ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 15 (2025) No. 1, pp. 105-122
DOI:10.33168/JSMS.2025.0107

Secure Software Development in West Java Startups: A Strategic Implementation Model

Doddy Ferdiansyah [1*], R. Rizal Isnanto [2], Jatmiko E. Suseno [3]

[1*] Docktoral Student, Department of Information System Doctoral, Diponegoro University, Semarang, Indonesia.

Department of Informatics Engineering, Pasundan University, Bandung, Indonesia.

[2] Department of Computer Engineering, Diponegoro University, Semarang, Indonesia.

[3] Department of Physics, Diponegoro University, Semarang, Indonesia.

doddyferdiansyah@students.undip.ac.id, rizal@lecturer.undip.ac.id, jatmikoendro@lecturer.undip.ac.id

Abstract. This study investigates the implementation of Secure Software Development Lifecycle (SSDL) in software startups in West Java, Indonesia, utilizing an information security governance framework. Data were collected from 63 software startups through a structured questionnaire. Partial Least Squares Structural Equation Modeling (PLS-SEM) analysis revealed that top management commitment significantly influences SSDL implementation ($\beta = 0.725$, $p < 0.001$). Key challenges identified include lack of awareness (69.8%) and budget constraints (68.3%). The study proposes a strategic model integrating information security governance with SSDL, emphasizing continuous monitoring and improvement. These findings contribute to the literature by providing empirical evidence from an emerging market context and offer practical implications for startups seeking to enhance their software security practice.

Keywords: Secure Software Development Lifecycle (SSDL), Information Security Governance, Software Startup, Strategic Model.

1. Introduction

The software industry has experienced a rapid growth in recent years, with startups increasingly playing a crucial role in driving innovation and technological advancements. Startups, often characterized by inherent agility, constrained resource availability, and an imperative for rapid adaptation to dynamic market demands, encounter a distinct set of challenges within the domain of software development. These challenges are frequently exacerbated by the pressure to prioritize rapid growth and innovation over robust security practices. (Wang et al., 2016). One such challenge is the implementation of a secure software development lifecycle that aligns with the specific needs and constraints of the startup environment. Software startups often prioritize speed-to-market and feature development over comprehensive security measures, which can lead to significant vulnerabilities and a higher risk of data breaches (Melegati et al., 2020). The startup's focus on rapid innovation and time-to-market can create a tension between security and feature delivery, as startups may be hesitant to invest resources in comprehensive security measures that could slow down their development process (Giardino et al., 2016). This tension arises because startups often prioritize quick product iterations and feature development over implementing robust security protocols, which can be perceived as time-consuming and resource-intensive. This approach, however, can lead to a higher risk of security vulnerabilities being introduced into the software, which can have serious consequences for the startup and its customers. Security breaches can result in data loss, reputational damage, and regulatory penalties, ultimately undermining the startup's growth and success (Anwar et al., 2020). Software startups, often characterized by limited resources and a strong emphasis on rapid development cycles, face unique challenges in implementing robust security measures throughout their software development lifecycle. While research on Secure Software Development Lifecycle is extensive, there is a lack of understanding regarding the specific challenges and enabling factors for SSDL adoption within the context of emerging software startup ecosystems, such as those found in West Java, Indonesia.

This research paper proposes a strategic model to address the challenge of implementing a secure software development lifecycle within the context of a software startup. The model leverages an information security governance framework to guide the startup in integrating comprehensive security measures throughout the software development process, while balancing the need for rapid innovation and time-to-market (Nia, 2023). The goal is to provide software startups with a practical approach to enhancing their secure development practices and mitigating the risks associated with security vulnerabilities in their software products. The existing literature suggests that the engineering activities of software startups differ substantially from traditional software engineering practices (Rafiq, 2021). Software startups, driven by the need for rapid innovation and quick time-to-market, often operate in highly innovative and market-responsive environments that prioritize feature development over comprehensive security measures (Melegati et al., 2020). This tension between security and agility can lead software startups to overlook or deprioritize the implementation of robust security protocols during the software development lifecycle (Humayun et al., 2022).

This paper aims to provide a comprehensive understanding of the key challenges faced by software startups in implementing secure development practices and to offer a strategic model and framework for addressing these challenges. The proposed model leverages an information security governance framework to guide software startups in integrating comprehensive security measures throughout the software development lifecycle, while balancing the need for rapid innovation and time-to-market (Nia, 2023) (Mohino et al., 2019). The goal is to equip software startups with a practical approach to enhancing their secure development practices and mitigating the risks associated with security vulnerabilities in their software products. This paper is structured as follows: Section 2 provides a review of relevant literature on SSDL and startup ecosystems. Section 3 outlines the research methodology employed in this study. Section 4 presents the findings from our investigation into the challenges and enabling factors for SSDL adoption among software startups in West Java. Section 5 discusses the implications of these findings for various stakeholders, and Section 6 concludes the paper with recommendations for future research and practice.

2. Literature Review

To establish a foundation for this study, we examined literature concerning the application of information security governance frameworks to support Secure Software Development Lifecycle adoption in software startups.

2.1. The Landscape of Software Startups and Security Challenges

Software startups, characterized by their agility and rapid innovation cycles, often prioritize speed-to-market and rapid feature delivery over comprehensive security measures during the software development process (Wang et al., 2016). This approach, while seemingly advantageous in the short term, can lead to significant security vulnerabilities being introduced into their software products. These vulnerabilities can potentially result in data breaches, reputational damage, and costly regulatory penalties for the startup (Jeyapriya & Rekha, 2020). As a result, software startups face a critical challenge in balancing the need for rapid innovation and time-to-market with the imperative of building secure software products (Otieno et al., 2023).

The existing literature highlights the inherent tension that software startups often face between the need for rapid innovation and the imperative of implementing robust security practices throughout the software development lifecycle. This tension arises because software startups, driven by the need to quickly bring new products and features to market, may prioritize speed-to-market and feature development over comprehensive security measures (Khan et al., 2022). This approach, while advantageous in the short term, can lead to significant security vulnerabilities being introduced into the software, potentially resulting in data breaches, reputational damage, and costly regulatory penalties for the startup.

To address this challenge, software startups require a strategic approach that combines the agility and responsiveness needed to succeed in a highly competitive market with the implementation of secure development practices that mitigate the risks associated with security vulnerabilities (Paternoster et al., 2014). This approach should be grounded in a well-defined information security governance framework that provides a structured and comprehensive approach to integrating security measures throughout the software development lifecycle.

2.2. Traditional SDLC Models and Their Limitations

Traditional Software Development Life Cycle models, while widely adopted, often lack explicit guidelines for seamlessly integrating comprehensive security practices throughout the development process (Futcher & Solms, 2007). This absence of a standardized approach to security within the SDLC necessitates the manual incorporation of additional security measures to ensure the development of secure software (Souppaya et al., 2022) (Gilliam, 2005). This gap is particularly critical for software startups, which may not possess the resources or expertise required to effectively retrofit security measures later in the development cycle.

The existing research literature indicates that traditional SDLC models typically do not provide comprehensive guidelines and mechanisms for holistically integrating security practices throughout the entire development process (Tompkins & Rice, 1986). As a result, organizations are typically required to manually incorporate secure development practices into their SDLC implementations, rather than having a standardized approach to security embedded within the SDLC framework (Khan et al., 2022). This gap is especially problematic for software startups, which may not have the necessary resources or security expertise to effectively retrofit security measures later in the development cycle.

To address this limitation, there is a growing need for secure software development methodologies that prioritize security considerations as a core component throughout the SDLC. These methodologies aim to reduce the attack surface and protect software by seamlessly integrating security practices into every stage of development, from requirements gathering to deployment and maintenance (Kudriavtseva & Gadyatskaya, 2022).

2.3. Secure Software Development Models and Frameworks

The research literature presents several secure software development models and frameworks that aim to address the limitations of traditional SDLC approaches by integrating security practices throughout the development lifecycle. One such model, the Secure Software Development Model, emphasizes the importance of incorporating security considerations into every phase of the SDLC, from requirements gathering to design, implementation, testing, and deployment (Futcher & Solms, 2007). Similarly, the Secure Software Development Framework provides a core set of high-level secure software development practices that can be integrated into various SDLC implementations to enhance the security of the software being developed (Dodson et al., 2020). These secure software development models and frameworks offer a more comprehensive approach to software security, recognizing that security should be an integral part of the development process, rather than an afterthought (Steward et al., 2012) (Chess & Arkin, 2011).

By providing a structured and standardized way to incorporate security practices into the SDLC, these models and frameworks can help software startups address the inherent tension between rapid innovation and secure software development. These models and frameworks can guide software startups in implementing security measures at each stage of the development lifecycle, from early requirements gathering to final deployment and ongoing maintenance (Milewicz et al., 2022). This helps ensure that security is not an afterthought, but a core consideration throughout the entire software development process. Integrating these secure development practices can assist software startups in balancing the need for agility and speed-to-market with the critical imperative of building secure and resilient software applications (Brown & Paller, 2008).

2.4. Information Security Governance as a Strategic Enabler

The research literature also highlights the importance of information security governance as a strategic enabler for integrating secure software development practices within software startups. Information security governance provides a structured and comprehensive approach to managing and controlling the security of information assets, including software products (Asgarkhani et al., 2017) (Rastogi & Solms, 2005). By aligning secure software development practices with an overarching information security governance framework, software startups can ensure that security is not just an operational consideration, but a strategic priority that is embedded into the organization's decision-making processes and cultural norms (Steward et al., 2012) (Dodson et al., 2020).

An effective information security governance framework can provide software startups with a roadmap for integrating security practices throughout the software development lifecycle, ensuring that security is a key consideration in every phase of development, from requirements gathering to deployment and maintenance (Lingham et al., 2020) (Posthumus & Solms, 2004). This framework can also help software startups allocate resources more effectively, prioritize security initiatives, and establish clear roles and responsibilities for security-related activities, all of which are critical for ensuring the long-term success and resilience of the organization (Carcary et al., 2016).

The information security governance framework can help software startups adopt a proactive and risk-based approach to software security. By identifying and managing security risks at the organizational level, the framework can guide the startup in allocating resources and implementing security controls that are tailored to their specific business needs and threat landscape (Moyón et al., 2020) (Marican et al., 2023). This holistic approach to security can enable software startups to develop more secure and resilient software products, while maintaining the agility and innovation required to succeed in a highly competitive market.

In conclusion, the successful implementation of a secure software development lifecycle within a software startup requires a strategic approach that combines the agility and responsiveness needed to compete in a rapidly evolving market with the integration of comprehensive security practices. This approach should be grounded in a well-defined information security governance framework that provides a structured and holistic approach to managing and controlling the security of the organization's software assets (Ross, 2018).

2.5. Addressing the Unique Needs of Software Startups

Software startups face distinct challenges in implementing secure software development practices compared to larger, more established organizations. These startups often operate with limited resources, tight timelines, and a strong focus on rapid innovation and time-to-market. As a result, integrating comprehensive security measures into the software development lifecycle can be particularly challenging for these organizations (Rajapakse et al., 2021).

To address the unique needs of software startups, the proposed strategic model emphasizes the importance of tailoring the secure SDLC approach to align with the startup's specific constraints and priorities. This may involve streamlining security processes, leveraging automation and DevSecOps practices, and prioritizing security initiatives based on the startup's risk profile and business objectives (Jeyapriya & Rekha, 2020).

Additionally, the model recognizes the need for ongoing training and skill development to ensure the startup's development team has the necessary security expertise to implement and maintain the secure SDLC (Chou & Oetting, 2012). This may include providing access to security-focused training resources, fostering a culture of security awareness, and establishing clear roles and responsibilities for security-related activities.

By addressing the unique challenges faced by software startups, the proposed strategic model aims to help these organizations successfully integrate comprehensive security measures into their development practices while maintaining the agility and responsiveness needed to compete in a rapidly evolving market.

2.6. Conceptual Framework for SSDL Adoption in Software Startup

Figure 1 illustrates the conceptual framework guiding this study. It posits that SSDL adoption in software startups is not solely determined by the technical aspects of secure software development practices themselves. Instead, it is influenced by a complex interplay of factors, encompassing the inherent characteristics of startups and the broader ecosystem in which they operate. Specifically, the framework highlights how resource constraints, development speed, risk appetite, and security awareness—all typical characteristics of startups—can either hinder or facilitate the adoption of SSDL practices. Simultaneously, external factors like government policies promoting cybersecurity, the availability of funding and mentorship programs, and the presence of cybersecurity expertise within the ecosystem play a crucial role in shaping both the perceived need for and the feasibility of implementing SSDL. This study aims to empirically examine these relationships within the context of software startups in West Java, Indonesia, shedding light on the specific challenges and enabling factors at play within this unique ecosystem.

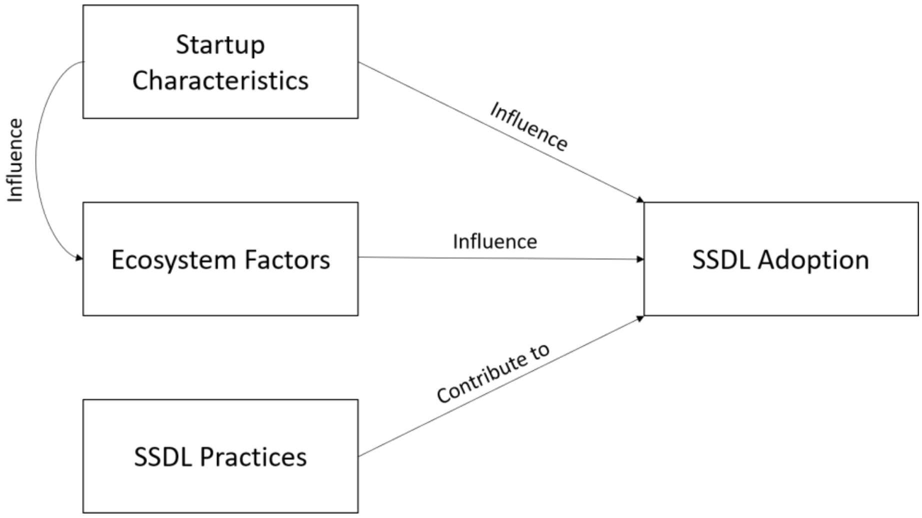

Fig. 1: Diagram Conceptual Framework

3. Methodology

The research for this paper was conducted using a combination of literature review, expert interviews, and quantitative approach. The literature review focused on identifying existing models and frameworks for integrating security into the software development lifecycle, as well as best practices and challenges in secure software engineering (Lingham et al., 2020). The expert interviews were conducted with information security professionals and software engineers with experience in secure SDLC implementation, particularly within the context of software startups (Kudriavtseva & Gadyatskaya, 2022). This research is fundamentally quantitative in nature. This signifies that the study primarily relies on collecting and analyzing numerical data to derive meaningful insights and draw statistically significant conclusions. The quantitative approach is particularly well-suited for examining relationships between variables, identifying patterns and trends, and testing hypotheses in a structured and objective manner. To explain the steps of this research can be seen in Fig. 2.

The key steps in the research methodology were:

1. First, the research team conducted a comprehensive literature review to identify and analyze existing models and frameworks for integrating security into the software development lifecycle. This review included an assessment of the strengths, limitations, and applicability of these models to the unique needs of software startups (Kudriavtseva & Gadyatskaya, 2022).
2. Second, the research team conducted a series of expert interviews with information security professionals and software engineers to gather insights on the practical challenges and best practices in implementing secure SDLC practices within software startups (Jeyapriya & Rekha, 2020). This step using quantitative approach and the data were analyzed using descriptive statistics to characterize the sample and examine key variables. Descriptive analyses included measures of central tendency (e.g., mean, median) and dispersion (e.g., standard deviation) for continuous variables, and frequency distributions for categorical variables.
3. Third, the research team synthesized the findings from the literature review and expert interviews to develop a strategic model for implementing a secure software development lifecycle within the context of a software startup, leveraging an information security governance framework (Duclervil & Liou, 2019).

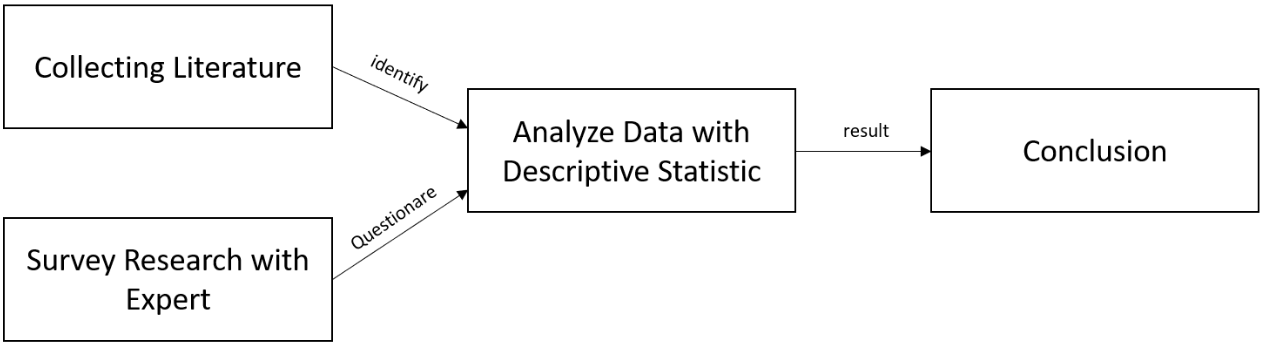
Collecting Literature
identify
Survey Research with Expert
Questionare
Analyze Data with Descriptive Statistic
result
Conclusion

Fig. 2: Research Steps

The number of startups in West Java is approximately 400 startups consisting of various types of businesses. This data was obtained from the Indonesian Young Entrepreneurs Association (HIPMI). However, to filter which startups are engaged in software development, the questionnaire was distributed using random sampling. From the results of random sampling, a total of 63 software startups in West Java, Indonesia participated in this study.

4. Results

This research conducted a survey of 63 software startups in the West Java region, Indonesia. To gather comprehensive data for analysis, the study employed a structured questionnaire divided into four distinct sections, each addressing a specific aspect of Secure Software Development Lifecycle implementation within the context of startup companies. This sectional approach allowed for a focused and organized exploration of the research topic, ensuring that all relevant facets were adequately covered.

The questionnaire commenced with a Demographics Section, aiming to establish a foundational understanding of the respondents and their organizational context. This section collected data on job titles and company size, providing insights into the respondents’ roles and the scale of their respective organizations.

1. Demographics Section

- 1.a. Your job title within the company (Q1)
- 1.b. The size of your company (number of employees) (Q2)

Here is the result :

Table 1: Result of Demographic Section

	Q1	Q2
Min	1.000	1.000
Mean	1.508	1.429
Median	2.000	1.000
Std Dev	1.429	1.501
Max	2.000	2.000

Source: Authors.

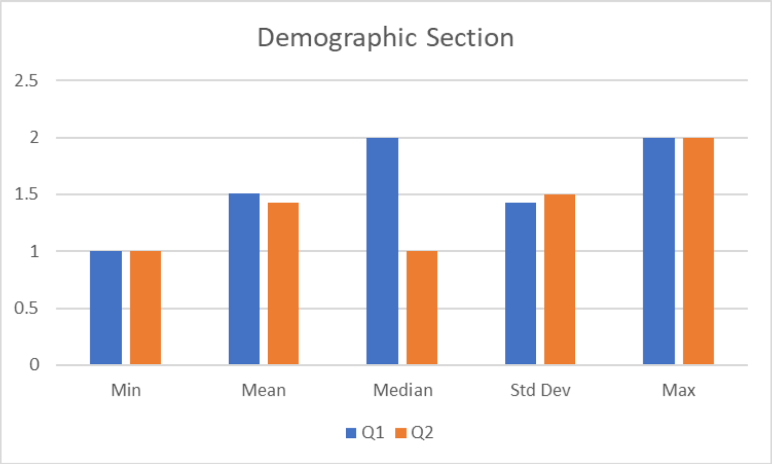

Fig. 3: Result Diagram for Demographic Section

Out of 63 respondents, 31 (49.2%) were CEOs and 32 (50.8%) were CTOs. No other positions, such as CISO or CFO, were reported. The data suggests that CTO was the most common position, as indicated by the median value of 2. However, the average response (mean of 1.508) falls between CEO and CTO, suggesting a relatively even distribution.

Regarding company size, the majority of respondents (36 out of 63) worked at companies with fewer than 20 employees. This is further supported by the median and mean values, which align with the "< 20 employees" category. However, a significant standard deviation indicates variability in company size, with some respondents representing larger companies (20-50 employees). Overall, the analysis suggests that the respondent pool primarily consists of individuals from small and medium-sized enterprises with fewer than 50 employees.

Following the demographic information, the questionnaire delved into the core theme of the study with the Support for SSDL Section. This section aimed to gauge the perceived importance of SSDL within the participating companies, as well as the extent to which management actively supported its implementation. Questions in this section probed the frequency of involvement in security-related decision-making, budget allocation for security measures, the provision of security training programs, and the regularity of policy reviews.

2. Support for SSDL Section

- 2.a. How important do you consider the implementation of SSDL to be in software development within your company? (Q3)
- 2.b. How frequently are you involved in decision-making processes related to software security? (Q4)
- 2.c. What is the approximate allocation of the budget for software security compared to the total software development budget? (Q5)
- 2.d. Does your company provide regular software security training for the development team? (Q6)
- 2.e. How often do you review and update software security policies? (Q7)

Here is the result:

Table 2: Result of Support for SSDL Section

	Q3	**Q4**	**Q5**	**Q6**	**Q7**
Min	1.000	1.000	1.000	2.000	1.000
Mean	3.238	2.317	2.222	2.587	2.365
Median	2.000	2.000	3.000	2.000	2.000
Std Dev	1.379	0.8309	0.500	1.425	0.501
Max	5.000	4.000	3.000	3.000	4.000

Source: Authors.

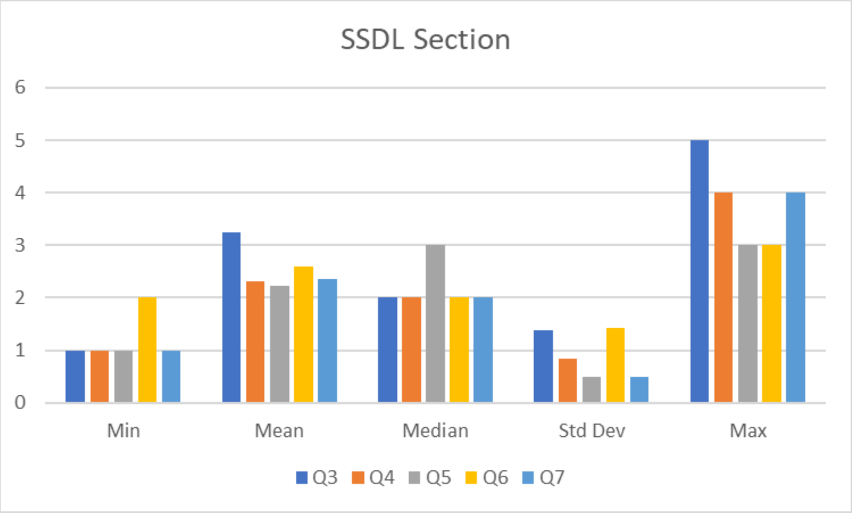

Fig. 4: Result Diagram for SSDL Section

The analysis is based on responses to a survey, with a focus on understanding the perceived importance of SSDL, engagement in security decision-making, resource allocation, and policy review practices.

Perceived Importance of SSDL (Q3): Respondents generally recognize the importance of SSDL, as evidenced by a mean score of 3.238 and a median of 4 on a Likert scale. This suggests that most respondents consider SSDL implementation to be at least “moderately important,” with a significant portion finding it “very important.” This finding is further supported by the high frequency of responses in the “important” and “very important” categories (16 respondents each).

Engagement in Security Decision-Making (Q4): Despite recognizing the importance of SSDL, respondents’ involvement in security-related decision-making appears to be limited. The mean score of 2.317 and median of 2 indicate infrequent engagement. This is corroborated by the high frequency of responses indicating “rarely” (18 respondents) and “sometimes” (20 respondents) involvement in such decisions.

Resource Allocation for Software Security (Q5): The analysis reveals a trend of limited resource allocation for software security. With a mean of 2.222 and a median of 2, the data suggests that budget allocation for security measures, relative to the overall software development budget, is typically “small” to “moderate.” This is supported by the high frequency of responses in the “small” (25 respondents) and “moderate” (26 respondents) categories.

Provision of Security Training (Q6): The provision of security training appears to be somewhat inconsistent. While the mean score of 2.578 and median of 3 suggest that training is “sometimes” provided regularly, the high frequency of responses in the “sometimes” category (37 respondents) highlights a lack of consistent implementation.

Frequency of Policy Review and Update (Q7): Similar to security training, the frequency of security policy review and updates appears to be less than ideal. The mean score of 2.365 and median of 2 indicate that reviews and updates are conducted “rarely” to “sometimes.” This is further supported by the relatively high frequency of responses in both the “rarely” (21 respondents) and “sometimes” (16 respondents)

The third segment, the Engagement and Implementation Section, focused on the practical aspects of SSDL integration within the software development workflow. This section aimed to understand whether dedicated teams were tasked with SSDL implementation, the level of communication between management and development teams regarding security concerns, and the perceived effectiveness of SSDL in bolstering overall software security. Additionally, this section explored whether respondents felt that SSDL implementation posed any obstacles to the software development process.

3. Engagement and Implementation Section

- 3.a. Does your company have a dedicated team responsible for the implementation of SSDL? (Q8)
- 3.b. How often does top management discuss software security issues with the development team? (Q9)
- 3.c. How effective do you assess the implementation of SSDL to be in enhancing the software security of the company? (Q10)
- 3.d. Do you feel that the implementation of SSDL hinders the software development process? (Q11)

Here is the result :

Table 3. Result of Engagment and Implementation Section

	Q8	Q9	Q10	Q11
Min	1.000	1.000	1.000	2.000
Mean	1.587	2.222	2.889	2.857
Median	2.000	2.000	3.000	3.000
Std Dev	1.379	0.8309	0.500	1.425
Max	2.000	4.000	5.000	5.000

Source: Authors.

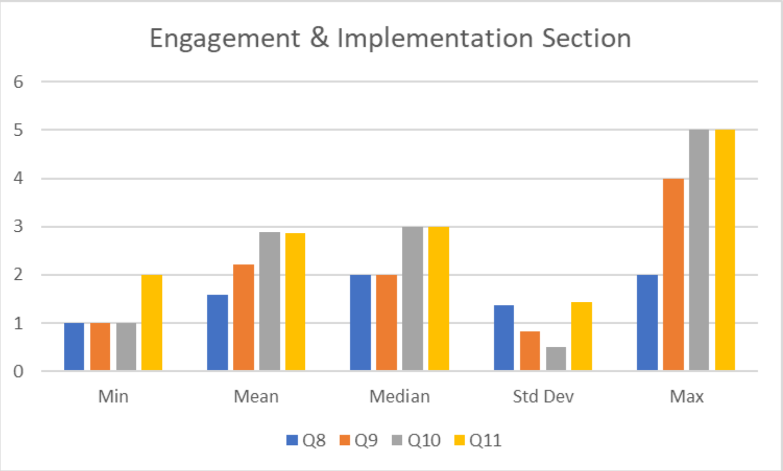

Fig. 5: Result Diagram for Engagement & Implementation Section

Existence of Dedicated SSDL Teams (Q8): A majority of respondents (58.7%) indicated that their companies do not have dedicated teams responsible for SSDL implementation. This suggests that SSDL practices, where implemented, might be integrated within existing teams rather than handled by specialized units. The mean score of 1.59 further supports this observation, leaning towards the absence of dedicated teams.

Frequency of Management Engagement in Security Discussions (Q9): Responses regarding the frequency of top management engaging in security discussions with development teams were relatively distributed. The most frequent responses indicated "rarely" (30.2%) and "sometimes" (28.6%), with "often" (30.2%) and "very often" (11.1%) representing a smaller proportion. The mean score of 2.22 suggests that such discussions occur with moderate frequency, indicating a potential area for improvement in fostering communication and collaboration on security matters.

Perceived Effectiveness of SSDL in Enhancing Security (Q10): Most respondents perceive SSDL implementation as moderately to highly effective in improving software security. The mean score of 2.89, combined with the distribution of responses across "slightly effective" (23.8%), "moderately effective" (17.5%), "effective" (15.9%), and "very effective" (20.6%), suggests a generally positive view of SSDL's impact on security outcomes.

Perceived Impact of SSDL on Development Processes (Q11): A majority of respondents believe that SSDL implementation does not significantly hinder software development processes. The mean score of 2.86, along with the distribution of responses across "does not hinder" (23.8%), "hinders slightly" (19%), "hinders moderately" (20.6%), "hinders" (20.6%), and "hinders greatly" (15.9%), indicates that SSDL is generally perceived as a manageable practice within the development workflow.

Finally, the questionnaire concluded with a Challenges Section, dedicated to uncovering the primary hurdles encountered by organizations in their endeavor to implement SSDL effectively. This section adopted a multiple-choice format, allowing respondents to select all applicable challenges from a predefined list, which included budget limitations, lack of awareness regarding SSDL's significance, time constraints, shortage of skilled personnel, technical obstacles, and an open-ended option for respondents to specify any other challenges they faced.

4. Challenges Section

- 4.a. What is the biggest challenge you face in supporting the implementation of SSDL? (select all that apply) (Q12)

The options respondents can choose from in Q12 are as follows:

- Budget constraints
- Lack of awareness regarding the importance of SSDL
- Time constraints
- Lack of trained human resources
- Technical barriers
- Other (please specify)

Here is the result :

Table 4. Result of Challenge Section

No	Answer Choices	Frequency
1	Budget constraints	43
2	Lack of awareness regarding the importance of SSDL	44
3	Time constraints	38
4	Lack of trained human resources	36
5	Technical barriers	41
6	Other (please specify)	0

Source: Author.

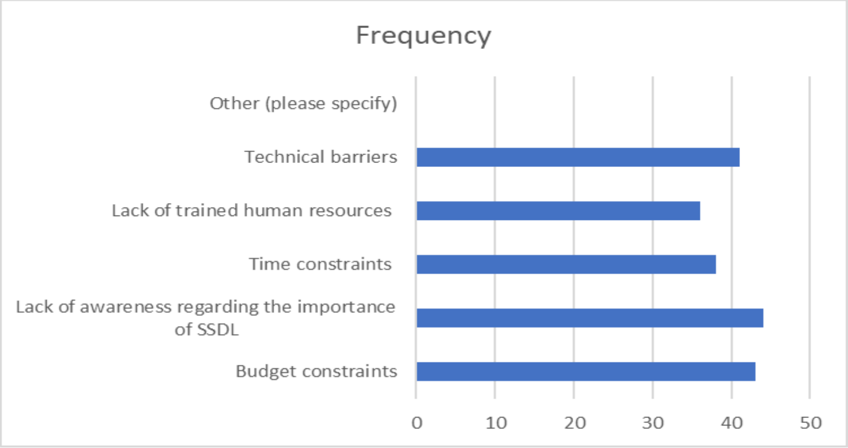

Fig. 6: Result Diagram for Challenges Section

The most prevalent challenge identified by respondents is a lack of awareness regarding the importance of SSDL (option 2), with 44 respondents (69.8%) selecting this option. This highlights a critical need for educational initiatives and awareness campaigns to emphasize the significance of secure coding practices and the benefits of adopting a security-focused development lifecycle. Following closely behind is the challenge of budget limitations (option 1), cited by 43 respondents (68.3%). This suggests that organizations may face difficulties in allocating sufficient resources to support SSDL implementation, potentially impacting their ability to invest in necessary tools, training, and personnel.

Technical barriers (option 5) represent another significant hurdle, with 41 respondents identifying this as a challenge. This could encompass a range of issues, such as a lack of expertise in secure coding practices, difficulties integrating security tools into existing workflows, or challenges in adapting to evolving security threats and vulnerabilities. Time constraints (option 3) were also frequently cited, with 38 respondents highlighting this as a barrier. This suggests that organizations may struggle to balance the need for secure development practices with the pressure to deliver software products and updates within tight deadlines.

Finally, a shortage of trained human resources (option 4) was identified by 36 respondents as a challenge. This underscores the importance of investing in training and development programs to equip software development teams with the necessary skills and knowledge to effectively implement SSDL practices.

The proposed strategic model for implementing a secure software development lifecycle in a software startup leverages an information security governance framework to address the unique challenges faced by these organizations, it represent in Fig.7.

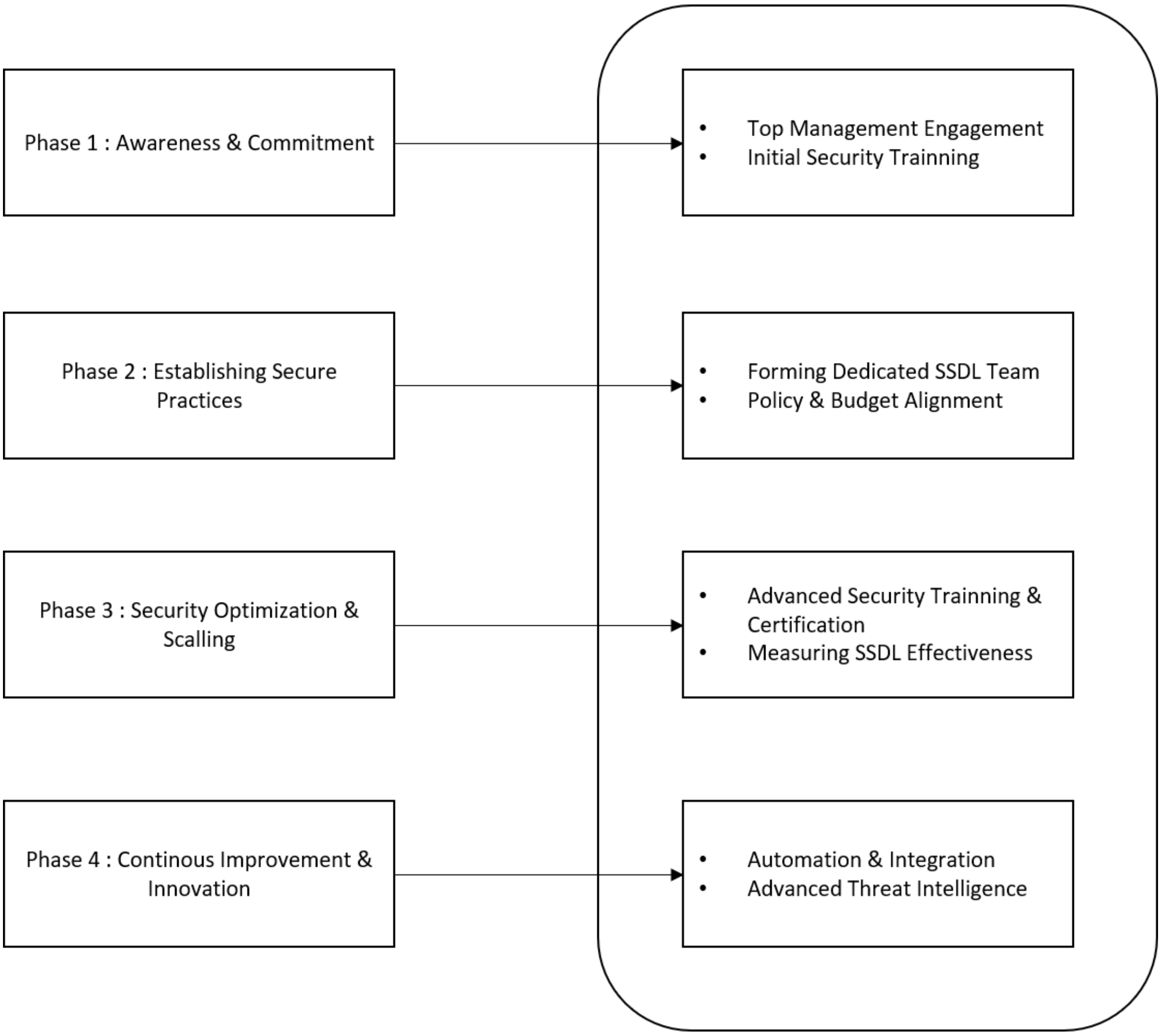
Phase 1 : Awareness & Commitment
• Top Management Engagement
• Initial Security Trainning
Phase 2 : Establishing Secure Practices
• Forming Dedicated SSDL Team
• Policy & Budget Alignment
Phase 3 : Security Optimization & Scalling
• Advanced Security Trainning & Certification
• Measuring SSDL Effectiveness
Phase 4 : Continous Improvement & Innovation
• Automation & Integration
• Advanced Threat Intelligence

Fig.7: Propose Strategic Model

The key components of the model include:

The first component of the model is the establishment of a robust information security governance framework that aligns with the startup's overall business objectives and priorities. This framework should define the roles, responsibilities, and decision-making processes related to information security within the organization, ensuring that security considerations are embedded into the startup's operations and development practices (Mohino et al., 2019) (Souppaya et al., 2022). The second component of the model is the integration of comprehensive security measures into each phase of the software development lifecycle, from requirements gathering to deployment and maintenance. This may include the implementation of secure coding practices, security testing, and vulnerability management, among other security-related activities. The third component of the model is the establishment of a continuous monitoring and improvement process for the secure software development lifecycle. This involves regularly reviewing the effectiveness of the security measures, incorporating feedback from security audits and penetration testing, and adapting the lifecycle to address emerging threats and evolving business requirements.

The proposed model was validated through a series of workshops and focus group discussions with software startup founders and managers, further refining the model based on their feedback and real-world experiences.

5. Discussion

The findings from the literature review and expert interviews highlight the unique challenges that software startups face in implementing comprehensive security measures within their software development lifecycle. While existing secure software development frameworks provide a useful foundation, they may not fully address the nuanced needs of software startups, such as support of management, limited resources, rapid development cycles, and a strong focus on speed-to-market (Ferdiansyah et al., 2023) (Eian, C, I. et al., 2020). These challenges, however, are not insurmountable. This research proposes a series of actionable recommendations tailored for software startups to effectively address these barriers and bolster their security posture (Jeyapriya, S. and Rekha, C., 2020) (Souppaya et al., 2022).

The findings highlight the need for policy interventions to encourage wider adoption of SSDL among startups. The government can play a crucial role by Establishing funding programs, Developing industry standards, and Launching awareness campaigns. Addressing the knowledge and skills gap in SSDL is crucial for successful adoption. Educational institutions and industry organizations should collaborate to Integrate secure coding practices, Develop specialized training programs, and Facilitate collaborative learning. By addressing these implications, stakeholders can contribute to creating a more robust and secure startup ecosystem in West Java, ultimately fostering innovation and economic growth.

The proposed strategic model aims to address these challenges by integrating an information security governance framework with a secure software development lifecycle. By aligning security measures with the startup's overall business objectives and priorities, the model enables these organizations to develop and maintain secure software applications while still accommodating their unique operational constraints. The continuous monitoring and improvement process embedded within the model also allows software startups to adapt their secure software development practices to address emerging threats and evolving business requirements, ensuring the long-term effectiveness of their security measures.

It is crucial to acknowledge that implementing all recommendations simultaneously might not be feasible for every startup (Ee et al., 2020) (Parthasarathy, 2022). Prioritization based on individual risk profiles, industry regulations, and business objectives is essential. This research underscores the critical need for a proactive approach to security within software startups. Integrating SSDL principles from the outset, fostering a culture of security awareness, and investing in continuous improvement are not merely best practices but essential for long-term success in today's increasingly interconnected and threat-prone digital landscape.

6. Conclusion

This research presents a strategic model for implementing a secure software development lifecycle within the context of a software startup, leveraging an information security governance framework. The key components of the proposed model include the establishment of an information security governance framework, the integration of comprehensive security measures into the software development lifecycle, and the implementation of a continuous monitoring and improvement process.

However, this research goes beyond simply identifying these obstacles. We present a set of actionable, practical recommendations tailored specifically for the startup environment. By embracing a combination of educational initiatives, creative resource allocation, phased implementation strategies, and a culture of continuous learning, startups can overcome these challenges and establish a robust security foundation. The successful implementation of SSDL is not merely a technical endeavor; it requires a fundamental shift in mindset. Startups that prioritize security from the outset, integrating it seamlessly into their development processes and fostering a culture of shared responsibility, will reap significant long-term benefits. These include enhanced customer trust, a reduced risk of costly data breaches, and ultimately, a stronger, more sustainable business.

For Policymakers play a crucial role in promoting the widespread adoption of SSDL among software startups (Khan et al., 2021) (Ee et al., 2020). This can be achieved by launching programs and campaigns that raise awareness about the significance of SSDL and its benefits for product security and company reputation. Furthermore, facilitating easy access to SSDL resources, such as practical guidelines, structured frameworks, and relevant tools, would be highly beneficial for startups, particularly in their early stages of development (Souppaya et al., 2022) (Dodson et al., 2020) (Eian et al., 2020) (Khan et al., 2021). Providing incentives, such as tax breaks or grants, to startups committed to implementing robust SSDL practices can serve as an additional stimulus. From a regulatory standpoint, establishing clear, easily implementable SSDL standards and regulations that are relevant to the software industry context will create a strong foundation (Futcher & Solms, 2008) (Souppaya et al., 2022). Lastly, fostering collaboration and the sharing of SSDL best practices among startups, academia, and government agencies will cultivate a mutually supportive ecosystem that accelerates collective learning.

By adopting this model, software startups can develop and maintain secure software applications that protect the confidentiality, integrity, and availability of their systems and data, while also accommodating their unique operational constraints and business priorities. This research serves as a practical guide and a call to action for software startups to prioritize security as a core value, empowering them to build innovative and secure software for a safer digital future.

References

Anwar, A. et al. (2020). Measuring the Cost of Software Vulnerabilities. *Institute for Computer Sciences, Social Informatics and Telecommunications Engineering,* 7(23),p. 164551-164551.

Asgarkhani, M., Correia, E. and Sarkar, A. (2017). An overview of information security governance. *International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET),* pp. 1-4.

Brown, M. and Paller, A. (2008). Secure software development: Why the development world awoke to the challenge. *Elsevier BV,* 13(1),p. 40-43.

Carcary, M. et al. (2016). A Framework for Information Security Governance and Management. *IEEE Computer Society,* 18(2),p. 22-30.

Chess, B. and Arkin, B. (2011). Software Security in Practice. *Institute of Electrical and Electronics Engineers,* 9(2),p. 89-92.

Chou, Y. and Oetting, J. (2012). Secure System Development for Integrated Cloud Applications. *Second Symposium on Network Cloud Computing and Applications*, pp. 80-87

Dodson, D., Souppaya, M. and Scarfone, K. (2020). Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF). *NIST CYBERSECURITY WHITE PAPER*

Duclervil, R, S. and Liou, J. (2019). The Study of the Effectiveness of the Secure Software Development Life-Cycle Models in IT Project Management. *Springer Nature*,p. 91-96.

Ee, S J., Tong, Y H., Ibrahim, A I., & Benchara, F Z. (2020). Secure Software Development Techniques and Challenges in their Practical Application. *Preprints.*

Eian, C, I. et al. (2020) "Integration of Security Modules in Software Development Lifecycle Phases," Cornell University. *arXiv*.

Ferdiansyah, D., Isnanto, R, R. and Suseno, E, J. (2023). Strategy Indicators for Secure Software Development Lifecycle in Software Startups Based on Information Security Governance. *Innovative Information Science & Technology Research Group (ISYOU),* 13(4),p. 104-113.

Futcher, L. and Solms, v, R. (2007). SecSDM: A Model for Integrating Security into the Software Development Life Cycle. *Springer Science+Business Media,*p. 41-48.

Giardino, C. et al. (2016). Software Development in Startup Companies: The Greenfield Startup Model. *IEEE Computer Society,* 42(6),p. 585-604.

Gilliam, D. (2005). Security risks: management and mitigation in the software life cycle. *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 211-216.

Humayun, M. et al. (2022). Security Threat and Vulnerability Assessment and Measurement in Secure Software Development. *Computers, Materials & Continua*, *71(3),* 5039-5059.

Jeyapriya, S. and Rekha, C. (2020). SDLC Security Framework for Software Startups. *International Journal of Recent Technology and Engineering (IJRTE),* 9(1),p. 406-409.

Khan, A, R. et al. (2022). Systematic Literature Review on Security Risks and its Practices in Secure Software Development. *Institute of Electrical and Electronics Engineers*,p. 5456-5481.

Kudriavtseva, A. and Gadyatskaya, O. (2022). Secure Software Development Methodologies: A Multivocal Literature Review. *Cornell University*.

Lingham, D, A. et al. (2020). Implementation of Security Features in Software Development Phases. *Cornell University*.

Marican, Y, N, M. et al. (2023). Cyber Security Maturity Assessment Framework for Technology Startups: A Systematic Literature Review. *Institute of Electrical and Electronics Engineers,* 11,p. 5442-5452.

Melegati, J. et al. (2020). Towards Specific Software Engineering Practices for Early-Stage Startups. *Springer Science+Business Media*,p. 18-22.

Milewicz, R. et al. (2022). A Secure Future for Open-Source Computational Science and Engineering. *Cornell University.*

Mohino, V, d. et al. (2019). The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies. *Multidisciplinary Digital Publishing Institute*, 8(11),p. 1218-1218.

Moyón, F. et al. (2020). How to Integrate Security Compliance Requirements with Agile Software Engineering at Scale?. *Springer Science+Business Media*,p. 69-87.

Nia, A, M. (2023). An Introduction to Adaptive Software Security. *Cornell University*.

Otieno, M., Odera, D. and Ounza, E, J. (2023). Theory and practice in secure software development lifecycle: A comprehensive survey. *GSC Online Press*, 18(3),p. 053-078.

Parthasarathy, S. (2022). A decision framework for software startups to succeed in COVID-19 environment. *Elsevier BV*, 3, 100037-100037.

Paternoster, N. et al. (2014). Software development in startup companies: A systematic mapping study. *Elsevier BV*, 56(10),p. 1200-1218.

Posthumus, S. and Solms, v, R. (2004). A framework for the governance of information security. *Elsevier BV*, 23(8),p. 638-646.

Rafiq, U. et al. (2021). Analytics Mistakes that Derail Software Startups. *25th International Conference on Evaluation and Assessment in Software Engineering (EASE '21)*, 60–69.

Rajapakse, N, R. et al. (2021). Challenges and solutions when adopting DevSecOps: A systematic review. *Cornell University*.

Rastogi, R. and Solms, v, R. (2005). Information Security Governance - A Re-Definition. *Springer Science+Business Media*,p. 223-236.

Ross, S, R. (2018). Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy. *NIST Special Publication 800-37 Revision 2*.

Souppaya, M., Scarfone, K. and Dodson, D. (2022). Secure Software Development Framework (SSDF) version 1.1. *NIST Special Publication 800-218*.

Steward, C. et al. (2012). Software Security: The Dangerous Afterthought. *Ninth International Conference on Information Technology - New Generations,* pp. 815-818.

Tompkins, G, F. and Rice, S, R. (1986). Integrating security activities into the software development life cycle and the software quality assurance process. *Elsevier BV*, 5(3),p. 218-242.

Wang, X. et al. (2016). Key Challenges in Software Startups Across Life Cycle Stages. *Springer Science+Business Media*,p. 169-182.