I. J. Education and Management Engineering, **2023, 2, 36-42**
Published Online on April 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijeme.2023.02.05

Security-aware Mobile Application Development Lifecycle (sMADLC)

Anthony Wambua Wambua*
Department of Computer Science, School of Computing & Information Technology, Murang'a University of Technology, Murang'a, Kenya
Department of Computer Science, School of Science & Engineering, Daystar University, Nairobi, Kenya
E-mail: awambua@daystar.ac.ke
ORCID iD: https://orcid.org/0000-0001-5110-9071
*Corresponding author

Gabriel Ndung'u Kamau
Murang'a University of Technology, Murang'a, Kenya
Email: gkamau@mut.ac.ke
ORCID iD: https://orcid.org/0000-0001-8433-3766

Received: 17 September, 2022; Revised: 22 October, 2022; Accepted: 28 November, 2022; Published: 08 April, 2023

Abstract: With the high mobile phone penetration and subsequent significant usage of mobile phone applications, mobile users have become prime targets of hackers. Secure Software Development (SSD) advocates incorporating security aspects at the initial stages of software development. This study proposes a novel Mobile Application Development Lifecycle by reviewing SSD concepts and incorporating these concepts into MADLC- a mobile-focused software development lifecycle to create a security-aware Mobile Application Development Lifecycle (sMADLC). The proposed development lifecycle, sMADLC, can potentially help mobile application developers create secure software that can withstand hacker aggression and assure mobile application users of the confidentiality, integrity and availability of their data and systems.

Index Terms: Security, Secure Software Development, Mobile Application Development, CIA, SDLC.

1. Introduction

Mobile phones have become prevalent owing to their decreasing cost, increasing computing power and the convenience they offer. World Bank [1] reported that Sub-Saharan Africa had a mobile phone penetration of 73%, with more developed economies reporting up to 98% mobile phone penetration rate. The same report notes that Internet access in Sub-Saharan Africa is about 30%, while in developed countries, it is at 80%. With such great mobile penetration rates and Internet access levels, businesses have attempted to explore newly created business opportunities. Governments have leveraged this penetration level to bring services closer to the people or offer more access to government services through e-government. Organizations and institutions have simplified their business processes through mobile phones to offer increased convenience and allow people to transact on the move.

Businesses, institutions, and organizations have used mobile applications to exploit these expanding markets. These applications have become very popular with every other institution developing applications. A study by Van Noort and Van Reijmersdal [2] found that branded apps enhanced brand attitude and the relationship between customers and companies. This explains why companies want to acquire mobile apps. Users use such applications to buy products, access government services, access information over the web, for entertainment or even to access learning. As such, these apps have access to confidential and sensitive information such as personal data and credit card information. Android and Apple are the dominant application stores, with android users having more than 2.6 million apps to choose from and Apple users have 2.2 million possible apps to download [3].

Mobile applications, popularly known as apps, pose more security challenges than desktop computers. Elsantil [3] observes that mobile phone design focuses more on portability, ease of access and low power consumption than security. Mobile users' security awareness is lower than desktop users [4]. Many users, reliance on services through mobile

applications, low awareness of security practices amongst users, the low expertise level of mobile applications developers and all-day internet connection of mobile devices are some of the factors that make mobile application users prime targets by hackers. The three critical dimensions of security, as shown in Fig. 1, can easily be compromised.

The confidentiality can be compromised as mobile phones can be stolen, and the wrong people can easily access data through malicious code embedded in mobile applications that users download.

Data integrity can be compromised as information is exchanged over the Internet. Users believe mobile phones are secure since they always have them. As such, they use less secure passwords, if any, increasing the vulnerability of the information they exchange. System Integrity of the mobile phone should be preserved so that the performance is not impaired due to attacks by hackers.

Data availability is easily threatened when hackers through backdoors [5], existent in the mobile applications that users download, deny users services they deserve to access. Hackers can block access to bank accounts and social media accounts.

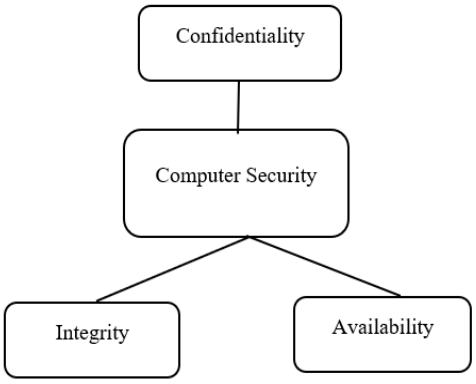

Fig. 1. The CIA Triad

Cope [6] argues out that security must be incorporated right from the beginning of the software development and not at the end of the process as advocated by the traditional Software Development Lifecycles (SDLCs), such as the waterfall model that places testing at the end of all other stages just before deployment.

Researchers also see the need to have separate SDLCs for mobile applications. Kaur and Kaur [7] opine that even though the basic steps in software development, such as requirements gathering, designing, implementation testing and delivery, remain similar for mobile phones and desktop computers, the details are different. Complexities and constraints such as performance, usability and bandwidth require more focused SDLCs. Applying security practices in the traditional SDLCs leads to conflict between business objectives and security engineering goals [8]. Other studies also proposed SDLCs that focused on mobile application development include [9, 10].

Given this background, security researchers and practitioners ought to ensure that mobile phone users are protected against security attacks. This study aims to propose a security aware SDLC for mobile application developers. The rest of the study is organized as follows; Section 1.1 will concretize the definition of the problem, and Section 2 will review attempts by researchers to offer a solution to the problem of the vulnerable software. Then, Section 3 will outline this study's methodology and steps towards developing sMADLC for secure mobile application development. Lastly, Section 4 will reflect and conclude the study.

1.1 Problem Statement

Increased mobile phone penetration and mobile application usage pose security challenges to phone users. The problem is exacerbated by the fact that mobile application development has attracted many developers who are not primarily computer science experts. These developers are unaware of security practices. Application users are unaware of security threats, with many just granting app permissions, oblivious of the consequences. Traditional SDLCs relegate testing to the last stage of SDLCs just before deployment; thus, security is not addressed right from requirements gathering. Mobile phone applications have complexities and constraints that require more focused SDLCs. Mobile applications developed through the traditional and generic software development lifecycle potentially expose app users to security threats.

The study aims to develop a security-aware model for mobile application development to address secure software development through the software development lifecycle. The study's objectives are to review SSD practices and MADLC – a mobile-focused SDLC. Concepts from SSD and MADLC will be merged to create sMADLC.

2. Related Work

This section will review attempts by researchers to incorporate security awareness in software development. Ivaki and Antunes [11] proposed a Security-aware Integrated Development Environment (SIDE). The study argued that Integrated Development Environments (IDE) should be able to predict code vulnerabilities at the early stage of software development, that is, coding. The IDEs should have machine learning-based code security analyzers and security actuators. This framework helps address many security challenges. The limitations include that security considerations start at the point of coding and not at the point of requirements gathering. Security-related issues should be addressed right from the beginning [5]. Further, in the context of mobile application development, the framework does not address issues specific to mobile application development but assumes that all software is the same and thus, addressing security challenges at the coding stage would suffice.

Goel [12] proposed a Security-aware Requirements Elicitation, Assessment and Design Methodology (SecREAD). The study argues that security practices should be considered right from gathering requirements and assessing the requirement to the system's design. Having security considered from the requirement gathering stage is bound to build secure systems. However, security practices must continue through all the stages of the SDLC [13]. The methodology also does not consider special requirements of mobile phone application development.

Authors have also attempted to raise security awareness among mobile software developers, who are regarded as the weakest link in combating insecurity. Qian, et al. [5] proposed an innovative and active Secure Software Development (SSD) approach. The approach advocated for SSD in institutions of higher learning. A module was developed and used to train students on how to develop secure mobile applications. The developed course modules can easily be plugged into many software development courses in computer science curriculums. The study reported positive feedback and increased student efficacy in secure mobile application development. Introducing a module within computer science software development is an excellent approach to solving the security problem at the software development level since learners have a hands-on approach. The limitation is that not all mobile application developers are formally trained. A study by Wambua and Maake [14] established that 5% of software developers either have no formal software development training or only hold a certificate.

3. Research Methodology

To meet this study's goal, that is, to develop a security-aware model for mobile application development, the steps followed in this study are shown in Fig.2 and explained after that. The steps will address the study's objectives and thus help in developing the secure mobile application development lifecycle.

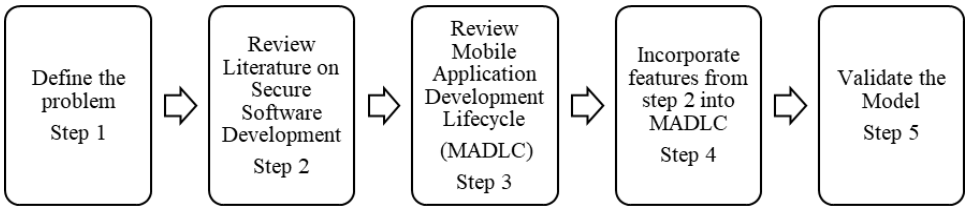

Fig. 2. Study Methodology

Step 1: Define the problem: This will involve the formal definition of the problem so that a model to solve it can be conceptualized. This has been done in Section 2 of the paper.

Step 2: Review of literature on secure software development: This will involve teasing out concepts of security ware software development and secure software development. The same keywords will be used to search research databases. Some of the studies have already been discussed in Section 3 of the paper.

Step 3: Review literature on mobile-focused SDLCs: Mobile application development has its own complexities and constraints that open call for more mobile-focused SDLCs. This will be reviewed in the study. The assumption is that adding security aspects reviewed in Step 2 would yield a security-aware-mobile development model.

Step 4: Modelling features from Step 1 and Step 2: This is the step in which the security-aware model for mobile application development will be conceptualized based on the features and concepts learned in steps 2 and 3.

Step 5: Model validation: In this step, the model developed in step 4 will be validated. The validation is two-fold.

i. Experts focus group: The model will be subjected to both experts in Software Engineering and Security. Their opinions and suggestions will then be implemented in the model. Depending on time and experts' availability, this step can be done iteratively to eliminate any bias and increase the study's validity.

ii. Survey mobile software developers: This will be a large-scale survey with the following objectives: -

1) To determine if the proposed model would be an effective SDLC for mobile application development.

2) To determine if the prosed security practices sufficiently address security concerns regarding SSD in mobile application development.
3) Any comments for improvement that the mobile application developers have.

3.1 Mobile Application Lifecycle (MADLC)

Vithani and Kumar [9] proposed a mobile-focused SDLC called MADLC – Mobile Application Development Lifecycle. The proposed SDLC has the stages depicted in Fig. 3.

Fig. 3. Pictorial depiction of MADLC

Table 1. MADLC stages as described by [9, 15]

No	Stage	Process
1.	Identification	New ideas are collected, or an existing one is improved through brainstorming. The novelty of the idea is established. Initial requirements gathering is completed.
2.	Designing	Initial development of the design. A decision is made on whether this should be a trial of the full version. Functional requirements are defined. A storyboard for the user interface describing the application flow is created.
3.	Development	An appropriate programing language is used to code the application's functional requirements and user interface
4.	Prototyping	The application is tested to see if user requirements are met. Experts do the testing. The application is also sent to the client, and the feedback is incorporated – this can be iterated until all functional requirements are met.
5.	Testing	Testing through both virtual devices and real devices is conducted. The application is also tested on all targeted platforms are also
6.	Maintenance	The continuous support and improvement of the application. Fixing of bugs discovered by users. Incorporating user-changing needs.

MADLC accommodates the unique features and complexities of developing mobile applications, such as battery life, number of screens, cross-platform development and limited interfaces but does not entrench security practices.

3.2 Secure Software Development Concepts

The literature has identified certain practices to enhance security at different stages in the software development lifecycle. This has come to be appreciated by researchers and practitioners as Secure Software Engineering (SSE). These practices have been incorporated into the general SDLCs by some approaches, including Microsoft Security Development Lifecycle (SDL) [16], McGraw's Touchpoints [17], and Comprehensive Lightweight Application Security Process (CLASP) from Open Web Application Security Project (OWASP) organization [18].

McGraw defines the seven touchpoints listed below that every secure software development lifecycle should incorporate at different stages during software development.

1. Code review
2. Penetration testing
3. Risk-based security tests
4. Security requirements
5. Abuse cases
6. Architectural risk analysis
7. Security operations

As for Microsoft's SDL, the following security practices are added to the software development lifecycle. In the requirement gathering phase, user security feature requirements are defined, while in the design phase, the MS SDL threat modelling for security risk identification is carried out.

In the implementation phase, static analysis code is performed. At completion, security testing of critical components of the software is done. The final security review then follows.

On the other hand, the CLASP by OSWAP defines twenty-four security-related operations. These operations may be fully or partially implemented into the SDLC during the software development. The exact application depends on the type of software being developed.

CLASP advocates for threat modelling being carried out at the requirements elicitation phase, just like Microsoft's SDL. It further advocates for secure coding practices during the design and implementation phase, including static code analysis and inspections. In the final stage, CLASP advocates for security testing to be carried out.

In the next section, to build a Security-aware Mobile Application Development Lifecycle(sMADLC), security practices identified from these three leading secure software development lifecycles will be applied to the unique stages of developing mobile applications as described in MADLC.

3.3 Security-aware Mobile Application Development Lifecycle (sMADLC)

Table 2. sMADLC stages

No	Stage	Process	Security
1.	Identification	New ideas are collected, or an existing one is improved through brainstorming. The novelty of the idea is established. Initial requirements gathering is completed.	Perform threat modelling Identify possible attacks on the CIA triad Identify possible privacy breaches Define Security Metrics and acceptable levels
2.	Designing	Initial development of the design. A decision is made on whether this should be a trial of the full version. Functional requirements are defined. A storyboard for the user interface describing the application flow is created.	Define security requirements Non-functional security-related requirements are defined Security business goals are defined
3.	Development	An appropriate programing language is used to code the application's functional requirements and user interface	Perform Static Analysis Security Testing (SAST) Application developed in Security-aware Integrated Development Environment (SIDE) [11] Encryption of databases Definition of the needed user- permissions. Application of recommended code patterns
4.	Prototyping	The application is tested to see if user requirements are met. Experts do the testing. The application is also sent to the client, and the feedback is incorporated – this can be iterated until all functional requirements are met.	Verification of security-related non-functional requirements Verification of business security goals
5.	Testing	Testing through both virtual devices and real devices is conducted. The application is also tested on all targeted platforms are also	Perform Dynamic Analysis Security Testing (DAST) Security testing Penetration testing
6.	Maintenance	The Continuous support and improvement of the application. Fixing of bugs discovered by users. Incorporating user-changing needs.	Establish a Standard Incident Response Process Code refactoring to eliminate anti-patterns that bring vulnerability. Continual improvement of security measures

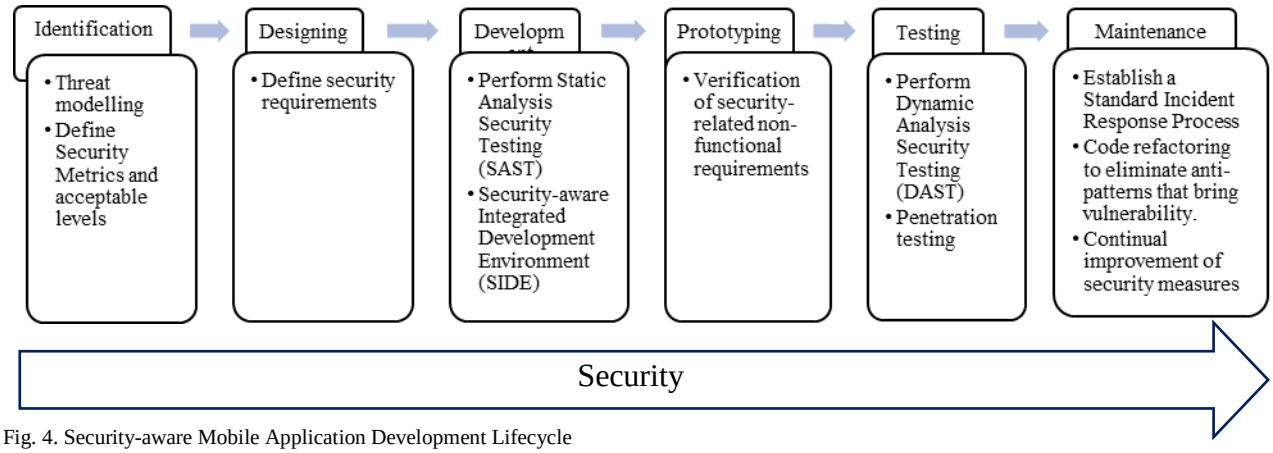

Fig. 4. Security-aware Mobile Application Development Lifecycle

Features of Secure Software Development (SSD) are incorporated into the Mobile Application Development Lifecycle (MADLC) to come up with Security-aware Mobile Application Development Lifecycle (sMADLC), as depicted in Fig. 4. The specific stages of sMADLC are illustrated in table 2. By incorporating specific and relevant security activities from SSD into every stage in MADLC, security will be addressed from Identification - the initial stage of the mobile application development to maintenance – the last stage in the lifecycle. Security should never be

viewed as a post-development undertaking if developed systems are to withstand hacker aggression and maintain privacy and integrity.

4. Conclusion and Future Work

This study sought to address the problem of insecurity in mobile application development. With higher mobile phone penetration, users have turned to the use of mobile applications for business, entertainment, or learning, making them prime targets for hackers. The study explored the concept of Secure Software Development (SSD), which advocates for incorporating security issues right from the first stage of software development to the last stage instead of focusing on security only at the testing stage. The study incorporated these concepts into a mobile-focused SDLC, Mobile Application Development Lifecycle -MADLC, to develop Security-aware Mobile Application Development (sMADLC).

Authors posit that the model can potentially assist mobile application developers in addressing the myriad of threats directed at mobile app users. Addressing security issues right from the beginning of the software development lifecycle and using mobile-focused SDLCs allow for addressing specific issues in mobile development. The study has proposed sMADLC that merges the two concepts, which have been in the past developed as independent areas, to address secure software in the context of mobile application development

In future, the model will be validated by experts and mobile application developers through surveys to understand the appropriateness of the model and the extent to which developers feel that the model addresses the security challenges in application development.

References

[1] W. Bank, "World Development Report 2016," in "Digital Dividends," DC, 2016.

[2] G. Van Noort and E. A. Van Reijmersdal, "Branded apps: Explaining effects of brands' mobile phone applications on brand responses," *Journal of Interactive Marketing*, vol. 45, pp. 16-26, 2019.

[3] Y. Elsantil, "User perceptions of the security of mobile applications," *International Journal of E-Services and Mobile Applications (IJESMA)*, vol. 12, no. 4, pp. 24-41, 2020.

[4] R. Bitton, A. Finkelshtein, L. Sidi, R. Puzis, L. Rokach, and A. Shabtai, "Taxonomy of mobile users' security awareness," *Computers & Security*, vol. 73, pp. 266-293, 2018/03/01/ 2018, doi: https://doi.org/10.1016/j.cose.2017.10.015.

[5] K. Qian, D. Lo, R. Parizi, F. Wu, E. Agu, and B. T. Chu, "Authentic Learning Secure Software Development (SSD) in Computing Education," in *2018 IEEE Frontiers in Education Conference (FIE)*, 3-6 Oct. 2018 2018, pp. 1-9, doi: 10.1109/FIE.2018.8659217.

[6] R. Cope, "Strong security starts with software development," *Network Security*, vol. 2020, no. 7, pp. 6-9, 2020.

[7] A. Kaur and K. Kaur, "Suitability of existing software development life cycle (sdlc) in context of mobile application development life cycle (madlc)," *International Journal of Computer Applications*, vol. 116, no. 19, 2015.

[8] K. Rindell, S. Hyrynsalmi, and V. Leppänen, "Aligning security objectives with agile software development," presented at the Proceedings of the 19th International Conference on Agile Software Development: Companion, Porto, Portugal, 2018. [Online]. Available: https://doi.org/10.1145/3234152.3234187.

[9] T. Vithani and A. Kumar, "Modeling the mobile application development lifecycle," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2014, vol. 1, pp. 596-600.

[10] A. Kumar and T. Vithani, "A comprehensive mobile application development and testing lifecycle," in *2014 IT Professional Conference*, 2014: IEEE, pp. 1-27.

[11] N. Ivaki and N. Antunes, "SIDE: Security-aware Integrated Development Environment," in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 12-15 Oct. 2020 2020, pp. 149-150, doi: 10.1109/ISSREW51248.2020.00056.

[12] R. Goel, "Secread: Security-Aware Requirements Elicitation, Assessment And Design Methodology," MNIT, Jaipur, 2018.

[13] H. Assal and S. Chiasson, "'Think secure from the beginning' A Survey with Software Developers," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1-13.

[14] A. Wambua and B. Maake, "Characterizing Software Quality Assurance Practices in Kenya," *International Journal of Software Engineering and Computer Systems*, vol. 8, no. 1, pp. 22-28, 2022.

[15] L. Shanmugam, S. F. Yassin, and F. Khalid, "Incorporating the elements of computational thinking into the Mobile Application Development Life Cycle (MADLC) model," *Int. J. Eng. Adv. Technol*, 2019.

[16] Microsoft. "Microsoft Security Development Lifecycle (SDL)." https://www.microsoft.com/en-us/securityengineering/sdl/ (accessed July 5, 2022).

[17] G. McGraw, "Software security," *Building security in*, 2006.

[18] A. Hudaib, M. Alshraideh, O. Surakhi, and M. Alkhanafseh, "A Survey on Design Methods for Secure Software Development," *International Journal of Computer and Technology*, vol. 16, pp. 7047-7064, 12/10 2017, doi: 10.24297/ijct.v16i7.6467.

Authors' Profiles

Mr Anthony Wambua Wambua, is currently pursuing his PhD at Murang'a University of Technology**,** he received his Bachelor of Applied Computer Science from Periyar University, India, in 2009 and Master of Computer Science from Bharathiar University, India, in 2011. He is currently pursuing a Ph.D. in Computer Science and currently working as a lecturer in the Department of Computer Science, Daystar University, Kenya, since 2014. Mr Wambua is a member of IEEE & ACM. His main research work focuses on Software Engineering, Metaheuristic Algorithms and eLearning. He has eight years of teaching experience.

Dr. Gabriel Ndung'u Kamau, received his Bed (Art) degree in Mathematics and Business in 1999 from Kenyatta University and Master of Business Administration (Management Information Systems) in 2008 and PhD in Strategic Information System in 2017 from University of Nairobi. Gabriel is also a Certified Network Security Specialist (2020) and Big Data Analyst (2019). Gabriel was a teaching assistant lecturer with Department of Computer and Information Technology, Kenya Methodist University from 2009 to April 2013. In June, 2013, Gabriel Joined Murang'a University of Technology as a lecturer in the department of Information Technology. Gabriel has a vast knowledge and teaching experience in the area of management information systems, information security and applied cryptography, computer forensics, enterprise risk management of information systems, and IT Governance. His research interest includes ICT4D, cybersecurity and forensics, data analytics, computing and information technology philosophy perspectives.

How to cite this paper: Anthony Wambua Wambua, Gabriel Ndung'u Kamau, "Security-aware Mobile Application Development Lifecycle (sMADLC)", International Journal of Education and Management Engineering (IJEME), Vol.13, No.2, pp. 36-42, 2023. DOI:10.5815/ijeme.2023.02.05