation which reduces general comprehension. |
| [23] | * Results have been supported with figures which increases the comprehension of readers. * Results have been implemented and verified on a real working software | * Not very well-referred article with just 16 cited articles. * No mentioin/ use of automated tools for having collected present status of security measurement |

1. ***OVERVIEW OF ARTICLES SELECTED***

Engines that are searched to obtain references for the articles on security assessment in software development are as follows

* + IEEE Xplore
  + Springer
  + ScienceDirect
  + Emarlds

These search enginres have been selected based on their reputation for hosting high-quality scientific literature in the field of computer software, including software development and cyber security [10]. Furthermore, the inclusion and exclusion criteria have been applied to ensure that the articles selected for review meet specific quality and relevance standards. They have been accessed in two stages namely, primary and secondary as shown in Figure 2.

***C. INCLUSION CRITERIA***

**Fig 2 : Overview of Article Selected**

The inclusion criteria are based on various factors [9] for selecting the articles.

* **Period**: Articles published between the years 2010 and 2023 are considered for this literature review.
* **Source** : Articles from well reputable digital libraries such as IEEE, Springer, Science Direct etc are included in the review.
* **Language**: Only articles published in the English lan- guage are included in the review.
* **Relevance**: Articles that directly address the research field and meet the mentioned inclusion criteria are included in the review [11].
* **Accessibility** : Articles easily available and in full text form are generally included.

1. ***DATA MANAGING AND REFERNCING***

In data sifting all the data is organized as per the libraries and it is ensured that all articles are complete. Data is preprocessed through Mendely for further organizing as per requirements. It is a popular tool among academics and scholars because it makes managing research materials simple and efficient. Users can import articles directly from databases, websites, and other sources Mendely is a free and open-source tool Integra table with MS World and allows researchers to save and organize articles, annotate them, and generate bibliographies [12],[13].

1. ***EXECUTION OF PROPOSED METHODOLOGY***

The methodology employed in this research involves the assessment and analysis of 46 security risks that have been identified through the security assurance model [3]. To estimate the levels of risk associated with these vulnerabilities, the protection poker technique is utilized. It involves an organized and efficient strategy to assess the likelihood and consequences of each risk, leading to an in-depth understanding of potential threats. It generally involves quantifying a risk level associated with every identified security concern. Following stages are involved.

* **Risk Estimation.** In order to effectively evaluate the risks linked to a specific circumstance, it is imperative to measure each risk using a numerical scale that varies from 1 to 10. This scale terms 1 as the minimum risk level whereas 10 as the maximum one. Linking of a numerical figure to each risk helps in the evaluation of both risk’s probability of occurrence and its probable impact [14].
* **Data Analysis.** In order to assess risks level in a company, it is crucial to compare them against industrial/ recommended practices. This comparison enables software houses/ firms to identify areas that need additional stringent measures and also in the formulation of efficient risk mitigation plans. The process involves pinpointing risks with the highest scores. These risks represent the most significant threats to the organization and demand urgent action. By prioritization of these risks, companies can optimize their resource allocation and can focus on resolving the most critical concerns[15][16].
* **Data Collection**. The survey is conducted among three software firms (AppInSnap, Omnisoftex & Centangle Interactive) to have their insights on how they address the identified risks during the application development process. These responses are being utilized to comprehend how industry practices manage risks and validate the risk assessments. Through distributing the survey and analyzing the replies, a comprehensive understanding of risk management practices in the industry is being acquired, ensuring the accuracy of risk assessments. This interactive procedure provides a detailed examination of how various software firms handle security risks in their application development procedures[17]
* **Synthesis and Analysis.** Risk assessment process’s output and feedback from the software firms are analyzed. Data obtained is meticulously reviewed to identify potential risks and grasp the feedback from the software companies. This feedback obtained from software firms is useful in guiding decision-making. Merger of response from software firms with that of risk assessments gathered through protection poker, helps in better understanding of security requirements, hence leading to the resolution of security issues early in the development phase [18],[19].

1. **RESULTS AND DISCUSSIONS**

Empirical research format is used for data collection and analysis to address this research, which are consistent throughout the adopted methodology. While qualitative research emphases on collecting non-numerical data that can be analyzed for obtaining patterns [20], quantitative research often gathers numerical data which can then be analyzed statisticallyv[21]. In the case of this research, it is primarily qualitative in nature, as the goal is to analyze and compare the effectiveness of SAM model through the application of Protection Poker Technique as risk assessment tool [22]

1. **CONCLUSION AND FUTURE WORK**

A detailed literature review has been given on software security and its role during designing and various development phases of the software. a proactive approach to risk management, ensuring thorough examination of potential threats to software security is need of the hour. The methodology emphasizes the significance of identifying and prioritizing risks, facilitating the allocation of resources based on severity. Moving forward, future research could probe deeper into exploring the additional manpower or appointments required to effectively mitigate and resolve security risks throughout the development process which would indeed provide valuable insights into resource allocation strategies for enhancing overall security.

**REFERENCES**

1. E. Khanna, R. Popli and N. Chauhan, "Identification and Classification of Risk Factors in Distributed Agile Software Development," in Journal of Web Engineering, vol. 21, no. 6, pp. 1831-1851, September 2022.
2. P. Singal, A. C. Kumari, and P. Sharma, “Estimation of software development effort: A differential evolution approach,” Procedia Computer Science, vol. 167, pp. 2643–2652, 2020.
3. R. A. Khan, S. U. Khan, M. Alzahrani and M. Ilyas, "Security Assurance Model of Software Development for Global Software Development Vendors," in IEEE Access, vol. 10, pp. 58458-58487, 2022.
4. Myrbakken, H., Colomo-Palacios DevSecOps: A Multivocal Literature Review. In: Mas, A., Mesquida, A., O'Connor, R., Rout, T., Dorling, A. (eds) Software Process Improvement and   
   Capability Determination. SPICE. Communications in Computer and Information Science, Cham vol 770, 2017.
5. Samar Al-Saqqa, Samer Sawalha and Hiba AbdelNabi “Agile Software Development: Methodologies and Trends”, International Journal of Interactive Mobile Technologies, Vol. 14, No. 11, pp 246-270, 2020.
6. Goldschmidt, Murray, and Michael McKinnon. Devsecops-agility with security. Technical report, Sense of Security, 2016.
7. Juncal Alonso, Radoslaw Piliszek, and Matija Cankar.. Embracing IaC Through the DevSecOps Philosophy: Concepts, Challenges, and a Reference Framework. IEEE Software 40, 1 pp 56-62, 2023.
8. Gregory B. White and Natalie Sjelin. "The NIST Cybersecurity Framework." Establishing Cyber Security Programs Through the Community Cyber Security Maturity Model (CCSMM), IGI Global, 2021, pp.171-192.
9. B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering– a systematic literature review,” Information and software technology, vol. 51, no. 1, pp. 7–15, 2009
10. C. Onwubiko, "Security operations centre: Situation awareness, threat intelligence and cybercrime,"  International Conference on Cyber Security And Protection Of Digital Services (Cyber Security), London, UK, pp. 1-6, 2017.
11. J. Shah and N. Kama, “Extending function point analysis effort estimation method for software development phase,” in Proceedings of the 2018 7th International Conference on Software and Computer Applications,

pp. 77–81, 2018.

1. Panda, Subhajit. "Reference Management Software for Assisting Researchers: A Comparative Analysis of Usage and Usability." Library Technology with New Perception (2023): 191-206.
2. Childress, Dawn. "Citation tools in academic libraries: Best practices forreference and instruction." Reference and User Services Quarterly 51.2, pp 143-152, 2011.
3. Mamdouh Alenezi and Sadiq Almuairfi “Security Risks in the Software Development Lifecycle” International Journal of Recent Technology and Engineering, Vol 8 (3), 2019.
4. W. Wang, Q. Zeng, and A. P. Mathur, "A Security Assurance Framework Combining Formal Verification and Security Functional Testing," in Proceedings of the 12th International Conference on Quality Software, Xi'an, China, 2012.
5. I. A. Tøndel, M. G. Jaatun, D. S. Cruzes, and L. Williams, "Collaborative security risk estimation in agile software development," Information and Computer Security, vol. 27, no. 4, pp. 508-535, 2019.
6. K. Qian, R. M. Parizi and D. Lo, "OWASP Risk Analysis Driven Security Requirements Specification for Secure Android Mobile Software Development," IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, pp. 1-2, 2018.
7. S. -J. Chen, Y. -C. Pan, Y. -W. Ma and C. -M. Chiang, "The Impact of the Practical Security Test during the Software Development Lifecycle," 24th International Conference on Advanced Communication Technology (ICACT), Pyeong Chang, Republic of Korea,pp 313-316, 2022.
8. M. G. Stochel, “Reliability and accuracy of the estimation   
     
     
     
     
     
   process- wideband delphi vs. wisdom of crowds,” in 2011 IEEE 35th Annual Computer Software and Applications Conference, pp. 350–359, IEEE, 2011.
9. Y. Zeng, Y. Cheng, G. Xie and R. Wang, "Design of Mobile Application Lifecycle Security Management Platform,"  International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 2021
10. A. I. Al-Darwish and P. Choe, "A Framework of Information Security Integrated with Human Factors," in HCI for Cybersecurity, Privacy and Trust, A. Moallem, Ed. Cham, Switzerland, 2019.
11. O. Kovalenko, O. Smirnov, A. Kovalenko and S. Kavun, "Quantitative Risk Assessment Method Development in the Context of the SDLC-model," IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021.
12. A. R. Shehab Farhan and G. M. Mostafa Mostafa, "A Methodology for Enhancing Software Security During Development Processes,"  21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 2018.