A Security Testing Framework for Scrum based Projects

Nagy Ramadan Darwish
Department of Information Systems and Technology
Institute of Statistical Studies and Research

Ihab Mohamed Abdelwahab
Department of Information Systems and Technology
Institute of Statistical Studies and Research

ABSTRACT
Agile software development methods are characterized by adapting to changing customer requirements and delivering software products in less time. Scrum is one of the most common agile development methods that are used in large software companies like HP, Yahoo, Google, etc. Scrum achieves advantages in time and cost but they may fail in producing software that has good security properties. The weakness in security properties may due to the lack of clear security standard or framework that can be adopted from the beginning of the project. In addition, several studies mentioned that most security vulnerabilities that were left in software during development processes cause threats and cybercrimes. The paper proposes a Scrum security framework that focuses on testing the security of software in Scrum projects. Moreover, the proposed framework can help the team to enhance the security of the software product, minimize the risk of threats, and reduce the cost of fixing the software bugs.

Keywords
Security Framework, Scrum, Security Threat, Cybercrime, Vulnerabilities, Software Development.

1. INTRODUCTION
Agile methods become the most commonly used by software companies in software development. Agile software development processes were originated primarily to support timely and economic development of high-quality software that meets customer needs at the time of delivery. The advocates of agile methods claim that these methods can be accomplished by using development processes that continuously adapt and adjust to [1]:

- Collective experience and skills of the developers, including experience and skills gained thus far in the development project.
- Changes in software requirements.
- Changes in the development and targeted operating environments.

Actually, agile methods are used to reducing risks of project failure. However, they need to follow several rules related to the agile manifesto, including: less documentation, more interactions among team members, and good communication with customer. The most widely used agile methods are Scrum and the hybrid framework combining scrum and extreme programming. Scrum was developed by Schwaber and Sutherland and is described in the Scrum Guide [2].

Scrum is an agile software development framework that is basically used for iterative and incremental software development as shown in figure (1). The main objective of the Scrum is that customer requirements which can be changed rapidly during software development. The sprint goal gives the development team some flexibility regarding the functionality implemented within the Sprint [3]. Iterations are called sprints and every sprint starts with a sprint planning meeting where the customer reviews and prioritizes requirements. A Scrum sprint ends in a Sprint Review, which is a quality gate for the sprint. It has a very important role in the product security risk management [4].

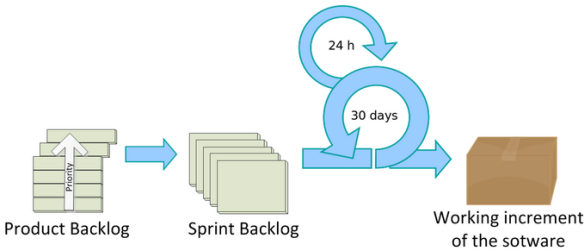

Figure (1): Scrum framework [3]

Generally, security did not take the required priority when developing software using agile methods. Since, Scrum method focuses on rapidly creating features to satisfy customers' direct needs, and security is a customer need, it's important that it not be overlooked [5]. In today's highly interconnected world, where there are strong regulatory and privacy requirements to protect private data. Also, security must be treated as a high priority and examined well. However, some researchers and practitioners highlighted a critical software problem – software security [6]. Security may be considered as non-functional requirement in agile development processes to Scrum developers. Many studies highlighted the existence of software vulnerabilities which eventually lead to threats causing a cybercrime. Also, the conflict appeared to the developers between maintaining a good quality for software that addresses security issues or delivering business functioning code according to the project schedule, and anything else is less important [5].

The remainder of the paper is organized in six sections. Section 2 explores the software security vulnerabilities and threats. In Section 3 presents the software security testing and cybercrimes costs. Section 4 shows the related work of other papers. In Section 5 explores the impact of integrating security on agile methodology. In Section 6 proposes a scrum security framework and describe the framework. Section 7 presents the conclusion of the paper and the research issues that can be focused in future work.

2. SOFTWARE SECURITY VULNERABILITIES AND THREATS
Vulnerability is a weakness in a product that could allow an attacker to compromise the integrity, availability, or confidentiality of that product [7]. One of the direct reasons for vulnerabilities is software complexity which means more functions, more code, hidden code errors. Complexity provides both opportunity and hiding places for attackers and security failures come from it [8]. Vulnerable code appears to be more complex than faulty code [8]. Other factor that may cause software vulnerabilities is related to development phase. When developers produce a new code or inject a code

originally developed for other applications. Incorporate open source code or components from external resources the company might not have much control over their sources of code with insufficient security test and security risk analysis. Consequently, these factors increase the exploits and threats.

[21].

According to U.S National Vulnerability Database (NVD) number of discovered vulnerability in year 2015 was 6488 as shown in figure (2) [9]. The result compared with previous years illustrated that the number of vulnerability is still very high. Although there are a lot of security technologies in different layers for protection. This led to conclude that software security should be a major function in software development. In other words, handling the real cause of the observed threats correctly will save time, money and risk.

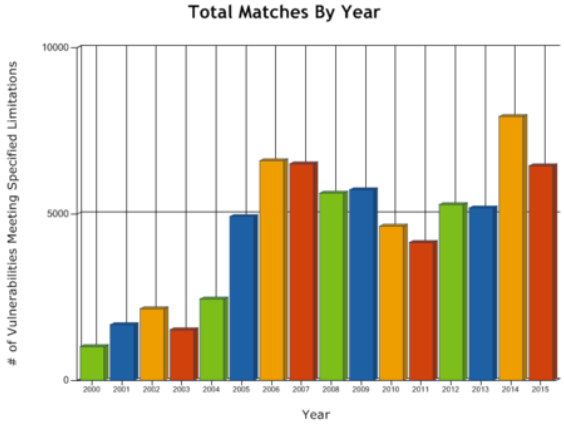

Figure (2): The number of vulnerabilities per year from January 2000 until December 2015 found in NVD [9]

3. SOFTWARE SECURITY TESTING AND CYBERCRIMES COSTS

One of the goals of software testing is to find bugs. The cost of fixing a bug is highly related to where in the process the bug is found as can be seen in figure (3) [10]. Another study prepared by Ponemon Institute US shows that malicious code is the most costly problem for US companies [11]. Countries with the highest costs related to denial of services attacks are the UK and Australia. Malware is most costly in the Russian Federation. In most countries, botnets are the least costly type of attack and the most costly cybercrimes are those caused by malicious insiders, denial of services and web-based attacks as shown in figure (4) [11]. Mitigation of such attacks requires enabling technologies such as applications security testing solutions, SIEM, intrusion prevention systems [11]. The priority should be given to software testing to minimize the cybercrime and Implications.

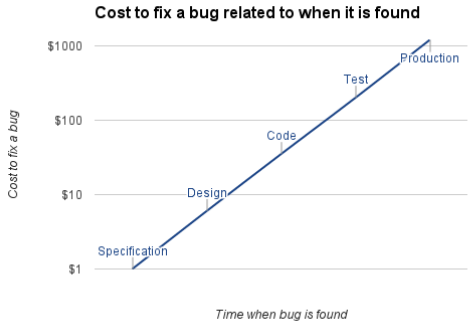

Figure (3): The cost of fixing bugs related to where it is found. This figure is an adapted version of the original from Building Security [10].

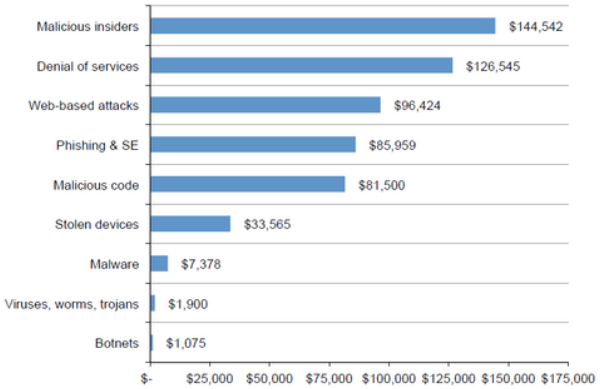

Figure (4): Average annualized cybercrime cost weighted by attack frequency, Consolidated view, n = 252 separate companies [11]

4. RELATED WORK

During the last years several studies have been submitted papers with suggested techniques for enhancing software security within the agile development projects and measuring effectiveness of security activities in agile projects.

- Sonia et al. [12] propose a novel approach which provides quantitative measure of agility for security activities in terms of real agility degree (RAD). It determines the degree of compatibility of a security activity with agile process. Furthermore a comparative analysis had been presented for security activities with each other in context of RAD and risk removal efficiency factor (RREF). RREF is an assessment of how much effective a security activity is for removing the risk and assist a developer during software development in deciding which security activity is beneficial than the other for integration.
- Chowdhury et al. [13] prove with experimental indication that complex, coupled, and non-cohesive software Entities are generally less secure.
- Yonghee et al. [8] investigate if complexity metrics hypothesis is true. The initial results show that the nine complexity measures have weak correlation (ρ=0.30 at best) with security problems for Mozilla JavaScript Engine. The conclusion according to the study results that software complexity is the enemy of software security according to the study results. However,

vulnerable code seems to be more complex than faulty code.

- C.Pohland et al. [14] suggest a Secure Scrum model which is a variation of the Scrum framework with special focus on the development of secure software throughout the whole software development process. The proposed model puts emphasis on implementation of security related issues without the need of changing the underlying Scrum process or influencing team dynamics. Secure Scrum allows even non-security experts to spot security issues, to implement security features, and to verify implementations. Secure Scrum field test shows that the security level of software developed using secure Scrum is higher than the security level of software developed using standard Scrum.
- A.Jøsang et al. [15] provide an agile method for secure software design .The method include a security review in the phase of the sprint iteration cycle where the current version of the system is evaluated. Also, requires team members to have received adequate security education and training.
- S.Jürimäe et al. [16] compare the differences between SDL, CLASP & Touchpoints and which area it cover for software security improvement as it cover only one phase of the development process.
- D. Mougouei et al. [17] offer a security-enhanced version of Scrum i.e. Secure Scrum (S-Scrum) to incorporate security analysis and design activities into the Scrum processes. The proposed methodology has modified the scrum processes to accommodate documentation actives which reflect the security aspects of the target web service. Moreover, the proposed methodology cares for both security and also changing requirements during the release planning, sprints and spikes.
- I.Ghani at el. [18] Succeed in their research to enhanced Scrum model they proposed, evaluation of such model is carried in the requirement, development and testing phases .The results showed that agility is improved if the security backlog (SB) is implemented, which means that agility is not negatively affected if the SB is added to the Scrum model.
- M.Tomanek et al. [2] have also, proposed an agile software development framework Scrum can be enriched by considering the penetration tests and related security requirements during the software development lifecycle. M.Tomanek et al. [2] apply the knowledge and expertise from their previous work focused on development of the new information system penetration tests methodology PETA with focus on using COBIT 4.1 as the framework for management of these tests, and on previous work focused on tailoring the project management framework PRINCE2 with Scrum.
- A.Broström et al. [10] investigate how security can be continuous integrated in an agile development process in source code analysis and automatic security testing. The paper produce a guide for developers telling them how to set up automated security tests, helping them to understand security implications and risks and showing them how they can mitigate certain risks to reduce vulnerabilities.
- Sonia et al. [19] propose an approach that attempt to assist security engineering community in measuring the effectiveness of each security activity separately. The paper use this model concludes that measuring effectiveness of security activity will further assist decision maker by providing guidelines to select security activity which could be more beneficial to integrate with the development process.

The previous studies prove that software security function software security can be enhanced in Scrum. Also, security integrated in an agile development process without negative effect on agile methodology.

5. IMPACT OF INTEGRATING SECURITY ON AGILE METHODOLOGY

Recently some studies discuss the relation between security as a function and agile software developments methodology. It is known to whom caring about software security that may be contradict with one of main features in agile methodology fast delivery, rapid changes and flexibility. Based on figure (5), the degree of agility has improved in the practices from 0.8 (before implementation) to 0.83 (after implementation) [18]. The improvement had been shown after implementation indicates that the security backlog does not add any delay to speed, flexibility, leanness, learning or responsiveness if the security applied as a part of the Scrum method. The pervious implementation is relevant and can be performed without any fear of affecting agility negatively [18]. Also, Sonia et al. [19] proposed an approach that attempt to assist security engineering community in measuring the effectiveness of each security activity separately.

Other studies provide similar papers related to other agile methods such as XP, FDD, DSDM, etc. These studies will encourage the developers to consider the security as a functional requirement without fear from delay of delivery.

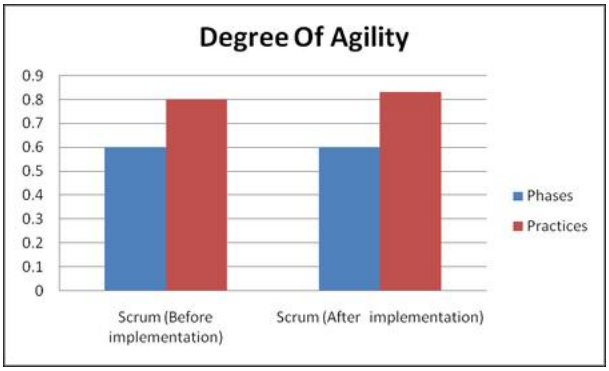

Figure (5): Comparison Degree of Agility [18]

6. PROPOSED FRAMEWORK FOR SCRUM SECURITY PROJECTS

There are some assumptions: Implementing (MS SDL, OWASP CLASP or Cigital's Security Touchpoints…etc'). Testing software security for agile software development project is essential. There is a different proposed security development process practices such as MS SDL, OWASP CLASP, Cigital's Security Touchpoints…etc. These practices are proposed to enhance the security of software products by recommending a set of security activities. However, this will assist the developer to determine which activity they recommend depending on the phase of development. The practical view proves that the security practices (MS SDL, OWASP CLASP) is not enough and still many security bugs appeared in the software maybe because the testing done in an ideal environment (Standalone) without testing it in the business life environment to consider different factors like (Server OS, Client OS, Network Commination…etc'). This

factors required a certain setting to keep the whole environment secured and must be Consider in software development like using a specific ports in network communication device (Firewall , Intrusion prevision system (IDS) , Router …etc') or some security setting in the client side operating system (PCs , Embedded Device …) as a perquisite implementation software. So, without testing the effects of those factors, many threats can be appeared in the whole system.

Testing software within the work environment helps to detect vulnerabilities that may not appear. For example, the malicious user who uses (backdoor or bug) in the software to illegal operations or a malicious code that causes a threat from either outsider or insider user (malicious active content, backdoor).

The proposed framework as shown in figure (6) assumes using one of software security practices for example (OWASP CLASP, Cigital's Security Touchpoints, SANS...). The security framework is used in each sprint and applies security test 1 (static, dynamic tool) before the probable shipped product then integrate it in the work environment. At this stage the security test 2 has been performed either static or dynamic. The proposed framework inspects any security vulnerability or bug appeared within the work environments. Consequently, these elements will be handled and considered again in the sprint backlog task or release a final version of software. Also, it's recommend to apply a security architecture survey to evaluate an application's design and deployment environment (major system components). Based on such survey can be helpful in determining which area pose the greatest security risk, weaknesses in application design enabling you to prioritize the area that needs more in-depth security analysis. Mitigating the risk can be achieved by adjusting the design to cope with security best practices. The proposed security tests are:

- Security test 1 the following points describe the 12 subcategories of application penetration testing methodology you use all or part of this subcategories [20] :
    - Introduction and Objectives.
    - Information Gathering.
    - Configuration and Deployment Management Testing.
    - Identity Management Testing.
    - Authentication Testing.
    - Authorization Testing.
    - Session Management Testing.
    - Input Validation Testing.
    - Error Handling.
    - Cryptography.
    - Business Logic Testing.
    - Client Side Testing.
- Security test 2: Network Pen test (Automatic or manual) discovers the vulnerabilities in the network that can be compromised by attacks and pose threats. The test will cover all or part of the following elements:
    - Operating system.
    - Server application.
    - Intrusion prevention system.
    - Router.
    - Firewall.
    - Network services.
    - Access management.
    - Authentication controls.

After performing the security tests, the results shows a full detailed knowledge about the vulnerabilities and prioritize it according to the business risks impact .Also, propose a clear mitigation strategy to enhance the software security and maintaining the business environment secured.

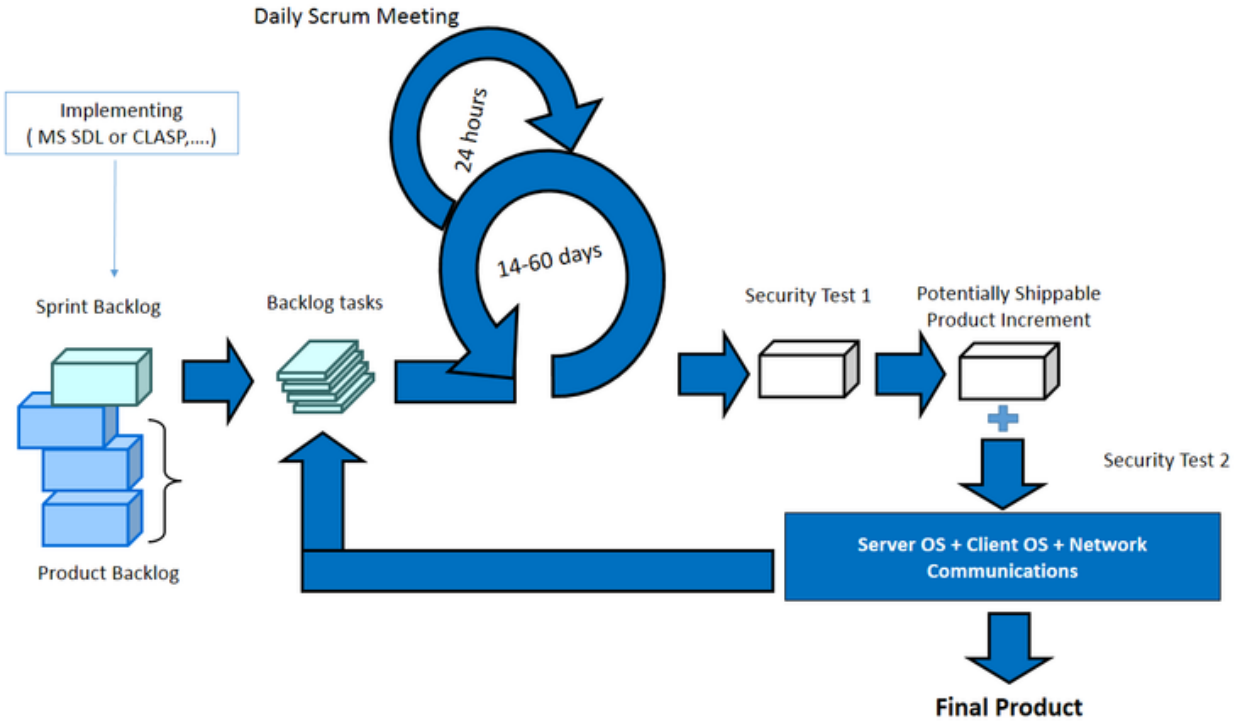

Figure (6): Scrum software development Security Framework

7. CONCLUSION AND FUTURE WORK

This paper reveals that a great attention must be taken to the security issue during Scrum project. The proposed framework provides an effective tool in reviewing and enhancing Scrum projects to produce secure software. The framework explores the security concerns in different stages and perspective considering the software system to avoid any security threat. The proposed framework can help the team to enhance the security of the software product, minimize the risk of threats, and reduce the cost of fixing the software bugs.

In future, a more detailed framework can be targeted to provide the Scrum team with a step by step framework. In addition, a set of security metrics can be used to evaluate the quality of implementing the security of software products in Scrum projects.

8. REFERENCES

[1] D. Turk, R. France and B. Rumpe, "Assumptions Underlying Agile Software-Development Processes", Journal of Database Management, vol. 16, no. 4, pp. 62-87, 2005.

[2] M. Tomanek and T. Klima, "Penetration Testing in Agile Software Development Projects", International Journal on Cryptography and Information Security, vol. 5, no. 1, pp. 01-07, 2015.

[3] K. Schwaber, and J. Sutherland, "The scrum guide, The Definitive Guide to Scrum: The Rules of the Game", (1991st–2013th Ed.). Scrum.org

[4] A. Vaha-Sipila, "Product Security Risk Management in Agile Product Management", Stockholm, Sweden, 2010.

[5] "Agile Security Successful Application Security Testing for Agile Development", white paper, Veracode, Inc, 2010.

[6] I. Ghani and Izzaty Yasin, "Software Security Engineering in Extreme Programming Methodology: A Systematic Literature Review", Sci.Int. (Lahore), 25 (2), P.P. 215-221, 2013.

[7] Microsoft MSDN, "Definition of a Security Vulnerability", 2016. [Online]. Available: https://msdn.microsoft.com/en-us/library/cc751383.aspx. [Accessed: 13- Jan- 2016].

[8] Y. Shin and Laurie Williams, "Is Complexity Really the Enemy of Software Security?", ACM QoP 08, October 27 2008

[9] "NVD - Statistics Results", 2016. [Online]. Available: https://web.nvd.nist.gov/view/vuln/statistics-results?adv_search=true&cves=on&pub_date_start_month=0&pub_date_start_year=2000&pub_date_end_month=11&pub_date_end_year=2015. [Accessed: 13- Jan- 2016].

[10] A. Broström, "Integrating Automated Security Testing in the Agile Development Process", KTH Royal Institute of Technology, Stockholm, Sweden, 2015.

[11] "2015 Cost of Cyber Crime Study: Global", by Ponemon Institute, October 2015.

[12] Sonia and Singhal, "Integration Analysis of Security Activities from the Perspective of Agility", International Conference on Agile and Lean Software Methods, Bengaluru, India, February 17–19 (2012).

[13] I. Chowdhury, M. Zulkernine, "Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities", Journal of Systems Architecture, vol. 57, Issue 3, pp. 294–313, March 2011

[14] C. Pohland, H. Hof, "Secure Scrum: Development of Secure Software with Scrum", arXiv preprint: 1507.02992, 2015.

[15] A. Josang and M. odegaard, E. Oftedal, "Cybersecurity Through Secure Software Development", 9th World Conference on Information Security Education (WISE9), Hamburg, May 2015.

[16] S. Jurimae, "A Literature Survey of the Development Processes for Secure Software", Bachelor's Thesis, Faculty of Mathematics and Computer Science, University of Tartu 2015.

[17] D. Mougouei, N. Fazlida, M. Sani and M. Almasi, "S-Scrum: A Secure Methodology for Agile Development of Web Services", World of Computer Science and Information Technology Journal (WCSIT), ISSN: 2221-0741, Vol. 3, No. 1, PP. 15-19, 2013.

[18] I. Ghani1, Z. Azham and S. Jeong, "Integrating Software Security into Agile-Scrum Method", Ksii Transactions on Internet and Information Systems, vol. 8, no. 2, February 2014.

[19] Sonia and Singhal, "An Evaluation Approach: Measuring Effectiveness of Security Activities", ICDMW 2013, PP. 202–210, 2013.

[20] Owasp.org, "Web Application Penetration Testing - OWASP", 2016. [Online]. Available: https://www.owasp.org/index.php/Web_Application_Penetration_Testing. [Accessed: 19- Jan- 2016].

[21] Cigital, "Third Party Security for Apps & Software", 2016. [Online]. Available: https://www.cigital.com/solutions/by-security-need/third-party-security/. [Accessed: 01- Feb- 2016].