Volume: 12 Issue: VI Month of publication: June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63494

Constraints that Hinders Secure Software Implementation and Development Processes

Babagana Ali Dapshima[1], Samaila Kasimu Ahmad[2], Khadija Mika Dawud[3]
[1, 2]Department of Computer Science and Engineering, Sharda University, Greater Noida UP India
[3]Department of Agricultural Engineering, Sharda University, Greater Noida UP India

Abstract: Internet of a things have change the entire world in the 21st century, software applications and system are now becoming part of human endeavour. The widely reliance software system is now an issue of concern, because almost everything aspect of human existences dwelled on it. Considering such intimacy and it used in storing of sensitive information and data in almost domain used by human, the need to come up with strategies that will overcome factors affecting implementation of secure software development process has become paramount important. However, most research stress building a secured software but with limited emphasis on the challenges that lead to poor implementation of secure software development process. The paper aims at evaluating the constraints that hinders secure software implementation and development process. Forty-five studies were reviewed using the Systematic literature review and concluded thirteen (13) factors affects successful implementation of secure software development practice.

Keywords: Constraints, implementation, security, software, development

I. INTRODUCTION

Rapid development in business strategies has been observed among organizations especially in the of e-commerce, where business transactions at the satisfaction of customers are done with ease and maximum profits[1][2]. Nowadays, almost all organizations rely on internet to carry out their daily operational activities. Internet applications in recent times have become an issues of concern due to threat posed on it by hackers, because networks activities are observed by Intrusion detection system and firewalls[3][4]. Attacking internet (Web Application) is a serious problem to organizations due to the risk at which their activities will be due to insecurity [5] [6]. Substandard construction of software exposed the weakness of software security and create a better chance for penetration by hackers or unauthorised users [7]. Insecurity in software is scenario that came into existence due negligence resulting from security measured not taken during the early stages of software implementation and development stages [8]. Software quality and reliability depends on the level of security entrusted in it, and this can be achieved through adequate planning and thorough design and implement process [9]. Thus, the need for integrating security throughout the developmental stages of software becomes very necessary, which gives birth to the research paper ‘‘the constraints that affects secured software development practices’’

II. REVIEW OF RELATED LITERATURE

Numerous research related to software security using different methods and approaches toward SDLC (Software Development Life Cycle) have been explored. Research carried out in UCL (University College London) came up with AEGIS (Appropriate and Effective Guidance in Information Security), that uses a unified modelling for integrating security in software, in which the model used (spiral model) defines the overall system [10]. The model contributes tremendously toward dealing with designing secure system, identifying risks and analysing vulnerabilities and potential threats to ensure secured system. However, with all effort geared toward security measured by this model, it lacks experts in the field of security, and decision in term of security in the system is decided by the stakeholders [11]. In NUA (Nigeria University of Agriculture), a model named SSDM (Secure software Development Model) was invented in order to integrate activities that will strengthen the security of a system in the engineering processes [12]. The activities carried out by this model are security review, testing, training, specification, and threat modelling.

Similarly, the concept of Comprehensive Lightweight Application Security process (CLASP) was initiated toward provision of necessary practices that should be adhered to obtained a secure and reliable software [13]. The model laid down seven (7) activities which if strictly followed will provide a secure software or system free from threat and vulnerabilities. The main reasons behind the seven practices is to promote higher level of effectiveness, reliability, risk analysis, code review, testing, as well as adequate requirement for security and software operations [14].

Thus, with all the promising results exhibited by the model toward provision of adequate security measures in software, it only participates much toward building secure software but limited to constraints that affects secure software successful implementation process [15].
More also, in an attempt to reduce the vulnerabilities in software, Microsoft Company developed model as part of security measures [16].

A. Secure Software Development Life Cycle

The process of providing adequate security to a system or software at the initial stage to the final stage, which involves designing building and testing of a software with the capability of resisting malicious attack or free from vulnerability is termed as Secured Software Development life cycle[17].

B. Phases of Secure Software Development Life Cycle

Basically, Secure software development life cycle is processes by five (5) namely;

1) *The Requirement Phase:* This is initial stage of secured software development, which involves collection or gathering of vital information regarding the security measured to integrated starting from the implementation process to deployment stage from the stakeholder[18].
2) *The Design Phase:* This is process of process of interpreting the requirement gathered by stakeholder in a professional technical term. (How the requirement should look like) [19].
3) *Development Phase:* This involves implementation of the design into the actual application.in this phase, secure code practice is very importance. Also since most software are not built from scratch, security measures such as checking vulnerabilities in open source library is importance [19].
4) *Verification Phase:* This the stage where the application undergoes thorough verification process by ensuring its meets the requirement and standard of the software needed as well testing the strength of the software in term of security measures taken [18] [19].
5) *Evolution and Maintenance Phase:* This involves maintaining the software in case of failure and also upgrading the software to the updated version from time to time. This phase is used for patching vulnerability as various techniques or approach for attack keep changing [20] [21] [22].

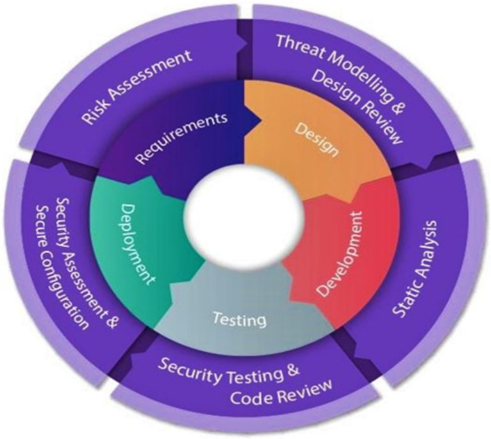
Risk Assessment
Threat Modelling & Design Review
Static Analysis
Security Testing & Code Review
Security Assessment & Secure Configuration
Requirements
Design
Development
Testing
Deployment

Fig 1: Diagram showing various phases of secure software development life cycle

III. RESEARCH METHODOLOGY

Survey literature review approach was used to evaluate the constraints which hinders implementation of secure software process. The methodology is categories into three (3) review stages: Planning, Conducting and Reporting review respectively.

A. Planning Methodology

This involves how to examine the constraints that affect successful process implementation of secure software. In an attempt to identify the constraints, the following question is considered.
What are the constraints that hindered successful development and implementation of secure software?

B. Searching Methodology

This is a process of exploring of vital information related to implementation of secure software in database of scientific journals, libraries etc. examples were such information's can be obtained are; IEEE Digital library, Springer link, Scopus, Taylor and Francis, Science direct etc.

C. Conduction and Decision Inclusion Review

In this stage, information is selected based on add and drop criteria. The add signifies information that are to be included in extraction of data while the drop signifies the information that may not be included in the review paper. Figure 1 on shows the how add and drop process is carried out.

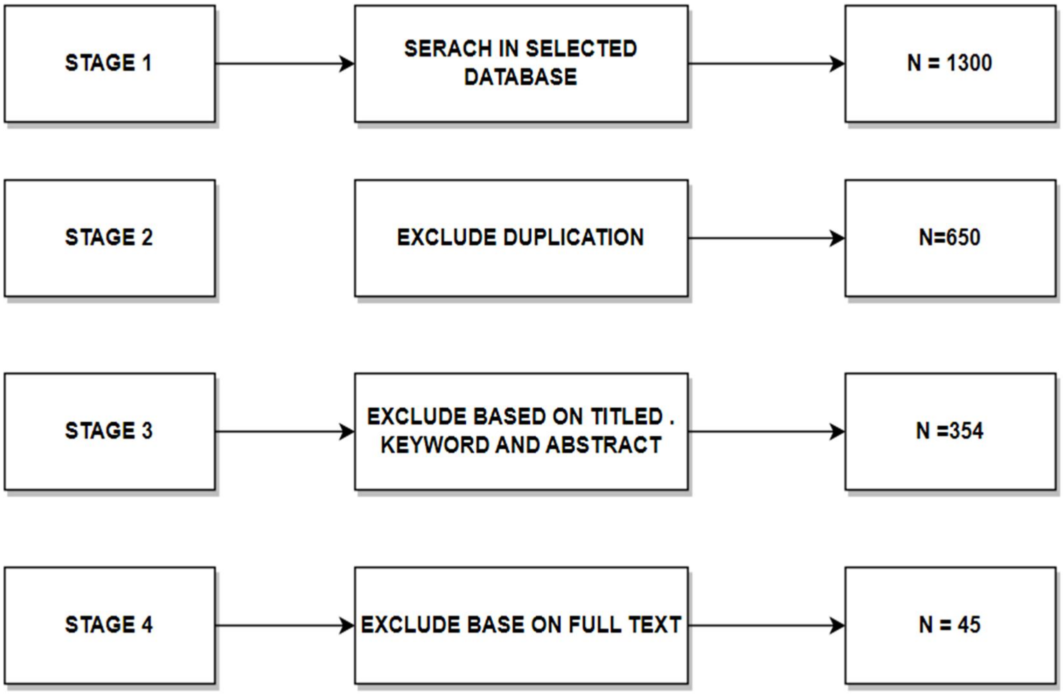
STAGE 1
SERACH IN SELECTED DATABASE
N = 1300
STAGE 2
EXCLUDE DUPLICATION
N=650
STAGE 3
EXCLUDE BASED ON TITLED . KEYWORD AND ABSTRACT
N =354
STAGE 4
EXCLUDE BASE ON FULL TEXT
N = 45

Fig 2. Diagram showing stages of inclusion and exclusion.

From the figure above, a total of 1300 related studies were review in the first stage, out of the 1330 reviewed studies, 650 were excluded as a result of duplication. In stage three studies were excluded due to irrelevancy in keywords, title and abstract, leaving only 354 as relevant to the review paper in question. In the last stage (stage 4) only 45 papers were considered as the one that have the basic requirement for the review paper.

Afters the final selection procedure, the following questions are used in evaluating the papers for the purpose of the research.

1) Is the implementation process of secure software adequately discussed?
2) Are the constraints that affect implementation of secure software properly discussed?
3) Are related issues affecting secure software development practice properly addressed?
4) Are real life scenarios affecting secure software development and implementation addressed properly?

IV. RESULTS INTERPRETATION

After evaluating the 45 papers selected, Table 1 describes each paper in respect to stage it belongs. It is indicated in the table that 40% belongs to IEEE, 22.22% belongs to Springer and 13.33% to Scopus. At the initial stage, IEEE and Scopus were in active but after exclusion of duplicated papers and others papers whose significant are less in term of the review paper, the became more active and contribute tremendously to the findings of the research.

Table 1: Paper distribution process

SOURCES	STAGE 1	STAGE 2	STAGE 3	STAGE 4	PERCENTAGE
ACM	128	100	57	5	11.11%
IEEE	544	211	160	18	40%
Science Direct	141	111	43	2	4.44%
Scopus	247	105	35	6	13.33%
Springer	131	101	50	10	22.22%
Taylor & Francis	56	8	6	3	6.70%
Wiley online	53	14	3	1	2.20%
Total	1300	650	354	45	100%

A. *Research Findings*

Based on the review of the studies carried out it was concluded the following are constraint that hinders the successful implementation of secure software development process.

1) Inadequate support for automated tools
2) Lack of adequate time for software development
3) Inadequate budget planning and cost of implementation of secure software
4) Poor motivation of developers and other team members by their employees
5) In experience of lack of sound knowledge on security of software by project managers
6) Lack of clear and precise requirements and statement by stakeholders toward secure software implementation process.
7) Lack of standard methodology toward implementation and development of secure software.
8) Poor policies toward security implementation in software development
9) Lack of mutual understanding between stakeholders and developer in software security related issues.
10) Lacks of maximum support by the top management toward strengthening of security in software during implementation process.
11) Inadequate training of staffs toward security enhancement from time to time.
12) Lack of security expert's involvement toward provision of secure software development process.
13) I don't care attitude of developers toward ignoring some security measures during the implementation process, serious affect the implementation of secured software.

According to [23] [24] [25] [26] [27], by overcoming some the constraints highlighted above, a much improved implementation of secured software development process can be attained by organization. Thus reducing the changes of security attacks or vulnerabilities in the system or software.

V. CONCLUSION

The main focus of the paper is to identify the constraints that hinders successful implementation of secure software development process. Thirteen (13) basics factors were identified and suggested that by overcoming the constraints observed an improved implementation of secured software development process with zero risks and vulnerabilities can be attained.

REFERENCES

[1] W. Enck and L. Williams, 'Top Five Challenges in Software Supply Chain Security: Observations From 30 Industry and Government Organizations', IEEE Secur. Priv., vol. 20, no. 2, pp. 96–100, Mar. 2022, doi: 10.1109/MSEC.2022.3142338.

[2] K. Abhari and S. McGuckin, 'Limiting factors of open innovation organizations: A case of social product development and research agenda', Technovation, vol. 119, p. 102526, Jan. 2023, doi: 10.1016/j.technovation.2022.102526.

[3] B. A. Dapshima, R. Mishra, and P. Tyagi, 'Transformer faults identification via fuzzy logic approach', Indones. J. Electr. Eng. Comput. Sci., vol. 33, no. 3, p. 1327, Mar. 2024, doi: 10.11591/ijeecs.v33.i3.pp1327-1335.

[4] M. F. Arroyabe, C. F. A. Arranz, I. F. de Arroyabe, and J. C. F. de Arroyabe, 'The effect of IT security issues on the implementation of industry 4.0 in SMEs: Barriers and challenges1', Technol. Forecast. Soc. Change, vol. 199, p. 123051, Feb. 2024, doi: 10.1016/j.techfore.2023.123051.

[5] A. Baksi, S. Bhasin, J. Breier, D. Jap, and D. Saha, 'A Survey on Fault Attacks on Symmetric Key Cryptosystems', ACM Comput Surv, vol. 55, no. 4, p. 86:1-86:34, Nov. 2022, doi: 10.1145/3530054.

[6] B. A. Dapshima, R. Mishra, and P. Tyagi, 'Detection of Faults in Power System Transformers Using Fuzzy Logic Approach', in 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Nov. 2023, pp. 93–99. doi: 10.1109/ICTACS59847.2023.10390205.

[7] A. Attaallah, A. Algarni, and R. Ahmad Khan, 'Managing Security-Risks for Improving Security-Durability of Institutional Web-Applications: Design Perspective', Comput. Mater. Contin., vol. 66, no. 2, pp. 1849–1865, 2021, doi: 10.32604/cmc.2020.013854.

[8] Y. C. Essa, S. Chaturvedi, and S. Khurana, 'Improvement of Augmented Reality in Tourism using Deep Learning', in 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), Feb. 2024, pp. 129–134. doi: 10.23919/INDIACom61295.2024.10498521.

[9] B. Dapshima, Y. Essa, and D. Chaturvedi, 'Fault Detection and Protection of Power Transformer Using Fuzzy Logic', Int. J. Res. Appl. Sci. Eng. Technol., vol. 11, pp. 1816–1824, Jan. 2023, doi: 10.22214/ijraset.2023.48748.

[10] E. Iannone, R. Guadagni, F. Ferrucci, A. De Lucia, and F. Palomba, 'The Secret Life of Software Vulnerabilities: A Large-Scale Empirical Study', IEEE Trans. Softw. Eng., vol. 49, no. 1, pp. 44–63, Jan. 2023, doi: 10.1109/TSE.2022.3140868.

[11] C. Feng, B. Liang, Z. Li, W. Liu, and F. Wen, 'Peer-to-Peer Energy Trading Under Network Constraints Based on Generalized Fast Dual Ascent', IEEE Trans. Smart Grid, vol. 14, no. 2, pp. 1441–1453, Mar. 2023, doi: 10.1109/TSG.2022.3162876.

[12] A. Gimba, S. Ahmad, I. Abubakar, F. Francisca, and A. Umar, 'Design and Implementation of Web-Based Patient Management System Using C#', vol. 24, pp. 69–72, Nov. 2023, doi: 10.9790/0661-2404016972.

[13] F. Hou and S. Jansen, 'A systematic literature review on trust in the software ecosystem', Empir. Softw. Eng., vol. 28, no. 1, p. 8, Jan. 2023, doi: 10.1007/s10664-022-10238-y.

[14] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, 'From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy', IEEE Access, vol. 11, pp. 80218–80245, 2023, doi: 10.1109/ACCESS.2023.3300381.

[15] M. K. Hasan, A. A. Habib, Z. Shukur, F. Ibrahim, S. Islam, and M. A. Razzaque, 'Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations', J. Netw. Comput. Appl., vol. 209, p. 103540, Jan. 2023, doi: 10.1016/j.jnca.2022.103540.

[16] N. Nahar, S. Zhou, G. Lewis, and C. Kästner, 'Collaboration challenges in building ML-enabled systems: communication, documentation, engineering, and process', in Proceedings of the 44th International Conference on Software Engineering, in ICSE '22. New York, NY, USA: Association for Computing Machinery, Jul. 2022, pp. 413–425. doi: 10.1145/3510003.3510209.

[17] R. Jabbar et al., 'Blockchain Technology for Intelligent Transportation Systems: A Systematic Literature Review', IEEE Access, vol. 10, pp. 20995–21031, 2022, doi: 10.1109/ACCESS.2022.3149958.

[18] A. Nurwidyantoro et al., 'Human values in software development artefacts: A case study on issue discussions in three Android applications', Inf. Softw. Technol., vol. 141, p. 106731, Jan. 2022, doi: 10.1016/j.infsof.2021.106731.

[19] D. Odera, M. Otieno, and J. E. Ounza, 'Security risks in the software development lifecycle: A review', World J. Adv. Eng. Technol. Sci., vol. 8, no. 2, pp. 230–253, 2023, doi: 10.30574/wjaets.2023.8.2.0101.

[20] R. N. Rajapakse, M. Zahedi, M. A. Babar, and H. Shen, 'Challenges and solutions when adopting DevSecOps: A systematic review', Inf. Softw. Technol., vol. 141, p. 106700, Jan. 2022, doi: 10.1016/j.infsof.2021.106700.

[21] M. Randevik and P. Olson, 'SecArchUnit Extending ArchUnit to support validation of security architectural constraints', 2020, Accessed: Jun. 28, 2024. [Online]. Available: https://hdl.handle.net/20.500.12380/302240

[22] P. M. Rao and B. D. Deebak, 'Security and privacy issues in smart cities/industries: technologies, applications, and challenges', J. Ambient Intell. Humaniz. Comput., vol. 14, no. 8, pp. 10517–10553, Aug. 2023, doi: 10.1007/s12652-022-03707-1.

[23] I. A. Tøndel and D. S. Cruzes, 'Continuous software security through security prioritisation meetings', J. Syst. Softw., vol. 194, p. 111477, Dec. 2022, doi: 10.1016/j.jss.2022.111477.

[24] R. U. Rasool, H. F. Ahmad, W. Rafique, A. Qayyum, and J. Qadir, 'Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML', J. Netw. Comput. Appl., vol. 201, p. 103332, May 2022, doi: 10.1016/j.jnca.2022.103332.

[25] E. Venson, B. Clark, and B. Boehm, 'The effects of required security on software development effort', J. Syst. Softw., vol. 207, p. 111874, Jan. 2024, doi: 10.1016/j.jss.2023.111874.

[26] D. Wermke, N. Wöhler, J. H. Klemmer, M. Fourné, Y. Acar, and S. Fahl, 'Committed to Trust: A Qualitative Study on Security & Trust in Open Source Software Projects', in 2022 IEEE Symposium on Security and Privacy (SP), May 2022, pp. 1880–1896. doi: 10.1109/SP46214.2022.9833686.

[27] W. Umeugo, 'SECURE SOFTWARE DEVELOPMENT LIFECYCLE: A CASE FOR ADOPTION IN SOFTWARE SMES', Int. J. Adv. Res. Comput. Sci., vol. 14, pp. 5–12, Feb. 2023, doi: 10.26483/ijarcs.v14i1.6949.