

A Systematic Review of Software Security Issues Associated With Agile Software Development

Prof. Manisha Tijare¹, Prof. Suman Tanwar², Prof. Seema Patil³

¹(CS and IT, Symbiosis Institute of Technology, Symbiosis International University, Pune, India)

²(CS and IT, Symbiosis Institute of Technology, Symbiosis International University, Pune, India)

³(CS and IT, Symbiosis Institute of Technology, Symbiosis International University, Pune, India)

Abstract: Main aim of this research is to review various software security issues associated with agile software development. Software security issues which are considered are types of design and code changes, lack of documentation, schedule delay and change of requirements through prior literature review. All software security issues associated with agile software development cannot be solved completely but can safeguard and avoid to certain extent. It can be stated that documentation issue can be rectified through giving more attention by highly experienced professional. It can be suggested that scheduled delay and change of requirements can be rectified through proper planning and implementation of the project. Flaws in the design can be rectified by proper communication and discussion among the members in the team. This research adopts only second handed data or secondary data. In future this work can be expanded by collected primary source of data through quantitative approach among various investigators.

Keywords: agile software development, change of requirements, code changes, lack of documentation, schedule delay, software security issues, and types of design

I. Introduction

In developing the software, agile methods have progressed more in commonplace. Agile methods (AM) also adopted in development of network and web applications that is condition where risks of security are well-known characteristics. Other than such risks, most of the prior methods of developing agile have some explicit features which particularly deal with security. As an outcome, security is included afterwards or entailed in the external resources process. There involve alternatives like methods of discrete security for example management standards, checklists and more which can add-on AM, some of these combine into AM in a quality-enhanced, seamless yet cost-efficient way [1]. According to Baca et al [2][3] when considering security, common issue with agile methods is don't have a complete picture of how all needs in the software are deployment which leads to obstacles to seize yet examine threats. Another illustration is the small iterations which differentiate agile projects for development which result in security flaws analysis is not carried out or overlooked in an effective way.

It was pointed out by Baca et al [2] that most of the organizations which develop software, entailing most of the online services and product within Microsoft, adopt the agile development of software and methods for management for building their applications. Security aspect was not given more focus it requires at the time of developing software with AM. AM concentrate on creating features which fulfill consumers' direct requirements and security is requirements of the consumer; it is significant that it should not be overlooked. In the present scenario, highly interrelated globe, there is strong privacy and security needs for protecting private security of data has to be considered as a best priority.

The purpose of the study is to review various software security issues associated with agile software development. Software security issues which are taken into consideration are lack of documentation, types of design and code changes, schedule delay and change of requirements through systematic literature review.

II. Literature Review

2.1 Software Security Issues

2.1.1 Lack of Documentation

A study done by Rubin and Rubin [4] examined the supporting agile software development through active documentation. Agile methods have required less documentation for tasks and the implementation promotion but the important knowledge may be lost during and after the development of the system. Data type, information structure, processing of service, device interaction and the aspects of user interface are some of the system related issues on processing subsystem (PS) apart from the documentation in agile software development. Sharma, Sarkar and Gupta [5] have examined the agile processes and methodology. They have pointed out that large documentation does not save the development time and it has affected the project delivery time. In addition to these, least documentation has saved the development time but it is one of the big

disadvantages for the developer. Writing documentation by less experience professional can create problem which also needs task specialization [6]. Lack of documentation has created a lot of problems in the maintenance of the product in agile software development [7]. It also has created the long term problems for the agile teams. If any old developer leaves from the team, then the system's valuable knowledge will be loss due to the lack of documentation. Lack of prominence has occurred on necessary security documentation and the security designing [8]. Less available information has become difficult for the new software developers and it has created an impact on the performance of the project. Author has pointed out that the detail documentation of implementation is difficult to maintain and it produces the negative outcomes of the project. The main issue of the agile process is that the engineer does not pay enough attention to create the documentation.

2.1.2 Types of Design and Code Changes

Agile methods have not adopted directly for the large and complex projects due to the lack of up front design [9]. A study done by Turk, France and Rumpe [10] examined the assumptions underlying agile software development processes. Designing is like an implicit part of the coding process. Designing and the code changing has created both the positive and negative impact on the performance of the project. Design has existed only in the heads of the developers who developed the product and the loss of these developers has led to the memory loss of an organization which could impair the stability of an organization to complete the projects in the timely manner. Design flaws have created the negative impact on the organizational performance and it also has required high cost to correct the errors. Detailed dependencies have not discovered on the detailed level due to lack of design focus [11]. Agile methodologies have lack of upfront design and developing codes causes higher risk for instance and the team has engaged irrecoverable architectural mistakes due to the lack of appropriate design. Web service secure application design, authorization and the authentication are some of the important research issues related to the system architecture [12]. Analyzing the source code is one of the difficult tasks in the agile software development and it is also responsible for the normal execution of the functionality. Simplifying the design or code is critical process and it has the negative impact on the performance of an organization [13]. If the implementation of design is highly complex, the security vulnerabilities will exist within the system increases. Log forging is one of the conventional security attacks present at the designing level of the software development.

2.1.3 Schedule Delay

Conventional heavy weight has characterized as heavy documentation, codified process and design up front. These are closely related with the schedule delay process in the agile software development. The shortcomings of conventional methods such as the slow adaptation to changing the requirements of the business also have the tendency to behind the schedule and over budget [14]. Delayed scheduled issues and the over-budget are reduced due to adaptability increase in the hybrid method. Accuracy and performance, elicitation technique selection, communication gap, business agility, customer involvement, variations of schedule, poor emphasis in use case modeling, user centric analysis and incompatible interface are some of the issues and challenges related to schedule delay which also has created an impact on the performance of an organization [15]. According to Cho [16], the schedule delay has created the negative impact on the business environments. Schedule delay has failed to provide the dramatic improvements in the productivity, reliability and the simplicity. Weaknesses are the tendency to be behind the schedule. Writing more number of codes with higher costs, increased code complexities etc are increasing the delayed schedule and poor performance [17]. Customer involvement has highly influenced the delay schedule in the software development process. Team's frequent communication and interaction with customer about the project is also related to the scheduled delay. Changing decision of the customer has sometimes created the negative impact on the project outcomes and causes delaying schedule [18].

2.1.4 Change of Requirements

Addressing non functional requirements, poor requirement traceability, immutable requirements, excessive requirements volatility and inadequate requirements managements are some of the issues of requirement engineering (RE) in software development methods. Conflicting viewpoints amongst team, incompatible interface, difficulties in evaluating NFR and lack of standardized RE activities are some of the issues of RE in agile software developments [19]. Ontology based KM practices has proposed to reduce the issues of requirement management in the software development projects. Lack of user input, incomplete requirements, cultural differences, geographical distribution, time zone variation, and lack of communication, lack of co-ordination and understanding and changing requirements are some of the issues related to the security in the software development projects [20]. Requirements inconsistency has created an impact in lot of organizations [21]. Shah and Patel [22] reviewed the requirement engineering issues and challenges in various software development methods. Poor quality of requirement has increased the development and sustainment

cost which also has resulted in the delayed schedule. Change in requirements has faced by both agile members and customers and it is considered as the major issues in the development of software which is of refactoring [23].

III. Discussion

From the above literature, it was clear that common issues faced by user in agile software development are lack of documentation, change of requirements, schedule delay and types of design and code changes.

Table 1: Software Security Issues Associated with Agile Software Development

Security Issues				
Sr. No	Lack of Documentation	Types of Design and Code Changes	Schedule Delay	Change of Requirements
1.	Lack of documentation has created a negative impact on the performance outcomes. [4]	Lack of upfront design has created a negative impact on the complex projects. [9]	Shortcomings of conventional method have created a scheduled delay highly. [14]	Conflicting viewpoints amongst team, incompatible interface etc are some of the issues related to the change of requirements in the agile development process. [19]
2.	Least documentation is the disadvantage for the developer.[5]	Design flaws have created a negative impact on the performance of the organization which requires high cost for correcting the errors.[10]	Technique, performance and the schedule variations are some of the factors which cause schedule delay in the software development.[15]	Lack of user input, incomplete requirements and variation in time zone are some of the requirements issues related to the software development process.[18]
3.	Writing documentation for agile teams by less experienced professional has created the issues in software development process.[6]	Lack of appropriate design and codes has influenced the team members to commit mistakes in higher level.[11]	Schedule delay has failed to provide the improvements in the productivity, reliability and the simplicity.[16]	Requirements validation, engineering issues, elicitation and specification issues are some of the requirements issues related to the agile software development. [21]
4.	Lack of documentation has affected the maintenance of the project.[7]	Web service secure application design is one of the main issues in the design of system architecture. [12]	Increased code complexities have created the delayed schedule and one of the main security issues in the agile software development. [17]	Poor quality of requirement has increased the development and sustainment cost.[22]
5.	Engineers do not pay enough attention to develop documentation and it has created the life cycle problems in the working environment.[8]	Design implementation is complex and it has also produced the security vulnerabilities within the system.[13]	Interaction with consumers and changing decision has caused the delay schedule in project.[20]	Software design and the documents have influenced the requirements change.[23]

Table 1 depicts software security issues associated with agile software development.

Lack of documentation affect the project maintenance, disadvantage for the developer, negative effect on the results of the software performance, documentation written by less experienced professional create the risks in process of developing software. When engineers do not pay proper focus for developing documentation and it has developed the issues in life cycle in working circumstances. Not proper upfront design would create a negative effect on complicated projects. Flaws in the software design also created a negative effect on performance of the firm which needs more amount to rectify the errors. Improper codes and design influenced the members in the team to do mistakes in higher level. Designing an application of secure web service is major problems in designing the architecture of the system. Implementing the design in the software is complicated and it has lead to vulnerabilities within the software system.

Conventional method in developing the software has created a scheduled delay. Performance, schedule and highly technique differences are few factors that lead to schedule delay. It failed to give the enhancements in reliability, simplicity and productivity. Maximized code difficulties would result in scheduled delay. Further it was clear that major security risks in the agile development of the software are changing decision and customers' interaction resulted in delay in project schedule. Miscommunication among the team members, not compatible interface are few risks in terms of changing of the needs in the process of agile development of software. Shortage of client input, not complete needs and difference in time zone are few changes needs related problems with respect to the process of software development. Validation in the requirements, engineering problems and the specification and elicitation problems are some other issues related to change requirements in developing the agile software. In addition to these, it was noted that poor requirement quality has maximized the sustainment cost as well as development cost.

IV. Conclusion

This research have reviewed the software security issues were taken into consideration like lack of documentation, types of design and code changes, schedule delay and change of requirements with specific reference to agile software development. It can be stated that documentation issue can be rectified through giving more attention by highly experienced professional. It can be suggested that scheduled delay and change of requirements can be rectified through proper planning and implementation of the project. Flaws in the design can be rectified by proper communication and discussion among the members in the team. All software security issues associated with agile software development cannot be solved completely but can safeguard and avoid to certain extent. This study makes use of secondary data. Further this work can be extended by collected primary data from developers of agile software development through quantitative approach through survey.

V. References

- [1]. Anderson, P., "Measuring the Value of Static-Analysis Tool Deployments", *Security & Privacy*, Vol. 10, No. 3, pp. 40-47, 2012.
- [2]. Baca, D., Carlsson, B., Petersen, K., and Lundberg, L., "Improving Software Security with Static Automated Code Analysis in an Industry Setting", *Software: Practice and Experience*, Vol. 43, Issue 3, pp. 259- 279, 2013.
- [3]. Baca, D., and Carlsson, B., "Agile Development with Security Engineering Activities", *Proc. of Int. Conf. on Software and Systems Process*, 2011.
- [4]. Rubin E and Rubin H (2010), Supporting Agile Software Development through Active Documentation, *Requirements Eng.*
- [5]. Sharma S, Sarkar D and Gupta D (2012), Agile Processes and Methodologies: A Conceptual Study, *International Journal on Computer Science and Engineering*, 4 (5).
- [6]. Stettina C J, Heijstek W and Faegr T E (2012), Documentation work in Agile Teams: The Role of Documentation Formalism in Achieving a Sustainable Practice, In *Agile Conference (AGILE)*, pp 31-40.
- [7]. Tripathi V and Goyal K A (2014), A Document Driven Approach for Agile Software Development, *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(4).
- [8]. Nazir M (2015), Agile Model of Software Security: Risk Perspective, *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(11).
- [9]. Rasmusson J (2003). Introducing XP into Greenfield projects: Lessons Learned, *IEEE Software*, 20(3), pp 21–28.
- [10]. Turk D, France R and Rumpe B (2005), Assumptions Underlying Agile Software Development Processes, *Journal of Database Management*, 16(4), pp 62-87.
- [11]. Mishra D and Mishra A (2011), Complex Software Project Development: Agile Methods Adoption, *Journal of Software Maintenance and Evolution: Research and Practice*.
- [12]. Shravani D, Varma S P, Rao V K, Rani P B and Kumar U M (2012), Dependable Web Services Security Architecture Development Theoretical and Practical Issues- Spatial Web Services Case Study, *Computer Science and Technology*, pp 79-98.
- [13]. Sani A, Firdaus A, Jeong R S and Ghani I (2013), A Review on Software Development Security Engineering using Dynamic System Method (DSDM), *International Journal of Computer Applications*, 69(25).
- [14]. Boehm B (2002), Get ready for agile methods with care, *IEEE Computer*, 35(1), pp 64-69
- [15]. Donald F (2007), Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them, *Journal of Object Technology*, 6 (1).
- [16]. Cho J (2009), A Hybrid Software Development Method for Large Scale Projects: Rational Unified Process with SCRUM, *Issues in Information Systems*, 10 (2).
- [17]. Uikey N, Suman U and Ramani A. K (2011), A Documented Approach in Agile Software Development, *International Journal of Software Engineering (IJSE)*, Vol. 2, No. 2, pp. 13-22.
- [18]. Kumar A S, and Kumar A T (2011), State of software metrics to forecast the variety of elements in software development process, *PDCTA 2011, Springer CCIS 203*, pp. 561-569.
- [19]. Mishra S and Weistroffer (2008), Issues with Incorporating Regulatory Compliance into Agile Development: A Critical Analysis, Retrieved on March 28, 2016 from http://sais.aisnet.org/2008/1AMishraWeistroffer.pdf&ved=0ahUKEwjIj6Gh5eLLAhUEGI4KHYv0BXo4FBAWCCswBg&usq=AFQjCNHiCmfXENYDxaSUcQsmBSjzbx_1RA
- [20]. Kumar A S and Kumar A T (2011), Study the Impact of Requirements Management Characteristics in Global Software Development Projects: An Ontology Based Approach, *International Journal of Software Engineering and Applications*, 2(4).
- [21]. Iqbal M R and Abbasi W A (2014), Requirement Engineering Process in Agile Software Development: Review, *Research Journal of Computer and Information Technology Sciences*, 2(5), pp 1-15.
- [22]. Shah T and Patel V S (2014), A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods, *International Journal of Computer Applications*, 99 (15).
- [23]. Lai T S (2015), A Maintainability Enhancement Procedure for Reducing Agile Software Development Risk, *International Journal of Software Engineering and Applications*, 6(4).