INTEGRATING SECURITY INTO A SOFTWARE DEVELOPMENT LIFE CYCLE

The students involved in this are: ¶

1 Name: Ehimika Obokhui Imobighe, Matric NO: 28098
2 Name: Ukhurebor Nathaniel Osagie, Matric NO: 28111

Abstract

Today's software is more vulnerable to attacks due to increase in complexity, connectivity and extensibility. Securing software is most often considered as a post development activity and not much importance is given to it during the development of software. However the amount of loss incurred by organizations over the years due to security flaws in software has invited researchers to find out better ways of securing there software. In the research done by many researchers, this thesis presents the importance of software security and how software can be secured considering security in different phases of software development life cycle. A number of security activities have been identified that are needed to build a secure software and it is also shown how these security activities are related with software development activities of the software development lifecycle in today's software, it also present an empirical study of a software company and how security is integrated in their software development life cycle. Keywords: Software, Security, Software Development Life Cycle, SDLC models, Building secure software and model comparison

Introduction

Software security is to engineer software in such a way that the required application functions uninterrupted and is able to nicely handle the security threats during malicious attacks. Software industry has seen some phenomenal growth over the last decade and this growth is continuing with rapid pace. Over the years the software industry has excelled in every field as it is hard to pinpoint any field that does not involve the use of software. Be it business, sports, media, defense, or exploratory missions, the need of software support cannot be turned down. Software is directly or indirectly involved in storing, controlling, communicating, presenting, computing and even generating information. Use of software has revolutionized the world in every aspect. Considering how important software is to every aspect of life, it is of paramount importance to build and develop a more secured software. It is seen today that software controls a large number of the financial sectors, databases, communications industries and even deadly missile programs. This is but, to mention a few areas largely controlled by software. Misuse of software or usage of unsecured software in any of the listed areas can lead to heavy economic loss in the financial sector and communication sabotage in the communication sector, critical data theft in the database and the worst of it all could be the misuse of missile controlling system that could endanger human life. That is why [1] believes that Whenever developers are going to develop a software using SDLC they should put the security of each and every phase of SDLC into consideration otherwise it will reach a big problem which will be harmful to the users and the organizations. In common practice, security is unnoticed in early phases of software life cycle (SLC).[2]. A good software engineering approach is to think about security right from beginning of SLC. According to [1], Secure Software Development requires a mutual commitment between application developers & information security team members. Both parties, which is the application developers & information security team members must be committed to the integration of security from the beginning & should agree that security concepts will be addressed at each stage of the SDLC. The roles and functionalities, that software performs, and their nature makes them vulnerable to attacks. Software is growing in terms of size, complexity, extensibility, and connectivity [3]. Windows 3.1 contained 3 million line of code while Windows XP contains 40 million lines of code [3]. Java and .NET platforms are extensible in nature, and are designed to accept mobile code update and extensions [3].

Organizations usually consider security as a post development activity. Security is not given consideration during the pre-development and development phases. Organizations do not realize that “Software security is an emergent property of a complete system, not a feature [3]”. After the completion of software development, organizations try to incorporate security as a patch [3]. Furthermore organizations spend lot of money in purchasing good firewalls and antivirus programs and consider that this outer shield is enough for making software secure. However their current approach is not working and organizations continue to incur heavily losses due to exploitation of security flaws.

It is clearly seen from the above discussion that a post development methods of securing software is not a good approach in making a software free from attack, hence there is a need in finding better ways in securing our software in the SDLC.

2.1 Purpose of statement

The purpose of this paper is to research on better ways to develop a more secured software. This will also lead us to finding out how security can be incorporated in the software development process making security a key factor in each phase of the development process in the SDLC.

2.2 Intended Audience

The research focuses on the industry and academia. It will be helpful to the industry in the area of software development. Software development organizations can follow the proposed security development activities in secure software development life-cycle for building secure software. In the academia this thesis can be used to understand software security, SDLC methodologies, how and when they are been used and also to understand how security activities are related with the software development activities in the software development life cycle

2.3 Research Methodology

An extensive and detailed literature survey is needed to be carried out to achieve the goal of this research. The literature survey is based upon research on articles from authentic and reliable sources like IEEE and ACM, proceedings of conferences, books and authentic web sources including of practical experience of a software development company.

3.0 WHAT IS SDLC

SDLC is a procedure or process followed before a software is developed, within a software organization. It outline a detailed plan describing how to develop, maintain, replace or enhance specific software. The life cycle of a software defines a methodology for improving the quality of software and the overall development process.

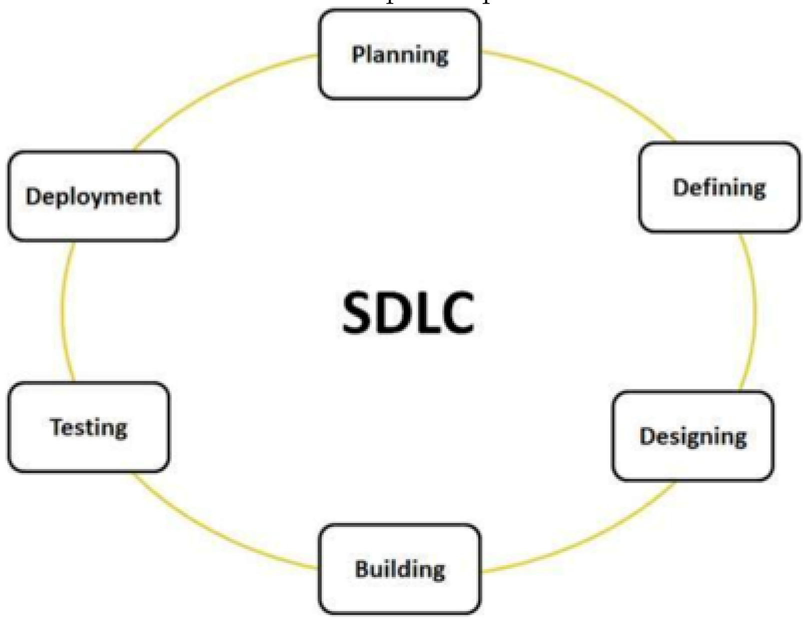


figure 1: link

3.1 Security by Planning

According to [4], Planning usually happens after there is an innovation or initiation that comes up from a group or a business end-user or a sponsor whom identify a need or an opportunity. Within the planning stage, scope or boundary of concept are defined. Product feasibility study in financial, operational and technical areas will be conducted by the senior members of the team with the input from the business users. Quality Assurance and Risk management plan are also prepared at planning stage to minimize any unpredictable risks. Because Case Documentation (BSD) should be ready at this stage to summarize all the ideas and have holistic view of the full plan.

3.2 Security by Defining Requirement

The goal of the requirements phase is to identify software requirements that fulfill the client's demands and represent them in a way that is both clear and understandable. Along with the identification of software requirements, efforts are made to identify security requirements. Security requirements come from different sources and at different times. Identification of security requirements is more difficult than the identification of functional requirements because they are not prominent like functional requirements. For any major project, requirements engineering has always been critical for its success. Security requirements may fall into three main categories [5] these are: i) Functional (Behavioral) security Requirements. ii) Non Functional security Requirements. iii) Derived security Requirements..

3.3 Security by Design

It is the most creative phase in Software Development Life Cycle. The goal of this phase is to transform the requirement specification into a structure or plan. It is the process of planning and problem solving for a software solution. It implicates software developers and designers to define the plan for a solution. The output of this phase is Software Design Document (SDD) [6]. The design phase serves as an aqueduct between the requirement and the implementation phase hence this stage needs to be completed and thoroughly checked as it is the starting point of a software implementation.

3.4 Security by Building

After the best or the most appropriate design has been selected, implementation starts immediately. Programmers should develop the software according to the DDS and at the same time follow the coding standards define by the company's closely. In this phase of a software development, Software Design Document (SDD) is converted into code by using some programming language [7]. This is one of the most delicate phase of an SDLC as it requires a lot of expertise to perform a well secure code. This phase of an sdlc brings its own flaws that are only born with codes and are impossible to detect in earlier phases.

3.5 Security by Testing

According to [4] software testing should be conducted at all stages as a sub-stage. While [6] believes that just after coding phase, testing is carried out to know the outcome of application. Testing is made to know the actual result and the expected result. This is most important and powerful phase. Effective testing will provide high quality software products, lower maintenance costs, and more accurate and reliable results [6].

3.6 Security by Deployment

When the software reaches the Release phase, it is assumed that most of the security flaws have been identified in preceding phases and the software is stable and is ready for deployment. But before deployment, the software is subjected to security review and audit. The purpose of this security review and audit is to identify any remaining security flaws and gain confidence over the software.

4.0 SDLC METHODOLOGIES

The process of building computer software and information systems has always been dictated by different development methodologies. A software development methodology refers to the framework that is used to plan, manage, and control the process of developing an information system. [6].There are several methodologies used in the process of developing a software, these method sometimes depend on the company. In this paper we will talk on the 3 most used methods.

4.1 Agile Software Development Method

The agile process follows the software development life cycle which includes requirements gathering, analysis, design, coding, testing and delivers partially implemented software and waits for the customer feedback. In the whole process, customer satisfaction is at highest priority with faster development time.[8]. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders.

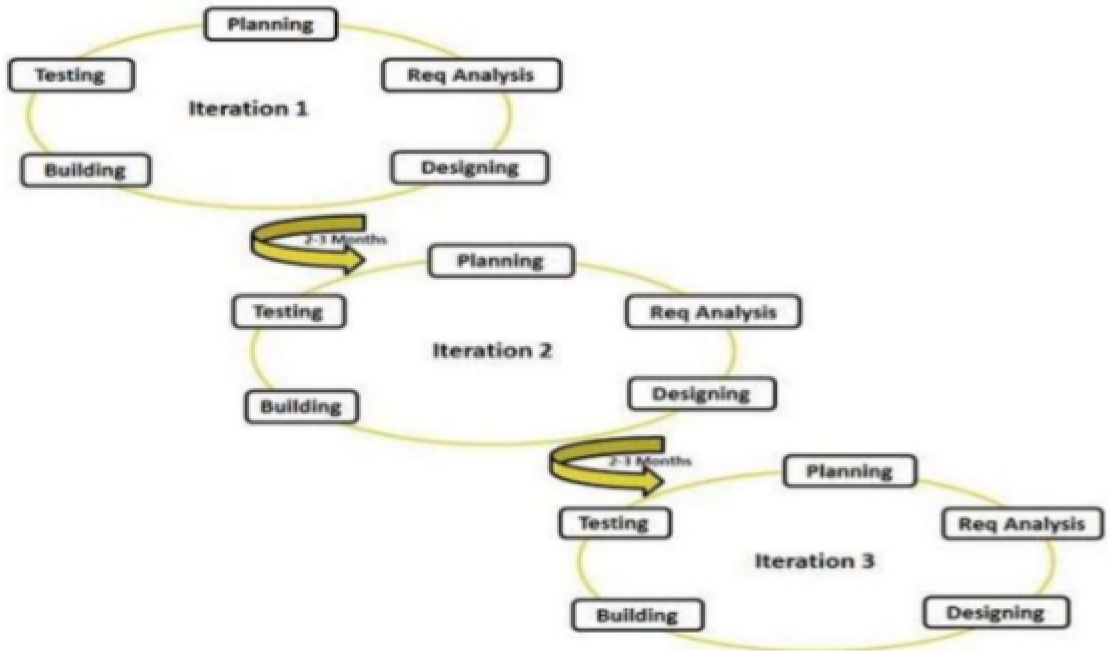
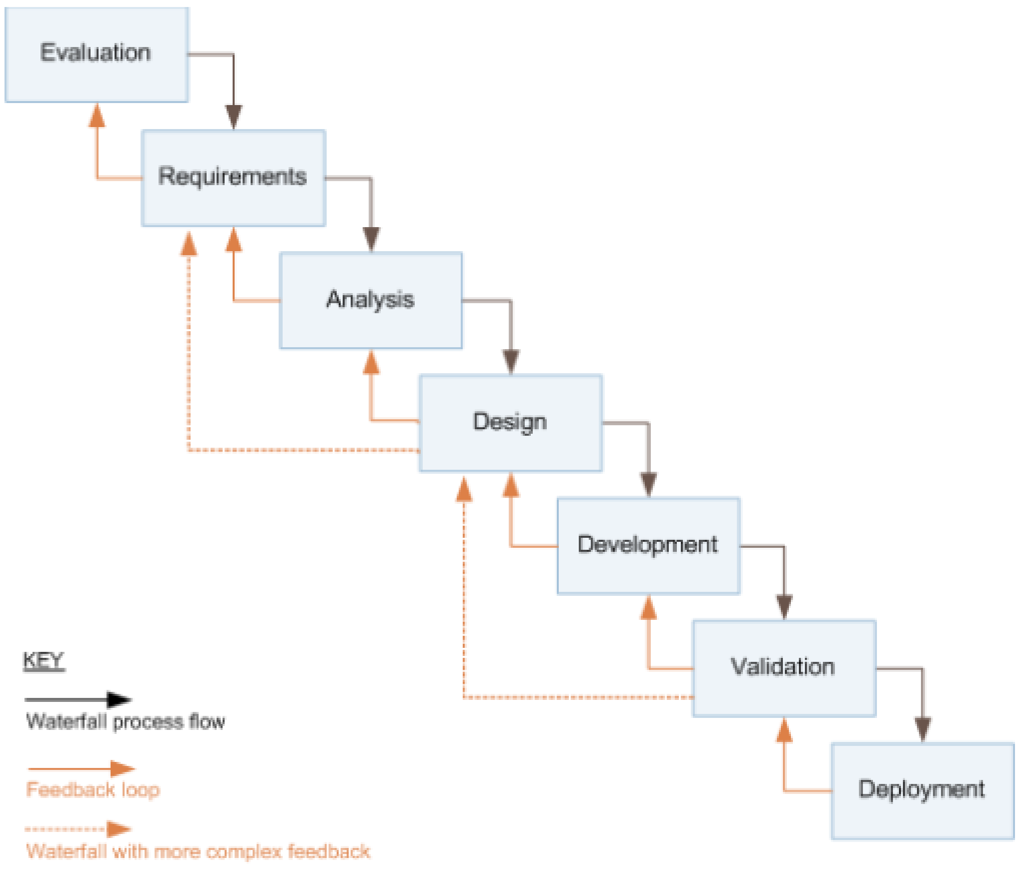


Figure 2.

Illustration of an agile method: Illustration of an agile method

4.2 Waterfall Method

The waterfall model, also known as the cascade model, was first documented by Benington [9] in 1956 and modified by Winston Royce [10] in 1970. It has underpinned all other models since it created a firm foundation for requirements to be defined and analyzed prior to any design or development. The original cascade model of Bennington recommended that software can be developed in stages: operational analysis – operational specification – design and coding specifications – development – testing – deployment – evaluation. . By recognizing that there could be unforeseen design difficulties when a baseline is created at the end of each stage, Royce enhanced this model by providing a feedback loop so that each preceding stage could be revisited.



Waterfall Model explained Figure 3

Quality assurance is built into the waterfall model by splitting each stage in two parts: one part performs the work as the stage's name suggests and the other validates or verifies it. For instance, the design stage incorporates verification (to assess whether it is fit for purpose), the development stage has unit and integration testing, and the validation stage contains system testing as its part and parcel.

4.3 Iterative Model

Security itself is a complete life cycle of software development. Malik Imran Daud [11] believes that iterative method is considered more efficient and reliable approach for software development. In Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. Blend of security and XP gives a new approach that is shown in the figure below. The figure below shows an iterative model of secure software life cycle (SSLC) based on extreme programming concept

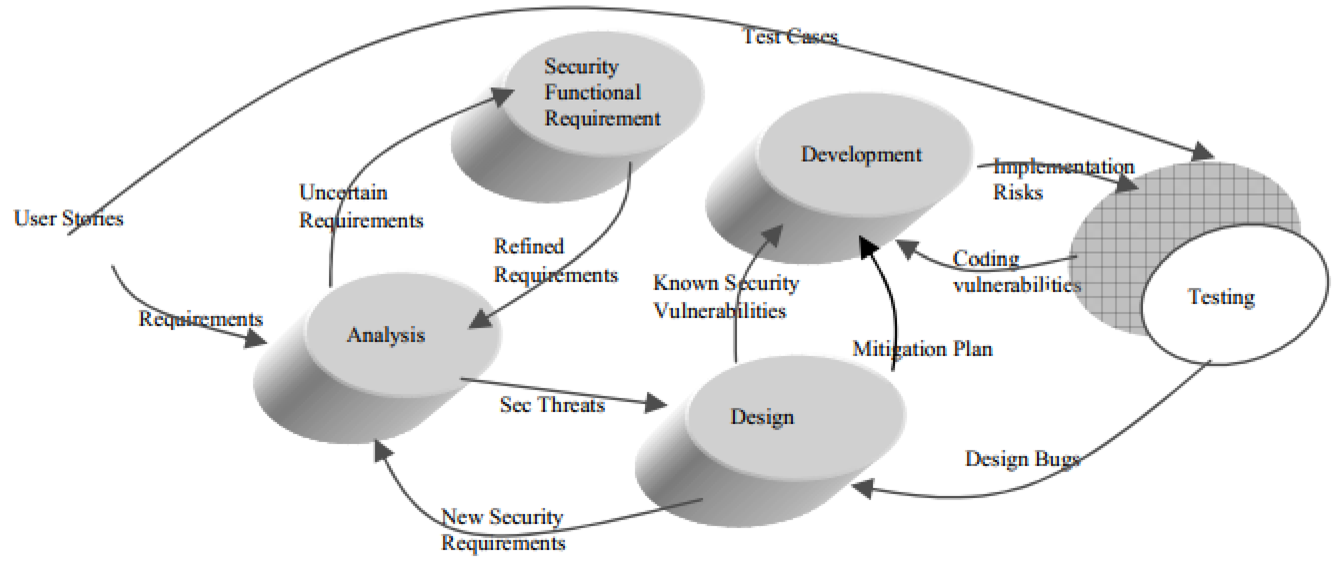


figure 4.iterative model explained

Once the uncertain requirements are refined by SFR module, then we are ready to start designing our software. Design phase is important and requires more consideration in terms of security. Based on the information provided by analysis phase (Security Requirements by user stories and SFR) a threat model is developed. If security engineer feels some of the information is missing or some other security threats are possible then it goes back to analysis for the refinement of the security requirements.

5.0 EXPERIMENT / SURVEY

This section explains a practical experiment / survey that was carried out in a software company in Vietnam to help have an indebt knowledge on how various software companies integrate security into their software and also to know how efficient and reliable these methods can be.

5.1 Company SDLC

Tibco Orchestra Network is a software company that uses the Agile Methodology and Model to build a secured software. Agile development, implement constantly changing functional and technical requirements, which help the company teams to focus on the rapid delivery of business value to her customers. As a result Tibco Orchestra Network is capable of significantly reducing the overall risk associated with software development e.g. Risk of changing functionality after software development has been concluded especially using the "Waterfall method". Tibco Orchestra Network uses the Scrum methodology which is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment.

In an interview with one of the developers on why the company adapt the scrum method, he said "the success of scrum lies in this six characteristics which my company has made an everyday diet"

1. Our product quality is optimized as it is developed

2. Our product is improved and updated frequently and continuously

3. Scrum saves us a lot of money
4. It has increased transparency among our team and company at large
5. Scrum encourages our team
6. We get feedback from our clients more frequently

Tibco orchestra network meetings are held every Monday. This meeting is to ascertain the progress report of the past week and to discuss on how to improve success rate in the new week. The meetings can be at any time of the day depending on the team, usually meeting are usually held in the mornings between the hours of 8:30am to 11:00am. The meetings are divided into various sections depending on the team.Business analysis team, Continuous integration team, Software Architectural Team, Testers Team, Mobile Development Team.

6.0 CONTRIBUTION

Based on some literature study, this thesis identifies the security activities proposed by different researchers, various methods that can be used to integrate security in an SDLC. An empirical study of SDLC of a software company was also carried out to help back up the research knowledge and also to ascertain whether research knowledge can be implemented in a real world scenario.

7.0 CONCLUSION

This research present security activities in relation to software in the software development life cycle. It is aim on developing a secured software that is free from attack. It goes further to show how the security activities are related with software development activities in the software development lifecycle

Form literature survey, it was clear that the rate of software vulnerability is increasingly growing by the day. This increase led me into finding possible measures on how a well secured software can be developed by software developers.

Bibliography

1. A “Securing the SDLC for dummies” by Jerry Hoff & Mike Chapple & published by Wiley Brand

2. Malik Imran Daud ‘Secure Software Development Model: A Guide for Secure Software Life Cycle’,(march 2010), proceedings of the international MultiConference of Engineering and Computer Scientists 2010 Vol 11, IMECS 201O, Hong Kong

3. G. McGraw, “Managing Software Security Risks”, IEEE Security & Privacy, Volume 2, Issue 2, Mar-Apr 2004, Page(s): 80-83.

4. Yan Ting Wong Tiky, “Software Development Life Cycle”. n.d. retrieved from: https://www.cse.ust.hk/ rossiter/independent_studies_projects/software_development/software_development_

5. Meier, J. D., Mackman, A. And Wastell, B.(2005), “Threat modelling web applications”, Available at: http://msdn.microsoft.com/enus/library/ms 978516.aspx

6. S.K Dora, P. Dubey ‘software development life cycle(SDLC) analytical comparison and survey on the traditional and agile methodology’ national monthly refereed journal of research in science & technology volume no.2, issue no.8issn 2277-117422

7. Goertzel, Karen Mercedes, ‘enhancing the development life cycle to produce a secure software’.(October 2008), DACS Data & Analysis Center for Software, ITT AES, 775 Daedalian Dr., Rome, NY 13441 US

8. Ross, Philip E. ’The Exterminators: A Small British Firm Shows That Software Bugs Aren’t Inevitable.” IEEE Spectrum 42, 9 (September 2005): 36-41

9. Benington, H.D. (1956): Production of large computer programs. In Proceedings, ONR Symposium on Advanced Programming Methods for Digital Computers, June 1956, pp 15-27

10. Royce, Winston W. (1970): Managing the development of large software systems. In Proceedings, IEEE Wescon, August 1970, pp 1-9

11. Malik Imran Daud ‘Secure Software Development Model: A Guide for Secure Software Life Cycle’,(march 2010), proceedings of the international MultiConference of Engineering and Computer Scientists 2010 Vol 11, IMECS 201O, Hong Kong