# ROLE-BASED EXTREME PROGRAMMING (*XP*) FOR SECURE SOFTWARE DEVELOPMENT

**Imran Ghani**
(Universiti Teknologi Malaysia, Skudai, Malaysia
imran@utm.my)
**Nor Izzaty / Adila Firdaus**
(Universiti Teknologi Malaysia, Skudai, Malaysia
izatyyasin@gmail.com/adilafirdaus@gmail.com)

***ABSTRACT:*** *Agile methods such as Extreme Programming (XP), Scrum and Feature Driven Development (FDD), are known as efficient development processes because of quick delivery of software even under time and budget constraints. However, certain researches make a point to criticize the XP, Scrum and FDD due to the unavailability of security elements in their various phases and practices. This paper particularly focuses on the limitations of XP, its roles and practices towards developing secure software. Based on our findings, we noticed that software developed using XP method software can be delivered quickly; however the developed software may not be secure. This causes spending more time and budget to repair the software (in terms of security) after the software has been delivered. In this paper, we introduce a new role called "Security Master" and relate certain existing XP practices to it. Based on the initial findings, it has been noticed that the Security Master role helped the XP team to develop secure software during development and after the integration of software.*

**Key words**: Agile methodology; security master; extreme programming; security elements

## 1. INTRODUCTION AND PROBLEM BACKGROUND

Recently, agile approach has become popular among other software development methodologies. Agile methods help quick delivery of software without over-time consuming and over run the budget. There are many agile methods with a little difference in them like example XP, Scrum and Feature Driven Development (FDD). One of the mostly popular methods of software development is XP [2]. XP exploits the reduction in the cost of changing software requirements to by using its twelve practices [3].

Like other agile methodologies that show their effectiveness for quick delivery of software, XP does not guide the security practice as their priority in developing the secure software. This lack in existing XP practices creates vulnerable software, though the software is delivered quickly. This, in fact, causes more time and money waste to repairing the software after the delivery. So, it is really important for XP team to mitigate the risk of threats, which result in potential attacks on a system leading to unwanted outcomes to stakeholders' assets [6].

XP faces the criticism for being inadequate towards the development of secure software, as it does not have a security focused practices or role to guide XP team. These issues have been raised as XP did not have any specific or standards process that the XP team needs to follow [9]. Based on the practices, the team creates and uses the appropriate guide based on the situation that they handle. Unfortunately, XP practices seem loosely coupled in developing the software in structured way.

In our findings, there are a few researches which provide overview on how to develop software more secure using XP model. Like [5], they discuss more on identifying important security threats and dealing with it at early stages. Also, use/misuse cases are carried out from the requirement and design stage, while risk assessment is done in iteration to uncover vulnerabilities that may scale through the initial phase.

By following their opinion and added some ideas into it, we provided the paper that produce an extra role as known as master security and added some security elements into certain practices. Thus, it will to improved agility in XP models. More details about the discussion about these findings will be explained in next section.

## 2. OVERVIEW OF THE ROLES

Based on findings, XP practices are suitable for large-scale, complex software development [11]. But without enough security focuses, these release software will have a lot of problems. It gives more pressure to them when they have to spend a lot expenses in order to maintain and repairing the release software.

On the previous paper, we introduced role-based XP for achieving quality attributes in the process. We also discussed the responsibilities on each role XP and some practices that involve in their activities [12].

In this paper, we focus on identifying the security related practices of XP. In order to establish acceptable level of security within a system, each XP role needs to adopt security focus practices as priorities to reducing the threats or vulnerabilities.

Table 1 highlights which practices are security focused related to XP role which describes how the role can be compatible with security elements among twelve practices in XP.

Based on the Table 1, there are five basic roles in XP, i.e., Managers, coach, customer, developer, and the tester. In addition, we introduce a new role "Security Maser". Each of the roles has his/her own security focus to make sure that the software is developed in secure way. The further detail of each role is as follows:

i. **Customer/Business Owner**

a. *Planning game* : According to the [6], customer to be delivered were written in story cards in detail and prioritized according to their importance. It been responsibility for customer to make sure the process of developing software and the software itself in secure condition. However, not all customers have knowledge on software secure. In this case, researcher from [13] suggests providing the security training for all participants of the project. They must understand basic terminology such as risks, vulnerabilities, assets and notions of risk assessment before the project started.

b. *Small release*: Customer is able to evaluate the product before it been release to determine whether this product have been secure or not[7].

c. *Metaphor* : Customer must be good in explanation in order to give briefing to developer about the whole project details especially to highlight the important function that need security focus.

d. *On-site customer* : Every part of development process, customer will involve to ensure the team follow all the requirement list include the security part. As a domain expert, they will be part of the development team and therefore at the development "site" until the project is finish [8].

ii. **Coach**

a. *Small release* : Coach will manage the quality of product during small release. By improving the security of software/system, the product that been release have the high quality and satisfied the customer's need.

b. *Coding standard* : Synchronize the format of writing code by following the organization standard. Each organization must provide the important standard code of security such as cross-site scripting, SQL injection or weak cryptography to be aware from any threat that try attack the software/system.

iii. **Manager/Tracker**

a. *Planning game* : As a manager, he will manage a meeting to check the progress of development process and the settling the problem as soon as possible in order to avoid from unwanted incidents. It also discuss on how to adapt or improve the security for making a better software.

b. *Small release* : Manager will ensure the team follow the deadline of project within security focus. Small function will release after it is been approval by customer.

c. *Metaphor* : Ensure the whole team understand the whole project with security requirement by providing the effectively architecture during planning game.

d. *Small design* : Monitor the team where they need to provide the simplest design with secure environment that should be functional to minimize the time consuming.

e. *Continuous integration* : Ensure the team integrated the code continuously where there is changes on the code like add a few line for security element or after run the unit test correctly.

f. *Coding standard* : Programmer need to apply the security elements at the same time they writing the code. Coding standard as their references is important during the implementation phases.

iv. **Programmer**

a. *Planning game* : Planning game was the important practice because it always determines the course of the software project [13]. Same as customer role, programmer also needs to have security training. Otherwise, fundamental security architecture also good for them to identified risks and vulnerabilities [13].

b. *Small release* : Goal to delivering an appropriate small release by going through customer satisfied and acceptance test [1].

c. *Metaphor* : Metaphor addressing architecture directly where it shows on how the whole system works [10].

d. *Small design* : Small design via architecture and design models to achieve a stable and simple system structure [4]. Moreover, it easy to see and check whether security elements have been apply in certain function of software.

e. *Refactoring* : Refactoring is a disciplined approach for supporting change in systems [14]. Refer to the [6], refactoring can be expensive in model-based development, however in other way it can be an important practice in agile processes. Base on their case study, they purposed to deliver the security for refactoring that can be achieve through small releases and short iterations by using relevant refactoring techniques [6].

f. *Pair programming* : Pair programming is unique practice that make XP more popular now a days. Indeed, it helps junior programmer to learn those details much more quickly with the right person include security training. Besides, this practice give improvements in the quality of the designs and the coding over what either person could have done on their own. Based on experiences [15], all programmer that experience this more confident and comfortable with making changes to a project. For more secure software, they can take 'security master' as one of their pair programming. Advantages to them in sharing knowledge together especially about security and achieve the security needed.

g. *Collective code ownership* : This may cause 'unsecure workspaces' because pair programming will sit together all the time. However, it not effected to secure software, on the contrary it is a good practices where they can shared the knowledge on how to improve ability of security itself.

h. *Continuous integration* : Teams must keep the system fully integrated at all times. Thus, this practice will prevent error at early stage which might not be detected during testing [7].

i. *40-hours week* :Team have to maintain their productive in developing system by working only 40 hours per week.

j. *Coding standard* : Each organization have their own standard coding include the standard security code. Team need to follow the coding standard so it can be more consistent, understanding and can be edited by other programmer.

v. **Tester**

a. *Testing* : Since Unit testing has been included as an important part of XP methodology of software development, efforts have been made to encourage tester to actually use it in their daily development routine [7]. Tester also needs security training like others to decrease the number of threats in further testing. If all the tests succeeds, a new current version of the project can be release and finally, customer will perform acceptance tests upon it [2].*40-hours week* : Tester need to maintain working 40 hour per weeks to be more focus on searching bugs. So, it can reduce the vulnerabilities of software.

| | Planning Game | Small Releases | Metaphor | Simple Design | Testing | Refactoring | Pair Programming | Collective Code Ownership | Continuous Integration | 40-Hour Week | On-Site Customer | Coding Standards |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SECURITY FOCUS PRACTICE | | | | | | | | | | | |
| Business Owner / Customer | ✔ | ✔ | ✔ | | | | | | | | ✔ | |
| XP Coach | | ✔ | | | | | | | | | | ✔ |
| XP Manager / Tracker | ✔ | ✔ | ✔ | ✔ | | | | | | | | ✔ |
| Programmer | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ |
| Security Master | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | | ✔ |
| Testers | | | | | ✔ | | | | | ✔ | | |

**Table 1 Agile Software Development using Role-based XP with Security Focus Practice**

## 3 Added Special Role in Role-based eXtreme Programming (XP)

Adding some security elements inside software is not effective if there is no professional person in the development of secure software. To make sure the security issues are fixed in proper way, we added a new role called "Security Master" to give advices and lead other roles about security specifications. Here, our research focuses on the activity of Security Master during development.

### A. Security Master

As mentioned earlier that we introduce Security Master as a new role in XP who is expert in security. This role gives a lot of advantages to XP team who wants to develop secure software using XP practices. In XP, the role of security master can be important to provide training to team member, and sharing the information about types of attacks in different types of software. Based on Figure 1, there are ten XP practices which are planning game, metaphor, coding standard, simple design, small release, continuous integration, pair programming, collective code ownership, 40-hours per week and refactoring that are related to security master. This activity is same as programming role in previous paper [12]. The difference between both roles is only this role more focus on writing code for security elements and check whether these security elements is compatible with the software compare to programming where they only write code and function for software during development process

.

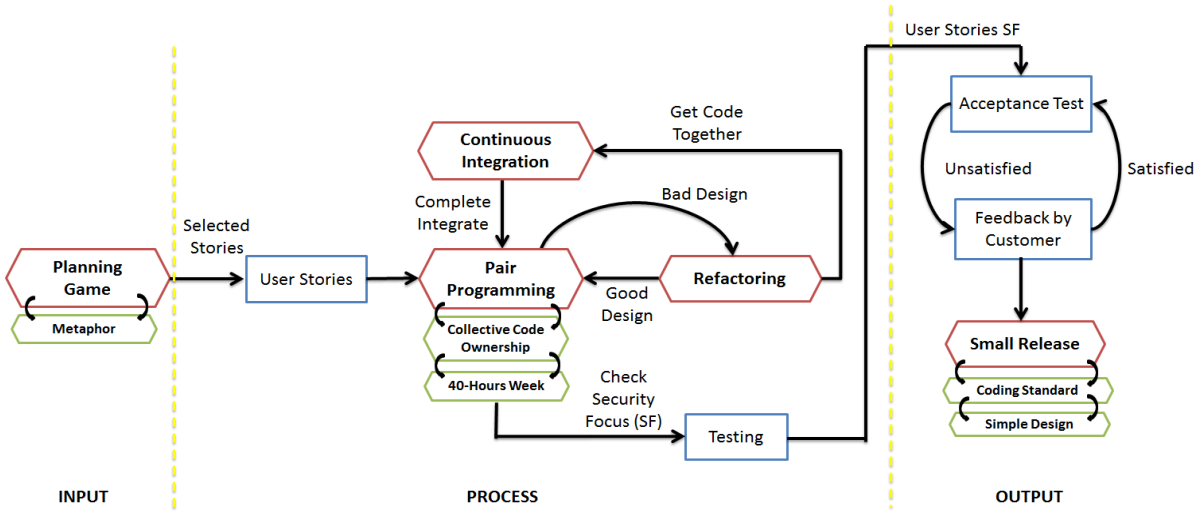


**Fig. 1 Security Master-based XP**

## 4 CONCLUSION AND FUTURE WORKS

In this paper, we have focused on how agile practices can be used to fit-for-purpose of secure software development that meets the security requirements based on XP roles. We also discussed the responsibilities of a new XP role called "Security Master" and some practices that are related to his/her activities. Based on the initial finding, we have found out that the new Security Master role helps in establishing quality of secure software and improving the effectiveness of security related practices during product development. After these improvements, we hope that the industry sector would have no problem in producing their products with security and quality.

## 7 ACKNOWLEDGEMENT

We would like to express our gratitude to Ministry of Science, Technology and Innovation (MOSTI) and UTM for funding this research project under Vot: 4S028.

## REFERENCES

[1] W. Xiaohua, Z. Wu, and M. Zhao, "The Relationship between Developers and Customers in Agile Methodology," *Proceeding from 2011 IEEE Symposium on Computer and Informatics*, pp. 556-572, 2008.

[2] I. Lokpo, M. Babri and G. Padiou, "*Assistance for Supporting XP Test Practices in a Distributed CSCW Environmant,"XP 2004, LNCS*, pp. 262-265, 2004.

[3] K. Beck, "Embracing Change with Extreme Programming," *Proceeding of IEEE*, 1999.

[4] K. Beck and C. Andres, "*Extreme Programming Explained: Embrace Change," Addison-Wesley Professional*, November 2004.

[5] S. Bala Musa, N. M. Narwawi *et al*. "Improved Extreme Programming Methodology with Inbuilt Security,"*International Conference on Computer Science and Information Technology* 2008, pp. 674-679, 2011.

[6] E. G. Aydal, R. F. Paige et al, "*Security Planning and Refactoring in Extreme Programming,"LNCS* 4044, pp. 154-163, 2006.

[7] E. Gamma and K. Beck,"T*est infected : Programmers love writing tests," Java Report*, 3(7), July 1998.

[8] T. Dudziak, "Extreme Programming : An Overview,"Methoden und Werkzeuge der Softwareproduktion WS 1999/2000.

[9] J. Wäyrynen, M. Bodén, and G. Boström, "S*ecurity Engineering and eXtreme rogramming: An Impossible Marriage?,"LNCS* 3134, pp. 117–128, 2004.

[10] R. Jensen, T. Møller, P. Sonder *et al.* "*Architecture and Design in eXtreme Programming; Introducing 'Developer Stories',"LNCS 4044*, pp. 133–142, 2006.

[11] L. Cao, K. Mohan et al, "How Extreme does Extreme Programming Have to be? Adapting XP Practices to Large-scale Projects, " *Proceedings of the 37th Hawaii International Conference on System Sciences*, 2004.

[12] O'reilly, "*Roles in Extreme Programming, Team-Based Software Development*", 2003.

[13] X. Ge, R.F. Paige *et al*, "*Extreme Programming Security Practices,"LNCS 4536,* pp. 226–230, 2007.

[14] Fowler M., Refactoring Home Page. http://www.refactoring.com/, 2005.

[15] D. Karlstrom, "Introducing Extreme Programming - An Experience Report," *Third International Conference on eXtreme Programming and Agile Processes in Software Engineering*, 2002.