International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

SDLC Security Framework for Software Startups

S. Jeyapriya, C. Rekha

Abstract*: *A major issue for software startup firms failing to develop strong security programs is that the perceived start-up price to hold it out. Secure software Development Life Cycle (SDLC) framework used for software developers, in saving the project cost throughout the first integration of design. As several developers advocate, software organizations usually adopt atop-down approach to implement secure SDLC methodologies. Interview was finished with developers within the software startups presently used in industry to explore real-life software security practices throughout every stage of the development lifecycle. The aim of the proposed framework is to assist developers in startups to explore real-life software security practices throughout every stage of the development lifecycle.

Keywords: Software, SDLC, Framework,* Security, *Startup.

I. INTRODUCTION

Even today, several organizations do not include information Security (IS) personnel within the choices created throughout the Software Development Life Cycle (SDLC) till they are able to implement a completely developed application. Several software developers, system analysts, software architects, business analysts, and project managers believe that information security is that the responsibility of the network administration employees, and as long as servers, firewalls, routers, and other hardware are organized properly, and that operating system is organized properly and patched frequently, then any applications they deploy on those servers, behind the firewalls, will be secure. Standards organizations like the International Organization for Standardization (ISO), The National Institute of Standards and Technology (NIST), The Software Engineering Institute (SEI), The Open web Application Security Project (OWASP), et al are providing guidance and standards for application security for years. A number of these organizations (NIST, ISO) have the facility to enforce the standards they publish, a minimum of for a restricted audience, whereas others merely plan to give guidance and help for those desire to make safer applications (OWASP). However, because most of the guidance provided by these organizations does not have the rule of law behind them, not like those made by a legislative or regulatory agency, the impact they had on creating safer applications is tough to accurately assess. Project start-up is defined as a short-term systematic method resulting in project execution. Literature so far lacks emphasis on a definition of the project start-up phase and also the tasks conducted throughout this phase (Land et.al, 2019).

Manuscript received on April 03, 2020.
Revised Manuscript received on April 18, 2020.
Manuscript published on May 30, 2020.
* Correspondence Author
S.Jeyapriya*, Department of Computer Applications, Madurai Kamaraj University College, Madurai, India.
C.Rekha, Department of Computer Applications, Madurai Kamaraj University College, Madurai, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Traditionally several companies are familiar with software Development Life Cycle (SDLC) that could be a series of steps followed to produce a framework backed by proper analysis, design, implementation and maintenance. after developing and simply before deploying the application live into the market, several firms notice the security loopholes and choose a Vulnerability Assessment and Penetration Testing, traditionally called VAPT, to seek out the security vulnerabilities present within the software. Software Development life cycle (SDLC) could be a religious model utilized in project management that defines the stages included in an information system development project, from an initial feasibility study to the maintenance of the finished application. There are different software development life cycle models specify and design, that are followed throughout the software development phase. These models are known as "Software Development process Models." Every method model follows a series of phase distinctive to its type to make sure success within the step of software development that is shown in figure1.

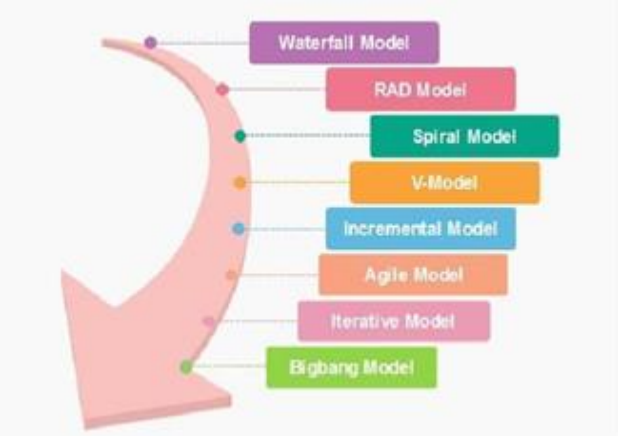

Fig. 1.Various SDLCModels

Startups are agents of modification that bring in innovations and realize solutions to issues at numerous scales. An all-rounded engineering team may be a key driver for the power to execute the entrepreneurial ambition, from building a minimum viable product to later stages of product vision (Devadiga, 2017). Startups face a dynamic environment and want to beat many challenges so as to become successful. One among these challenges is related to the software requirement process (Chanin et.al, 2017). Pompermaier (2019) presents a proposal of a minimal Viable Development process for be used throughout the MVP dev The goal is to create security so as to reduce the chance that people with malicious intent are able to manipulate applications and access, steal, modify, or delete sensitive and vital data. for several years, security was an afterthought in softwaredesign.

II. BACKGROUNDLITERATURE

Startup companies have become a significant provider of innovation and software intensive products. The adaptability and responsiveness of start up companies empowers quick advancement and launch of innovative products(Klotins et.al, 2019). Scientists contend that Design Thinking can add to software development by offering support on the best way to comprehend client needs so as to infer arrangement and item choices, though Lean Startup assists with finding out about business and scaling strategies (Dobrigkeit et.al, 2019).software startups create innovative products through which they scale their business quickly, and in this way, offer some benefit to the economy, including job generation. However, most startups fail inside two years of their launch because of a poor problem-solution work and negligence of the learning method throughout minimum viable product (MVP) development (Tripathi et.al, 2019). There are numerous root causes of software failures. Few years agone, software used to fail chiefly due to functionality related bugs. That used to happen because of demand misunderstanding, code issues and lack of practical testing. plenty of work has been done in past on this and software engineering has matured over time, because of that software's hardly fail because of functionality related bugs (Chhillar & Sharma, 2019). Cloud computing may be a terribly rapidly growing technology with a lot of facilities however also with a lot of problems in terms of vulnerabilities before and after deploying the applications into the cloud (Vijayakumar & Arun, 2019). Expanding awareness and research concentrated on the cyber security scene has brought about an enormous push for "shifting security left" in the SDLC. With security designing groups connected before and all the more frequently all through the SDLC, security issues will be found and fixed before, which builds efficiency while bringing down expense and overhead (Nguyen and Dupuis, 2019). A prologue to data security is given, trailed by a study hall case of an fictitious company, Fun and Fitness, during the time spent refreshing its web based business website for class enrollments (Spears and Parrish, 2019). Gathering secure estimation is the initial move toward the created of comprehensive secured software. Security is an immaterial measure additionally considered as a non-practical credit which should be evaluated in some way utilizing devices and strategies during the primer stages, i.e., the necessities in building phase of software development process (Poonia et.al, 2020).People peoples use software bearing in mind that it is reliable and can be trust upon and the activity they perform is made sure about. Presently,if these product have exploitable security gap at that point how might they be alright for use (Banerjee& Pandey, 2009). Security usage in software during its beginning times ofimprovement guarantees flaw free programming. The security necessities are subjective in nature along these lines they ought to be changed over into quantitative measure with the assistance of measurements (Banerjee et.al, 2014). In the present data age, programming is assaulted intentionally bringing about break of security and individuals' trust. These malevolent assaults give mischief to people, associations, and the world everywhere (Banerjee et.al, 2016). As figuring has become some portion of every human action, our interests about the security of so¬ftware system have developed (Fenton &Bieman, 2014).The data frameworks controlling our basic foundation are helpless against cyber attack. cyber war is in this manner unavoidable except if we improve our digital protections. The best way to do this is by building security into system at the design stage (McGraw,2013).A absence of security measurements connotes that it is beyond the realm of imagination to expect to gauge the success of security strategies, mechanisms and implementation, and security can't, thus, be improved if it can't be estimated (Mellado et.al, 2010).

III. SDLC SECURITYFRAMEWORK

The objective is to incorporate safety efforts with applications, so as to minimize the probability that people with vindictive goals will have the option to manipulate applications and access, steal, change, or erase sensitive and significant information. For many years, security was an untimely idea in software design. Indeed, even today, many organizations do exclude Information Security (IS) faculty in the decisions made during the Software Development Life Cycle (SDLC) until they are ready to execute a completely evolved application.Standards associations, for example, the International Organization for Standardization(ISO), the National Institute of Standards and Technology (NIST), the Software Engineering Institute (SEI), the Open Web Application Security Project (OWASP),and others have been giving direction and benchmarks to application security for years. A portion of these associations (NIST, ISO) have the ability to enforce the measures they distribute, in any event for a restricted crowd, though others simply attempt to give direction and help to those wishing to make more secure applications(OWASP).

Requirement Phase:

Examples of software system wellbeing prerequisites incorporate risky modes, for example,

- Out-of-sequence;
- Inappropriatesize;
- Inadvertentorder;
- Adverse condition;
- Wrongdeadlocking;
- Failure to order.

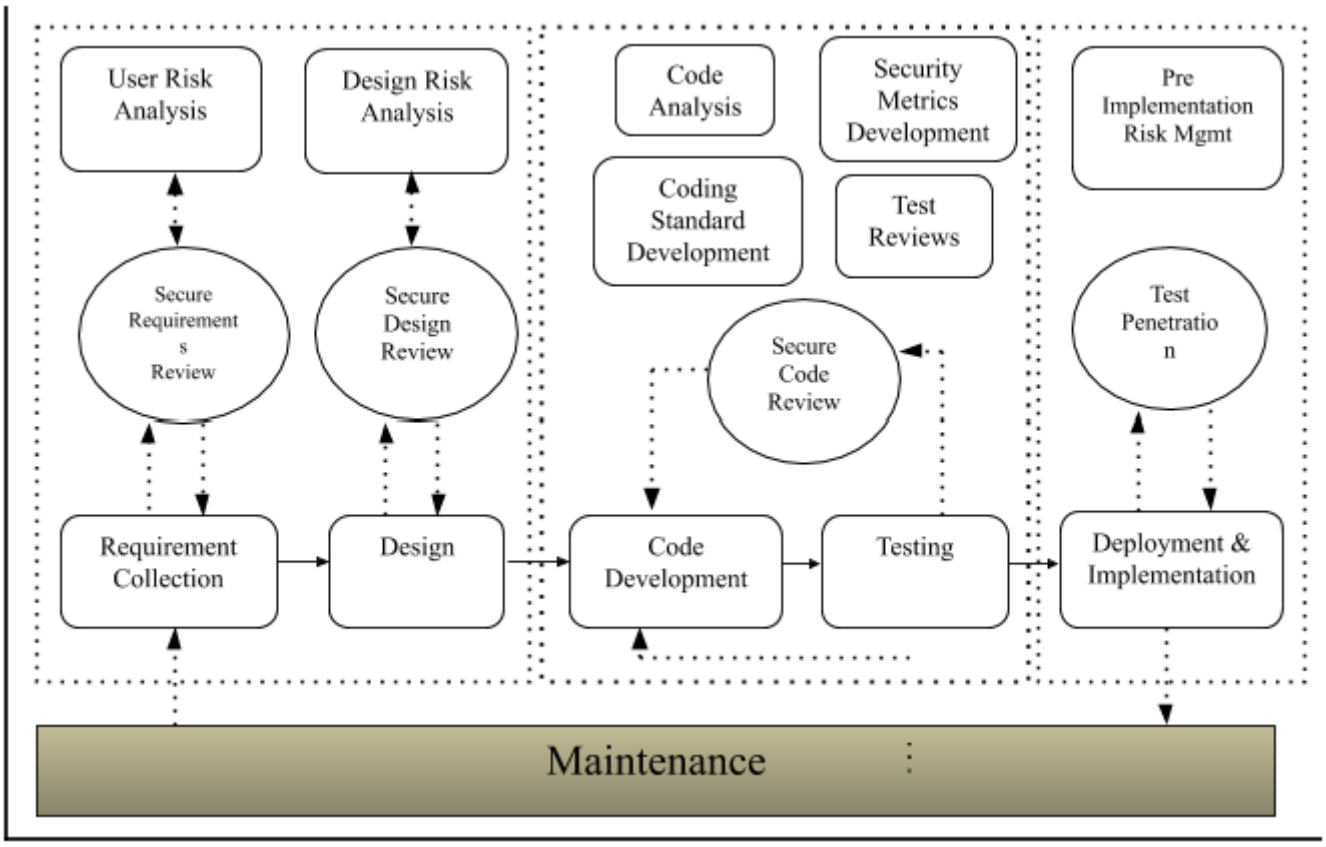

Fig. 2. SDLC Security Framework

A portion of these modes, for example, out-of-sequence and wrong deadlocking,are inherently inside the software system themselves; others identify with out-of range data or figuring results that may emerge from a blend of software and data entry errors; others identify with the earth or setting inside which the software system worksThese conditions can be overseen legitimately through the gadget designers and programmers, and others can also be out in their control anyway ought to be anticipated and represented in the design.A technique and obligations for risk management, and reporting the starter realized risks is accommodated Key exercises that must occur sooner or later of this stage join setting up. In any event the records ought to be distinguished by the mission administrators , and organize dangers to the gadgets. This framework should comprise of distinguishing resources for be secured and doling out their criticality in expressions of secrecy, respectability, and accessibility; deciding the dangers and coming about peril to those property, as appropriately as the overall or arranged controls to reduce that chance. The mission supervisors organize to concentrate assets on districts with the most elevated chance.The necessities and specs must be changed to comprise of new necessities for extra assurance controls distinguished during this stage when it isrequired.

Design Phase:

Right now, necessities are converted into solution, so precise characterization of benefit criticality and intentional controls are essential to effective improvement or procurement. For instance, if a system with an interest to transmit data over an open system and the criticality rating for the confidentiality of that records is high, at that point some will control, comprehensive use encryption or a virtual non-open network may turn into a piece of the appropriate response. As the developed gadget , testing of each control is must to guarantee that the controls complete asplanned.

Code Development and Testing Phase:

During this stage, the system is carried out and arranged in the structure that it should be worked. In this segment testing is also important , especially to check whether the planned security controls are operational inside the joined condition. The system proprietor needs to guarantee whether the endorsed controls, including any real or procedural controls, are in the past region to the system going live.Static frameworks arevery gadget are expected.The intends is fundamental and it is notable setup the executives system permits us to make certain changes to the gadget hardware,software or supporting techniques are evaluated and endorsed preceding usage. The subsequent exchange to the danger posture of the machine is the piece that is from time to time ignored .Any substitute to a framework can diminish the viability of existing controls, or to in some other case have some effect at the classification, accessibility, or respectability of the machine.A risk evaluation step is secured in assessing system alterations is the solution.

For organizations that choose a design control board, the including the risk administrator or safety expert to this body can encourage the blending of threat evaluation into arrangement the executives.

Deployment Phase:

This section manages the substitution progress as well as system removal. If a risk management plan was created at project inception, at that point the risk ought to be distinguished to secrecy of leftover information during this stage.

IV. RESULT

The SDLC Security Framework used by the software industry to design, develop and test high quality softwares. The proposed framework aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates. Interview was done with the developers within the software startups presently used in industry to explore real-life software security practices throughout every stage of the development lifecycle. As a result of the proposed SDLC security framework is to assist developers in startups to explore real-life software security practices throughout every stage of the development lifecycle. It was the first of a new group of life cycle approaches that seek to articulate the critical elements of security to be embedded within any existing development life cycle such that security is appropriately considered as part of normal development.

V. CONCLUSION & FUTURE RESEARCH

The work introduced here, be that as it may, is constrained to the account audit of research works from the academia, industry, and for the proposals of the standard organizations. This work checked is intended for demonstrating security in the existence life cycle of software development for software setups and the proposed structures can be utilized as a brisk reference for arranging the execution of suggested counter measures in a effective manner to address the vulnerabilities. Future work centers around consolidating the diverse security standards to locate a further developed Security measures for contrasting present projects with historical projects to give the best SDLC for large projects in newcompanies.

REFERENCES

1. Banerjee, C., & Pandey, S. K. (2009). Software security rules. *SDLC Perspective. arXivpreprint*.
2. Banerjee, C., Banerjee, A., & Murarka, P. D. (2014). Evaluating the relevance of prevailing software metrics to address issue of security implementation in SDLC. *International Journal of Advanced Studies in Computers, Science and Engineering, 3*(3),18.
3. Banerjee, C., Banerjee, A., & Pandey, S. K. (2016). MCOQR (misuse case-oriented quality requirements) metrics framework. In *Problem Solving and Uncertainty Modeling through Optimization and Soft Computing Applications* (pp. 184–209). IGIGlobal.
4. Chanin, R., Pompermaier, L., Fraga, K., Sales, A., & Prikladnicki, R. (2017, May). Applying customer development for software requirements in a startup development program. In *2017 IEEE/ACM 1st International Workshop on Software Engineering for Startups (SoftStart)* (pp. 2-5).IEEE.
5. Chhillar, D., & Sharma, K. (2019). Proposed T-Model to cover 4S quality metrics based on empirical study of root cause of software failures. *International Journal of Electrical & Computer Engineering (2088-8708),9*(2).
6. Devadiga, N. M. (2017, November). Software engineering education: Converging with the startup industry. In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)* (pp. 192-196).IEEE.
7. Dobrigkeit, F., de Paula, D., & Uflacker, M. (2019). InnoDev: a software development methodology integrating design thinking, scrum and lean startup. In *Design Thinking Research* (pp. 199-227). Springer,Cham.
8. Fenton, N., & Bieman, J. (2014). Software metrics: A rigorous and practical approach. CRCPress.
9. Klotins, E., Unterkalmsteiner, M., & Gorschek, T. (2019). Software engineering in start-up companies: An analysis of 88 experience reports. *Empirical Software Engineering*, *24*(1),68-102.
10. Land, L., Tandjung, M., Low, G., Chin, W. W., Nelson, R., &Fung,
11. K. H. (2019). Start-Up Tasks for Software Development Projects from Customer and Vendor Perspectives.
12. McGraw, G. (2013). Cyber war is inevitable (unless we build security in). *Journal of Strategic Studies*, *36*(1),109-119.
13. Mellado, D., Fernández-Medina, E.,& Piattini, M. (2010, August). A comparison of software design security metrics. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (pp. 236–242).ACM.
14. Nguyen, J., & Dupuis, M. (2019, September). Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education* (pp.93-98).
15. Pompermaier, L., Chanin, R., Sales, A., & Prikladnicki, R. (2019, November). MVP Development Process for Software Startups. In *International Conference on Software Business* (pp. 409-412). Springer,Cham.
16. Poonia, A. S., Banerjee, C., Banerjee, A., & Sharma, S. K. (2020). Proposed Algorithm for Creation of Misuse Case Modeling Tree During Security Requirements Elicitation Phase to Quantify Security. In *Performance Management of Integrated Systems and its Applications in Software Engineering* (pp. 21-28). Springer, Singapore.
17. Spears, J. L., & Parrish Jr, J. L. (2019). Is security requirements identification from conceptual models in systems analysis and design: The Fun & Fitness, Inc. case. *Journal of Information Systems Education*, *24*(1),2.
18. Tripathi, N., Oivo, M., Liukkunen, K., & Markkula, J. (2019). Startup ecosystem effect on minimum viable product development in software startups. Information and Software Technology, 114,77-91.
19. Vijayakumar, K., & Arun, C. (2019). Continuous security assessment of cloud based applications using distributed hashing algorithm in SDLC. *Cluster Computing*, *22*(5),10789-10800.