# R3 -freeAI-updated.docx

Assignment

Class

Organization

## Document Details

Submission ID

trn:oid:::1:2935022023

Submission Date

May 30, 2024, 1:01 PM UTC

Download Date

May 30, 2024, 1:02 PM UTC

File Name

uploads_5183_2024_05_30_R3_-freeAI-updated_891bbdc700c5d32c.docx

File Size

67.5 KB

Pages

Words

Characters

**How much of this submission has been generated by AI?**

**69%**

of qualifying text in this submission has been determined to be generated by AI.

**Caution: Percentage may not indicate academic misconduct. Review required.**

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

## Frequently Asked Questions

**What does the percentage mean?**
The percentage shown in the AI writing detection indicator and in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was generated by AI.

Our testing has found that there is a higher incidence of false positives when the percentage is less than 20. In order to reduce the likelihood of misinterpretation, the AI indicator will display an asterisk for percentages less than 20 to call attention to the fact that the score is less reliable.

However, the final decision on whether any misconduct has occurred rests with the reviewer/instructor. They should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in greater detail according to their school's policies.

**How does Turnitin's indicator address false positives?**
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be AI-generated will be highlighted blue on the submission text.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

**What does 'qualifying text' mean?**
Sometimes false positives (incorrectly flagging human-written text as AI-generated), can include lists without a lot of structural variation, text that literally repeats itself, or text that has been paraphrased without developing new ideas. If our indicator shows a higher amount of AI writing in such text, we advise you to take that into consideration when looking at the percentage indicated.

In a longer document with a mix of authentic writing and AI generated text, it can be difficult to exactly determine where the AI writing begins and original writing ends, but our model should give you a reliable guide to start conversations with the submitting student.

**Disclaimer**
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify both human and AI-generated text) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

# INTEGRATING SECURITY IN APPLICATION DEVELOPMENT LIFECYCLE USING PROTECTION POKER TECHNIQUE

**Abstract :** This research work presents a methodology for effectively addressing critical software security risks (CSSRs) through effective risk management techniques, particularly focusing on the application of protection poker. Beginning with a meticulous identification and assessment of the specified risks, the methodology utilizes protection poker to collaboratively evaluate the likelihood and impact of each risk, fostering informed decision-making and prioritization. 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.

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 purpose of the research is to evaluate the Protection Poker technique's efficacy as a risk assessment tool for CSSRs found in the Security Assurance Model (SAM) [3]. This estimation would allow to indicate the areas where the program under development needs special consideration. Software companies can now evaluate their security parameters and their abilities to create more secure applications.

## 2. LITERATURE REVIEW.

**Definition and Importance of Integrating Security.** The term "integrating security" refers to the methodical addition of security measures and concerns at each phase of the application development life cycle (ADLC). This method is essential for building applications that are resistant to cyberattacks, shielding users and software producers from any losses and breaches. Security integration in ADLC is not just a technical requirement but also a strategic business decision as it largely reduces vulnerabilities and post-deployment security fix costs. In the past, security was frequently neglected throughout the creation of applications, often being addressed until after the software was designed or even after it was put into use. There were more vulnerabilities and exploitation threats as a result of this reactive strategy. With approaches like DevSecOps garnering popularity for incorporating security into every stage of the software development process, security practices in ADLC have evolved over time to become more proactive.

(i) **Strategies and Practices**. Writing code with security in mind and planning to stop vulnerabilities at the source is known as secure coding techniques. Coding standards must be followed in order to prevent typical security mistakes, and code analysis techniques must be used in order to find and fix security problems [4].

(ii) **Security Requirements Engineering**. Early in the development process, security requirements must be determined and documented. It guarantees that the program is created with these specifications in mind, integrating security into the program's core architecture [5].

Risk assessment within the Application Development Life Cycle (ADLC) involves the evaluation of potential risks linked to security threats and vulnerabilities. Techniques like Failure Modes and Effects Analysis (FMEA are utilized to systematically examine possible failure points and their impacts on application security.

### a. Security at Various Phases of ADLC

**Initial Planning and Design**: During this stage, threat modeling and secure design principles play a crucial role. Threat modeling aids in identifying security threats and devising countermeasures, while secure design principles direct the application's architecture for enhanced security.

**Implementation**: Secure coding guidelines and code reviews are indispensable in this phase to ensure that the code is functionally accurate and resilient against known vulnerabilities.

**Testing**: Security testing methods, such as static and dynamic analysis, are employed to uncover vulnerabilities that may have been overlooked in earlier stages. This includes penetration testing and code reviews.

**Deployment:** Secure deployment practices and vulnerability management are vital at this point to uphold the application's security in its operational environment.

**Maintenance**: Continuous monitoring and incident response are essential for preserving the application's security post-deployment.

This encompasses regular updates and patches to address emerging vulnerabilities.

**Types and Lifecycle of Cross-Platform Application Development Platforms**: Cross-platform development involves crafting software applications that are compatible with multiple operating systems or platforms, using a unified codebase. This approach presents several advantages such as Cost-Effectiveness, Time Efficiency, and Broader reach.

## Popular Cross-Platform Frameworks

a) **React Native**: Developed by Facebook, React Native enables developers to build mobile apps utilizing JavaScript and React. It inherits security concerns associated with JavaScript, like cross-site scripting (XSS). Additionally, reliance on third-party libraries can expose vulnerabilities.

b) **Flutter**: Originated by Google, Flutter is an open-source UI software development kit for creating natively compiled applications for mobile, web, and desktop from a single codebase. While Flutter apps are relatively secure, they face risks related to insecure data storage and communication. The framework's novelty also means fewer security resources and community support compared to more established frameworks.

c) **Xamarin**: Owned by Microsoft, Xamarin utilizes C# for developing Android and iOS apps. Xamarin apps may be prone to typical .NET security issues. Challenges include ensuring secure data storage, proper cryptography usage, and secure communication.

## Evaluating Security Considerations with Native Platforms

a) **Code Security**: Cross-platform apps need to handle security across many environments, whereas native apps need platform-specific security mechanisms [13].

b) **Data Storage**: Shared security practices, which might not be as reliable as platform-specific ones, are frequently used by cross-platform applications to store data [14].

c) **API Security**: Since cross-platform apps frequently contact the same back end from many platforms, API security is essential.
d) **Regular Updates**: Cross-platform applications may encounter difficulties updating simultaneously on all platforms, which could impact the uniformity of security patches [15].

a. <u>**Potential Research Gap**</u>. A potential gap in quantitatively evaluating the severity of 46 x CSSRs discovered by a SAM during the ADLC is revealed by the literature review. While various security practices and risk assessment methodologies exist, a collaborative approach to estimate the impact of these risks on application security seems to be lacking. 46 x security risk identified by Security Assurance Model are given below :-

| Ser | Issue |
|---|---|
| a. | Injection |
| b. | Broken authentication and session management |
| c. | Cross-site scripting (XSS) |
| d. | Insecure direct object references |
| e. | Security mis-configuration |
| f. | Sensitive data exposure |
| g. | Missed function-level access control |
| h. | Cross-site request forgery (CSRF) |
| i. | Using components with known vulnerabilities |

| Ser | Issue |
|---|---|
| j. | Un-validated redirects and forwards |
| k. | Insufficient logging and monitoring |
| l. | Broken access control |
| m. | Improper input user data |
| n. | Buffer overflow |
| o. | Improper error handling |
| p. | Race conditions |
| q. | Failure to restrict URL access |
| r. | Insufficient transport layer protection |
| s. | Server-side request forgery (SSRF) |
| t. | Insecure cryptographic storage |
| u. | Session fixation |
| v. | Poor password policy and management |
| w. | Improper authorization |
| x. | Click-jacking |
| y. | Excessive data exposure |
| z. | Broken anti-automation defenses |
| aa. | Improper file and resource protection |
| bb. | Not enough security configuration |

| Ser | Issue |
|---|---|
| cc. | Broken business logic |
| dd. | Improper certificate validation |
| ee. | Time and state-related attacks |
| ff. | Fail to restrict upload of dangerous file types |
| gg. | Insufficient session expiration |
| hh. | Cryptographic issues |
| ii. | Insecure communications |
| jj. | Inadequate encryption strength |
| kk. | Authentication bypass |
| ll. | Improper use of a security feature |
| mm. | XML external entities (XXE) |
| nn. | Insufficient security controls in a third-party service |
| oo. | Missing security headers |
| pp. | Broken cryptography usage |
| qq. | Improper asset management |
| rr. | Out-of-band channel exploitation |
| ss. | Security relevant mis-configuration |
| tt. | Elevation of privilege |

**Proposed Solution**. The proposed solution necessitates engaging with software companies to assess the effectiveness of the protection poker technique in reducing identified critical software security risks. Feedback from these companies is gathered through a questionnaire. In Protection Poker, participants gauge the impact (probability and consequence) of security risks using playing cards. This game-like method encourages conversation and promotes a collective comprehension of security risks among the development team. By integrating Protection Poker into the ADLC, development teams could jointly evaluate the seriousness of CSSRs highlighted by a SAM, allowing them to prioritize security efforts and resources on the most significant risks. The integration of Protection Poker into the ADLC involves the following steps:

1. **Identify CSSRs**: A SAM would be utilized during the security assurance phase to pinpoint potential CSSRs in the application being developed.
2. **Plan a Protection Poker Session**: Organize a session for Protection Poker involving developers, security experts, and other relevant stakeholders.
3. **Assess Risk Impact**: Each CSR would be deliberated upon, and participants would utilize playing cards to determine the likelihood and consequence of the risk. The aggregated card values would be used to produce a risk score.
4. **Prioritize Security Focus**: With the risk scores allocated through Protection Poker, development teams can prioritize their security endeavors towards the most critical CSSRs.

**Research Questionnaire:**

1. **Identifying Risks:** How does your software house identify the 46 security risks outlined in the security assurance model during the application development lifecycle?

2. **Risk Management:** What strategies or methods do you use to manage and mitigate the identified risks?

3. **Risk Assessment:** How do you integrate risk assessment methods like the protection poker technique into your application development process?

4. **Risk Prioritization:** Based on the risk estimation provided, how do you prioritize risks and allocate resources for their mitigation?

5. **Protection Techniques:** Can you share your experiences with using protection poker or similar techniques to assess and address these risks during application development?
6. **Challenges and Solutions:** What specific challenges have you faced in managing the aforementioned risks, and what strategies have you found most effective in addressing them?

**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. To estimate the levels of risk associated with these vulnerabilities, the protection poker technique is utilized. This method presents a structured and systematic strategy to assess the probability and consequences of each risk, leading to a thorough comprehension of potential threats. It also facilitates the measurement of the risk level associated with every identified security concern. The procedure encompasses the subsequent stages.

**Risk Identification and Assessment:**

a. The procedure begins with carefully identifying 46 risks outlined in the security assurance model. These risks cover a wide range of potential vulnerabilities and threats that could jeopardize the security and integrity of the application. By investigating the characteristics of each risk, we develop a deeper understanding of their nature, extent, and potential consequences for the application's overall security stance.

b. Furthermore, it is crucial to investigate the underlying root causes of each risk as this knowledge is pivotal in creating effective mitigation strategies. By identifying the factors that contribute to these risks, we can proactively tackle vulnerabilities and enhance the application's resilience against potential security breaches.

c. Equally important is evaluating each risk using the protection poker method, a collaborative technique that harnesses the expertise of team members well-versed in security and development domains. This method involves a structured discussion where team members assess the probability and impact of each risk, drawing on their collective insights and experiences to reach a consensus.

d. Through the protection poker approach, team members participate in a thoughtful process that encourages critical thinking and well-informed decision-making. By evaluating the probability of occurrence and the potential impact of each risk, team members can prioritize their mitigation efforts and allocate resources wisely to address the most critical security issues.

1. **Risk Estimation:**

a. To effectively assess the risks linked to a specific circumstance or venture, it is crucial to measure them using a numerical scale that commonly spans from 1 to 10. The scale defines 1 as the minimum risk level and 10 as the maximum. Assigning a numerical figure to each risk streamlines the evaluation of both its probability of occurrence and potential impact.

b. The methodology of quantifying risks necessitates the contemplation of diverse factors influencing their overall seriousness. This involves scrutinizing the likelihood of a risk materializing and the extent to which it could impact the intended outcome. Deconstructing risks into two elements - likelihood and impact - offers a more thorough insight into their possible repercussions.

c. Following the assessment and numerical assignment of likelihood and impact for each risk, the subsequent phase involves merging these assessments to form an overall risk score. This unified score serves as a clear indicator of the risk level associated with each potential issue, aiding decision-makers in prioritizing their responses.

d. By adhering to this methodical approach to risk evaluation, organizations can enhance their comprehension of the potential hazards they confront and take preemptive actions to alleviate them. Ultimately, quantifying risks through a numerical scale enhances informed decision-making and promotes the successful fruition of projects and endeavors.

2. **Data Analysis:**

a. When assessing risk levels in a company, it is essential to compare them against industry benchmarks and recommended practices. This comparison enables firms to grasp their position relative to competitors and pinpoint areas that need additional scrutiny. This approach facilitates a thorough evaluation of potential risks and the formulation of efficient risk mitigation plans.

b. A critical step in this assessment involves pinpointing risks with the highest projected scores. These risks represent the most significant threats to the organization and demand urgent action. By giving priority to these risks for in-depth analysis and mitigation plans, companies can optimize their resource allocation and concentrate on resolving the most critical concerns.

a. **Data Collection from Software Houses:**

The survey was circulated among three software firms (AppInSnap, Omnisoftex & Centangle Interactive) to collect their insights on addressing the identified risks during the application development process. The 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 method enables a detailed examination of how various software firms handle risks in their application development procedures, offering valuable insights into risk management in software development research.

b. **Synthesis and Analysis** . The outcomes of the risk assessment process and the input from the software firms are analyzed. The data is meticulously reviewed to identify potential risks and grasp the feedback from the software companies. This evaluation requires a thorough assessment and consideration of the gathered information. The feedback from the software firms is being considered to guide decision-making. The risk assessment is being carried out to evaluate potential threats and formulate strategies to mitigate these risks. Ultimately, the conclusions drawn from this analysis will provide significant insights for the research. By merging the feedback from the software firms with the risk assessments (through protection poker), the study aims to gain a better understanding of industry practices and comprehend security requirements to address them effectively in the early stages of development.

## RESULTS

### Response from AppInSnap:

**Identifying Risks**:

Throughout the application development lifecycle, AppInSnap uses a proactive methodology to detect the 46 security risks specified in the security assurance model, they carry out comprehensive risk assessments by employing industry-standard standards and criteria. The team works closely with clients to comprehend their unique security needs and any weak points. They also use manual code reviews and automated security testing technologies to find any possible vulnerabilities during the development process.

**Risk Management:**

The core of AppInSnap's risk management approach is putting strong security measures in place during the whole development lifecycle. To successfully reduce hazards that have been detected, they adhere to industry standards and best practices, such as the OWASP guidelines. To quickly handle new threats and weaknesses, the team constantly checks and upgrades security protocols. They also regularly do penetration tests and security audits to guarantee the integrity of their apps.

.

### Risk Assessment:

The protection poker strategy and other risk assessment techniques are incorporated into the application development process at AppInSnap. With the use of this technique, the possibility and consequences of every risk (potentially a security concern) was detected, guaranteeing a thorough grasp of possible dangers. Stakeholders from a variety of fields, such as development and security, might be involved to better prioritize risks and distribute resources.

### Risk Prioritization:

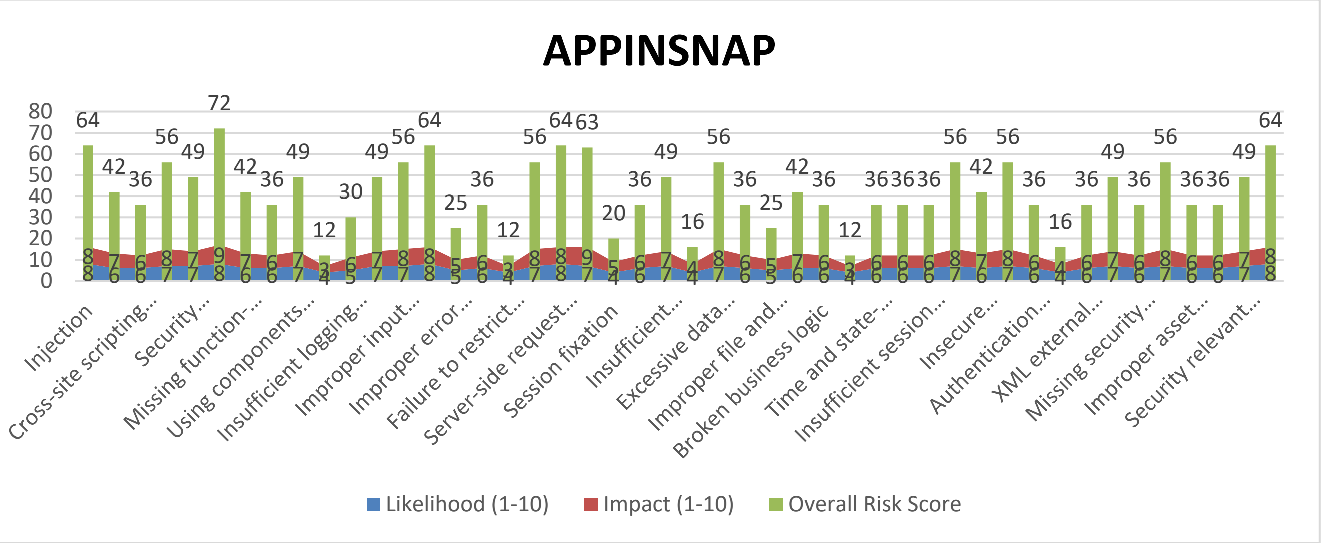
AppInSnap ranks risks based on how they might affect the application and organization, estimating risks using the protection poker technique. Areas deemed high-risk receive prompt attention, and resources are allotted for appropriate mitigation measures. To maintain a proactive approach to risk management, the team periodically analyzes and updates risk priorities as the project moves forward.

### Protection Techniques:

AppInSnap explicitly applied the protection poker technique for the 46 threats listed in the Security Assurance Model for the purpose of this research. They can efficiently prioritize mitigation activities and make well-informed judgments because to protection poker's structured methodology.

### Challenges and Solutions:

AppInSnap has faced difficulties in managing security concerns, including keeping up with changing threats and striking a balance between security and development schedules. Their approach to overcoming these security challenges involves investing in training and education for their staff, keeping abreast of the latest security trends, and incorporating flexible security methods. Furthermore, establishing a culture of security awareness and cooperation within their company has proven to be a successful tactic in reducing risks.



## Response from OmniSoftex:

### Identifying Risks:

OmniSoftex has established processes and protocols to identify the 46 security risks outlined in the security assurance model during the application development lifecycle. They conduct comprehensive threat modeling exercises, code reviews, and security assessments to identify potential vulnerabilities. The team collaborates closely with clients to understand their security requirements and prioritize risks based on their potential impact.

### Risk Management:

In order to detect the 46 security risks listed in the security assurance model throughout the application development lifecycle, OmniSoftex has set up procedures and guidelines. To find potential vulnerabilities, they carry out thorough threat modeling exercises, code reviews, and security assessments. The team works closely with clients to comprehend their security needs and rank risks according to possible consequences. **Risk management**: OmniSoftex's approach to risk management is centered on putting in place a multi-layered security framework in order to successfully reduce recognized risks. They secure the security of their applications by adhering to industry standards and best practices, including ISO 27001 and NIST.

**Risk Prioritization:**

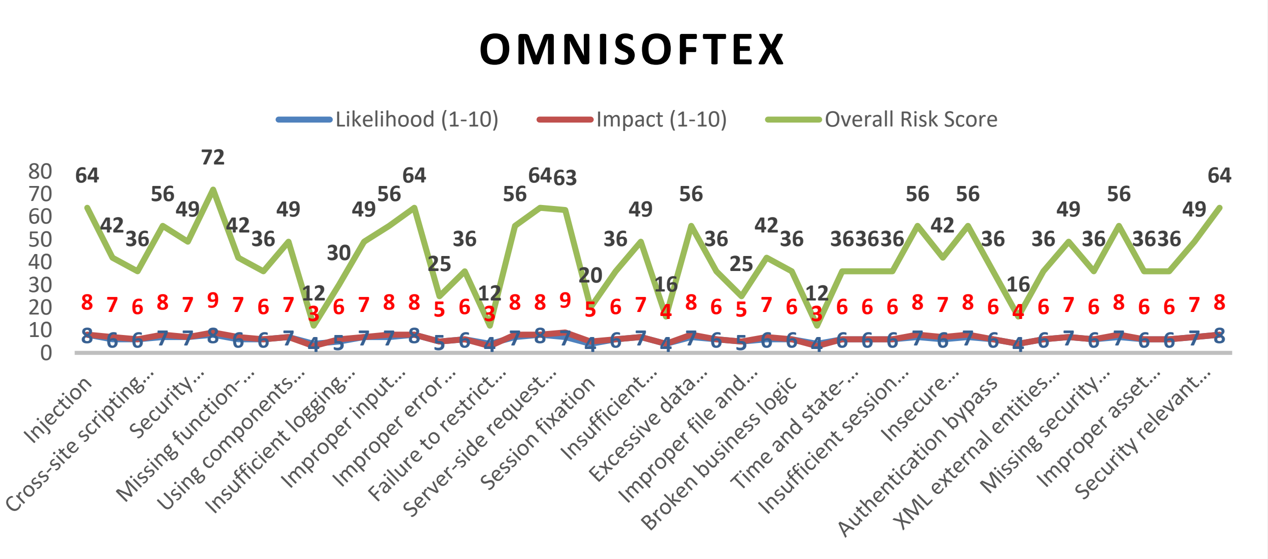
OmniSoftex assigns a risk a priority based on the protection poker technique's risk calculation, taking into account the possible effects on the application's confidentiality, integrity, and availability. High-risk locations receive prompt attention, and resources are allotted for mitigation in accordance with that assessment. The team keeps an eye on things and adjusts risk priorities on a regular basis to meet changing risks.

**Protection Techniques**:

OmniSoftex particularly used the protection poker technique to identify and evaluate the 46 security risks listed in the Security Assurance Model for the purposes of this questionnaire. They have had nothing but great experience with this strategy since it encourages team members to work together and come to consensus. They are able to apply focused mitigation techniques and efficiently prioritize threats because to the protection poker's organized methodology.

**Challenges and Solutions:**

OmniSoftex confronts difficulties in handling security threats, including the intricacy of incorporating security into the development lifecycle and striking a balance between project timeframes and security needs. They have made investments in threat intelligence platforms, automation technologies, and continuous training to solve these issues. Effective risk mitigation has also been made possible by building an organizational culture of security knowledge and accountability.



## Response from Centangle Interactive:

### Identifying Risks:

To identify and mitigate 46 security risks listed in the security assurance model, Centangle Interactive takes a proactive stance while developing applications. Using their knowledge of industry standards and security best practices, they carry out in-depth risk assessments and security audits. Furthermore, they work together with consumers to comprehend their security needs and worries.

### Risk Management:

The main goal of Centangle Interactive's risk management approach is to successfully reduce recognized risks by putting in place extensive security controls and procedures. To protect their apps, they employ a defense-in-depth strategy that combines technical restrictions, security guidelines, and user awareness training. In order to find and fix any flaws, they also regularly do security testing and vulnerability assessments

### Risk Assessment:

The application development process at Centangle Interactive incorporates risk assessment techniques such as the protection poker strategy. They can systematically assess and rank risks according to their impact and likelihood thanks to this technique. They guarantee a thorough grasp of potential hazards by incorporating stakeholders from several areas, such as development, business, and security.

**Risk Prioritization:**

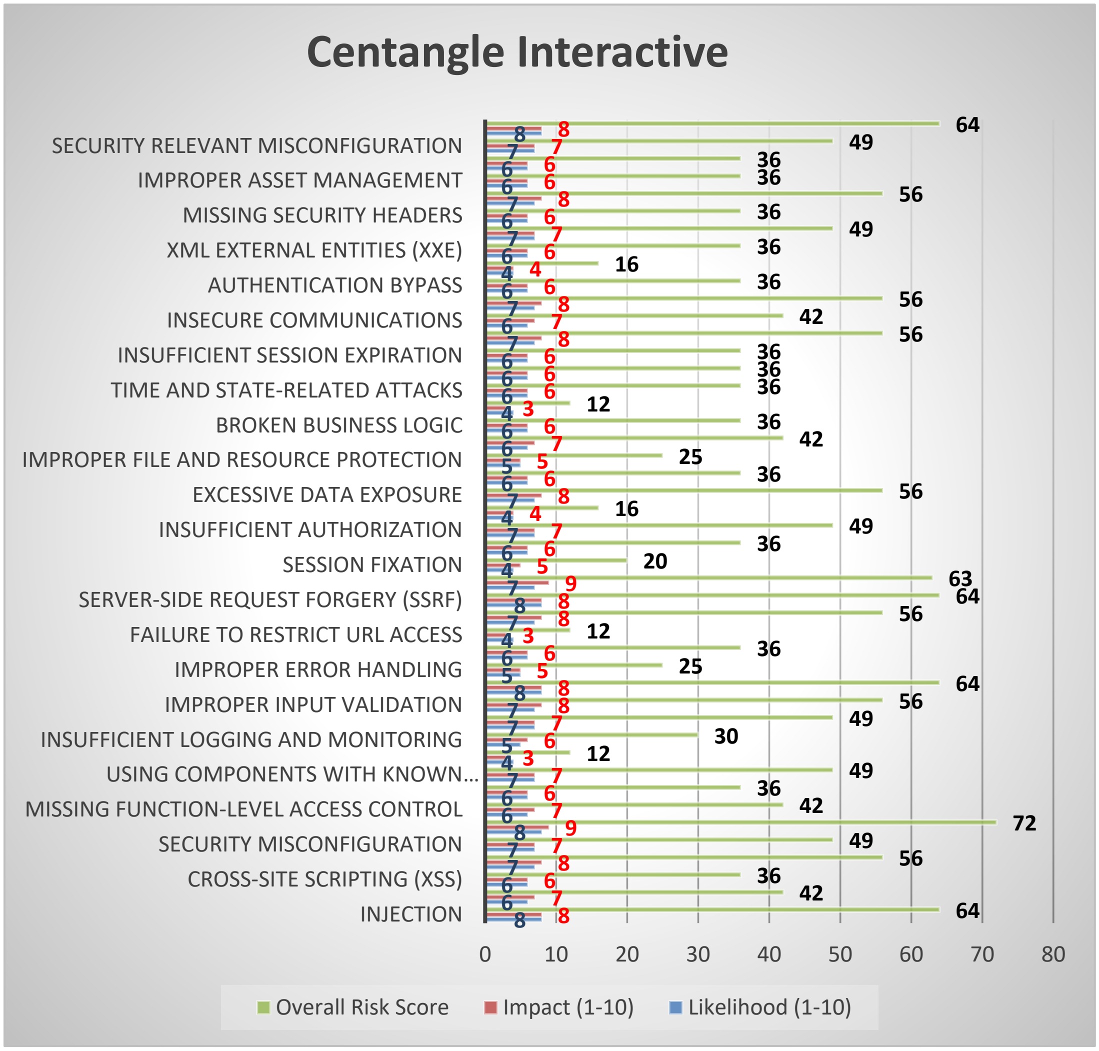
Centangle Interactive prioritizes risks based on the risk estimation offered by the protection poker approach. This prioritization considers the potential impact on the application's confidentiality, integrity, and availability. Immediate attention is given to high-risk areas, with resources allocated for mitigation in line with this approach. The team consistently updates and reviews risk priorities to stay responsive to evolving threat environments.

**Protection Techniques:**

Centangle Interactive employed the protection poker technique for the questionnaire to evaluate and tackle 46 security risks identified in the Security Assurance Model. Their positive experience with this method promotes teamwork, consensus-building, and well-informed decision-making. The systematic use of protection poker helps them efficiently recognize and prioritize risks, resulting in stronger security measures

**Challenges and Solutions:**

In dealing with security risks, Centangle Interactive faces resource limitations in terms of manpower, changing threat landscapes, and compliance demands. Moreover, instilling a culture of security awareness and responsibility within the organization has significantly reduced risk exposure.



## DISCUSSION AND ANALYSIS

The response from three different software companies AppInSnap, OmniSoftex, and Centangle Interactive was analyzed. The results demonstrate their use of protection poker, likelihood and impact-based risk prioritization, and approach to risk management. Variations in the likelihood, impact, and total risk scores ascribed to individual security risks are evident, based on the risk assessment input that was submitted. In order to substantiate the operation of a software firm, let us examine their risk assessment processes and results:

**AppInSnap:**

- Total Risk Score: 2306
- Number of Risks: 46
- Average Risk Score: 50.13

**OmniSoftex:**

- Total Risk Score: 1917
- Number of Risks: 46
- Average Risk Score: 41.67

**Centangle Interactive:**

- Total Risk Score: 2332
- Number of Risks: 46
- Average Risk Score: 50.69

**AppInSnap:** The majority of risk types have moderate total risk scores as a consequence of AppInSnap's risk assessment technique, which seems to concentrate on a balanced assessment of chance and impact. They give greater weight to risks with high likelihoods and substantial consequences, like injection, and sensitive data exposure, which raises overall risk scores. This method provides a rigorous risk assessment procedure in which potential threats to application security and integrity are carefully considered and analyzed.

**OmniSoftex:** OmniSoftex's risk assessment methodology seems to place more emphasis on the impact of risks rather than their likelihood. They assign relatively high impact scores to risks like Sensitive Data Exposure and Injection (same like AppInSnap) indicating such high impact risks needs to be given special consideration and effort to mitigate them during development. This approach indicates a focus on mitigating risks with significant potential impact, aligning with a strategy of prioritizing resources based on potential consequences rather than the probability of occurrence.

**Centangle Interactive:** The risk assessment process used by Centangle Interactive is notable for its thorough analysis of both likelihood and impact, which raises total risk scores for a variety of risk categories. They give high priority to risks with high likelihoods and major consequences, like injection, buffer overflow, and elevation of privilege, which raises overall risk scores. According to this method, threats should be carefully assessed and their ability to exploit flaws and jeopardize application security should be carefully considered.

## Conclusion:

The higher average risk score of Centangle Interactive indicates that its risk assessment technique is the most comprehensive among the three software businesses. Centangle Interactive exhibits a proactive approach to risk management by carefully assessing both possibility and impact, guaranteeing a thorough examination of potential threats to application security. Their approach places a strong emphasis on the importance of recognizing and ranking risks, making it easier to allocate resources according to their level of severity. Subsequent research endeavors could conduct a more thorough examination of the supplementary personnel or appointments necessary for efficiently mitigating and resolving security risks during the development process. This would yield significant insights into resource allocation tactics aimed at augmenting application security.

## References

[1]. H. Villamizar, A. Anderlin Neto, M. Kalinowski, A. Garcia and D. Méndez, "**An Approach for Reviewing Security-Related Aspects in Agile Requirements Specifications of Web Applications**," *IEEE 27th International Requirements Engineering Conference (RE)*, Jeju, Korea (South), pp. 86-97, 2019, doi: 10.1109/RE.2019.00020.

[2]. Martin Otieno, David Odera and Jairus Ekume Ounza **“Theory and practice in secure software development lifecycle: A comprehensive Survey”** , World Journal of Advanced Research and Reviews, 18(03), pp 53–78, 2023**.**

[3]. Wisdom Umeugo , Kimberly Lowrey and Shardul Y Pandya, “**Factors Affecting The Adoption Of Secure Software Practices In Small And Medium Enterprises That Build Software In-house**,” International Journal of Advanced Research in Computer Science, Volume 14(2), 2023.

[4]. 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, doi: 10.1109/ACCESS.2022.3178301.

[5] Mamdouh Alenezi and Sadiq Almuairfi “**Security Risks in the Software Development Lifecycle**” International Journal of Recent Technology and Engineering, Vol 8 (3), 2019, DOI:10.35940/ijrte.C5374.098319 .

[6]. W. Wang, Q. Zeng and A. P. Mathur, "**A Security Assurance Framework Combining Formal Verification and Security Functional Testing**," *12th International Conference on Quality Software*, Xi'an, China, pp. 136-139, doi: 10.1109/QSIC.2012.

[7] 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.

[8] 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 doi: 10.1109/DESEC.2018.8625114.

[9]. 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 doi: 10.1109/CyberSecPODS.2017.8074844.

[10] 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**," 2022 24th International Conference on Advanced Communication

Technology (ICACT), PyeongChang Kwangwoon_Do, Korea, Republic of, 2022, pp. 313-316, doi: 10.23919/ICACT53585.2022.9728868.

[11] Y. Zeng, Y. Cheng, G. Xie and R. Wang, "**Design of Mobile Application Lifecycle Security Management Platform**," *2021 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, Xi'an, China, 2021, pp. 26-30, doi: 10.1109/ICCNEA53019.2021.00017

[12] Al-Darwiash, A.I. and Choe, P. (2019), "**A framework of information security integrated with human factors", International Conference on Human-Computer Interaction (HCII)**, Springer, pp. 217-229. Arteaga, J.M., Gonzalez, R

[13] 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, doi: 10.13052/jwe1540-9589.2164.

[14] O. Kovalenko, O. Smirnov, A. Kovalenko and S. Kavun, "**Quantitative Risk Assessment Method Development in the Context of the SDLC-model**," *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, Kharkiv, Ukraine, 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143.

[15] Tøndel, I.A., Jaatun, M.G., Cruzes, D.S. and Williams, L. (2019), "**Collaborative security risk estimation in agile software development**", *Information and Computer Security*, Vol. 27 No. 4, pp. 508-535. https://doi.org/10.1108/ICS-12-2018-0138.