Developing Secure Websites Using Feature Driven Development (FDD): A Case Study

Adila Firdaus, Imran Ghani, and Nor Izzaty Mohd Yasin

***Abstract*—Agile processes, like Feature Driven Development (FDD), Scrum and Extreme Programming (XP), have been criticized for not providing a suitable framework for building secure software. In order to find the real-life issues, this case study was initiated to investigate whether the existing FDD can withstand requirements change and software security altogether. The case study was performed in controlled environment – in a course called Application Development—a four credit hours course at UTM. The course began by splitting up the class to seven software development groups and two groups were chosen to implement the existing process of FDD. After students were given an introduction to FDD, they started to adapt the processes to their proposed system. Then students were introduced to the basic concepts on how to make software systems secure. Though, they were still new to security and FDD, however, this study produced a lot of interest among the students. The students seemed to enjoy the challenge of creating secure system using FDD model.**

***Index Terms*—Agile methodology, security, software engineering, curriculum, computer science, feature driven development**

I. Introduction

Agile models have been promising methods towards the software development that run smoothly without overrun the budget and time [1]. However matters like change in requirements, and size [2] of software systems can bring vulnerability to the software. Therefore, recently agile methods such as Scrum [3]-[5] and Extreme Programming (Xp) introduced model that equipped with security[6], [7]. In addition, software security itself has its own model [8]-[13]. However, to develop secure software, it seem a long way to go for FDD since there are only a few research that available about this model.

So far, many researches included agile processes in undergraduate level case studies [14]-[17] in software engineering but the papers mostly cover Scrum and XP. The class only provides opinion on how they could improve those two agile models without implementing and get the professional perspectives. Therefore this case study was conducted in a class and took one semester to see the result of the case study. Two basic goals of this experiment of case student were to introduce students to a variety of software processes in Agile and to get students to adapt security in their system in FDD manners. We wanted to investigate that by adding security elements in the system while applying FDD processes, whether it was possible to handle changes [1].

In our case study, the students who did not have any experience or skills in implementing security element in the system, managed to learn new skill in a semester. Later in this paper, we shall see, how the student were assigned tasks on managing the system development process in addition security elements that have been added as late requirements.

First, this paper provides an overview of the course, in order to provide the context in which this discussion takes place. The paper then turns its attention to how security issues are introduced into the course. The main focus shall be on how the course gets students to think about how they manage when new security requirement when they are in FDD process in developing their system. Specific suggestions that arise during our in-class discussion will be presented

II. Overview of the Course

The course began by introducing a variety of software processes in Agile to students. Then we chose two groups to apply FDD while other 5 groups applied other Agile software process such as Scrum and XP. After that, a two-week introduction to FDD was conducted. Then, the students started the documentation as advised. After that the course continued with implementing the design.

While the students were busy doing the project, the lecturer suddenly changed the requirements during second month. The tasks list was to implement security element like SQL injection, encryption, session management and others that relate towards their system. However each agile group had an assistant to help them in accomplish all these tasks. At the end the students compared the estimated date of completion of the system with the previous date of completion set during the first week of course. They had to check whether they overrun the dateline or not and if they overrun the dateline, what were the reasons that they overboard the dateline.

III. Implementing FDD

As the class project started, the students were divided to seven groups. Two groups were given the task of completing their system project using FDD model and there were guided with all the phases of FDD [17]. The phases are in the next section.

A. Develop an Overall Model Phase

This is the first phase of FDD. In this phase the students were familiarized with this agile model. They needed to

Manuscript received December 9, 2012; revised Febnary 18, 2013.

The authors are with Faculty of Computer Science and Information Systems, Universiti Teknologi, Malaysia (e-mail: adilafirdaus@gmail.com, imran@utm.my, izatyyasin@gmail.com)

DOI: 10.7763/JOCET.2013.V1.73

perform their requirement and made it into a overall model for the whole system where they did using Enterprise Architecture (EA) tool. EA tool helped them create a whole model to depict the whole modules of the system. Both team managed to come out with 5 modules.

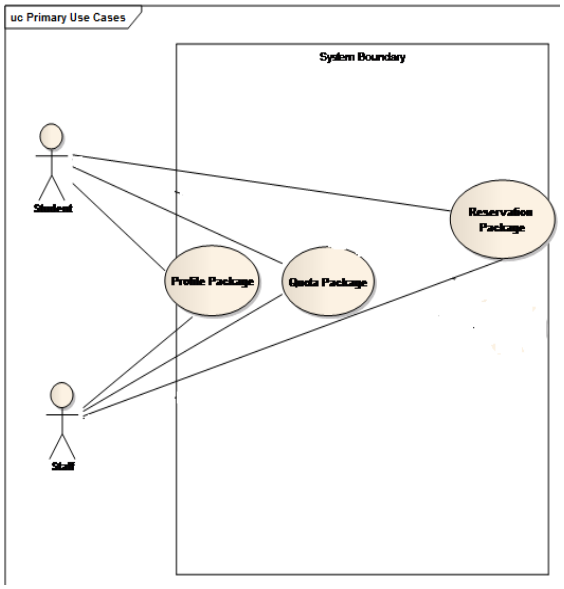

Fig. 1. Residential reservation system for tun razak college overall modules

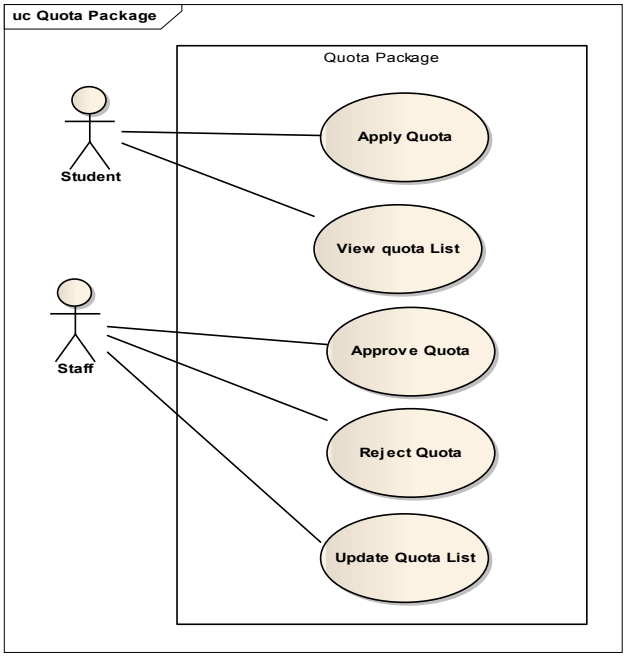

Fig. 2. Example of user profile module features

Based on Fig. 1, this overall modules diagram came from Residential Reservation System for Tun Razak College (KTR RSS) group where they had 3 basics modules that are Profile Package, Quota Package, Reservation Package.

B. Build a Feature List

This is the next phase after finishing the overall model for the system. They need to identify the features that listed under the module. For instance, the system has a module of Quota Package. Therefore the features are Apply Quota, View List, Approve Quota, Reject Quota, Reject Quota and Update Quota List. The task is that one group must have at least 20 features in their project. This phase also have been done using EA tool where each features are noted in use case diagram.

C. Plan by Feature

This third phase is the most crucial phase where all the planning of the project started here. The student must really need to prepare the planning documentation for each module that they have created. Each module must has estimation time to be completed. As far as concern, feature sets with completion dates, Chief Programmers assigned to feature sets, a list of classes and the developers that own them must be listed in their planning documentations. However each team has different kind of planning documentation. One of the group must do the planning using gantt chart where else the other group used mind maps as depicted in Fig. 2. The main reason why this method was used to determine whether by using gantt chart or by using mind map could help to see the overall progress of the development better. As the author of VeriSign said they tried the method of mind maps and they get a better view of the whole system development planning.

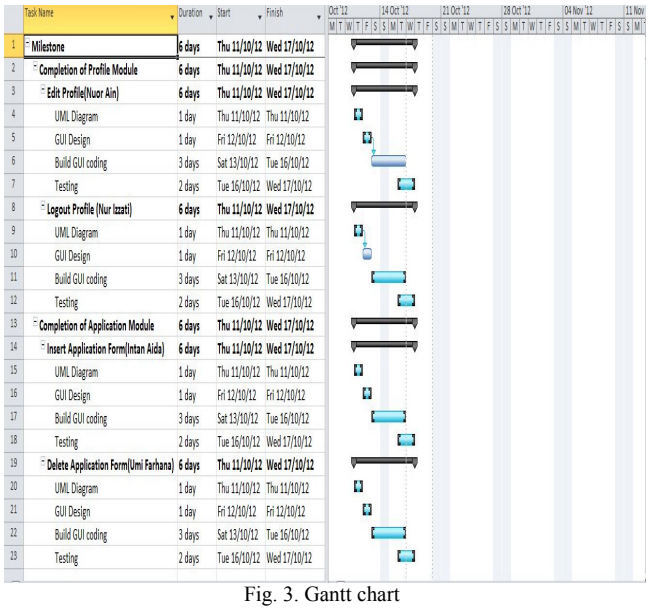

Fig. 3. Gantt chart

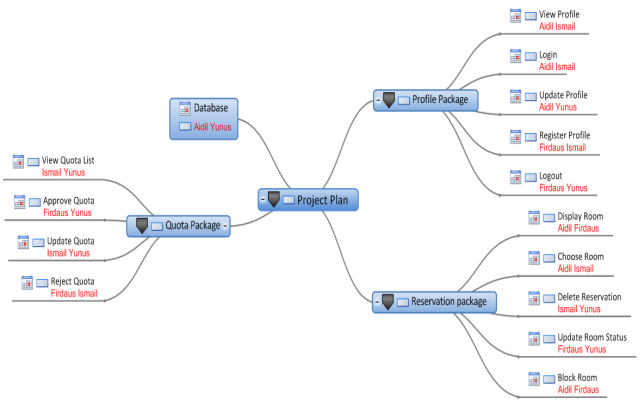

Fig. 4. Mind maps

As we can see Fig. 3 and Fig. 4, each group do different type of planning documentation. Based on the author of VeriSign it is easier to see the whole plan using mind maps. As this example of case study based on Fig. 4, we can see who in charge of each module, feature and the duration needed to complete the module compared to Fig. 3, the gantt

chart only limited to the duration of each feature and the roles are not quite well defined.

D. Design by Feature

This phase required the students to be more precise on how they going to design their system by feature. Each feature must have a sequence diagram and class diagram to picture their system looks like. This phase is also important to show to their customer how the system operates so that any confusion or disagreement could be settled before the developers starting developing the system. Based on Fig. 5, from the Apply Quota feature, the sequence diagram is to show the flow of that feature.

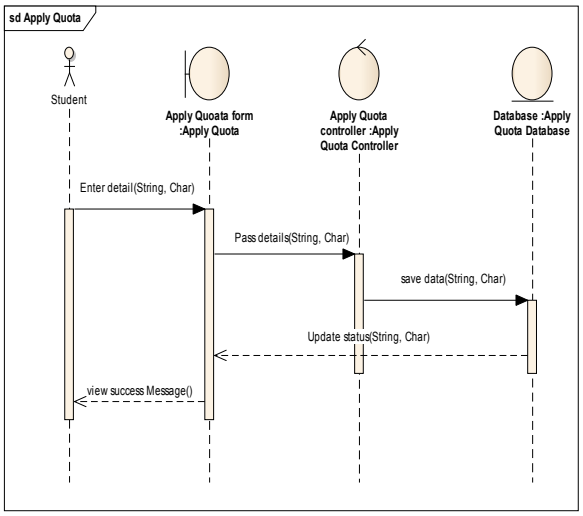

Fig. 5. Apply quota sequence diagram

Fig. 6(a). System interface for apply quota feature(*add quota*)

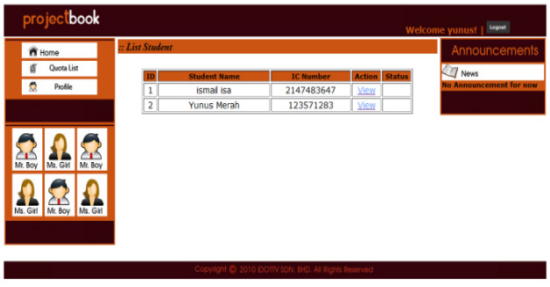

Fig. 6(b). System interface for apply quota feature(*quota list*)

E. Build by Feature

This phase is the last phase where the technical part comes in action. After a thorough check on the design, they started developing the system by module one after another. The lecturer has appointed one to two weeks to complete one module. Even though both teams were using FDD, but they were using different programming language and tools to complete their system. One of the groups was using PHP and the other group was using JSP. While completing the system they will also update their gantt chart or their mind maps. Based on Fig. 6 (a, b), it depicts the interface of apply quota feature interface of the system using php programming language, Dreamweaver software, Mysql database and Xampp server.

IV. The Sudden Security Requirements

This paper was written to see how well the existing FDD model adapting security in their process. As we all know that security does not really match with most Agile modeling and nowadays security have its own model, Software Development Life Cycle [18]. Therefore, while the students in the middle of their developing process, the lecturer gave another task where they need to apply security feature in their system. For instance based on Fig. 7 (a,b), the students need to perform password encryption to secure their login part. Therefore when the hacker tries to get other user password, the only data that the hacker gets is encrypted password.

Fig. 7(a). Password encryption*(login page)*

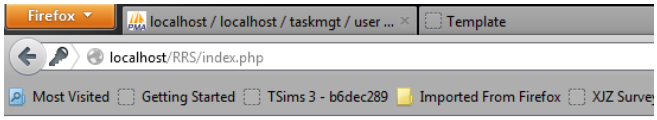

Fig. 7(b). Password encryption(*RESULT PAGE*)

If their system fails to secure it, they need to embed security feature in the log in part. Other than encryption, the students also need to implement session management, sql injection and cross site scripting in their system. After that, we checked if that the estimated completion date is exceeded or remain the same. At last when the system is done, is true that the students have exceeded their estimated completion date. Both of the FDD teams have put 29th of November 2012 as their estimated completion date.

The Online sticker application group has finished their system on 3rd of December 2012 and the KTR RSS application team completed their system on 10th of December 2012. After getting feedback from both teams, they mentioned that the exceeded time is not really due to the addition of security feature but the skills that they have is not sufficient enough and other subjects assignments, test and co –curriculum activities have pull them back in completing their system.

V. Recommendations

There are a few discussions about implementing security in agile models like XP and Scrum [19]-[21].This is the part where the students that have complete their phases in the FDD to voice out their opinion about the weaknesses that they find in the existing FDD model to adapt with security

feature. A set of questionnaire have been distributed to the FDD members to ask them about their opinion throughout the implementation of system. The questions that been asked are as follows:

1) Through your experience in FDD process development, explain briefly weaknesses in FDD modeling.
2) When the lecturer asked you to add security features/elements inside your system, do you think that it slowed down your system development progress? Give reason.
3) Based on your experience throughout the course, suggest ways to improve the whole FDD development process if security features need to be added inside the system.
4) Between gantt chart and mind map, which dou you find better in planning and tracking the progress of the project.

After collecting data from the questionnaire, Table I, Table II, Table III and Table IV show the feedbacks from the students.

TABLE I: FEEDBACKS FROM QUESTION 1

Feedbacks	No. of Students
FDD have less iteration between the phases and stakeholders have the right to add more requirements while the implementation in progress	1
Make sure everything run smoothly; the entire team member must have all the skills.	2
There are not specific roles for each team members	2
There are too much feature that hold by different member that will be very hard to synchronize between the features	2
That is hard to update the documentation until the final stage	3
Lacking in communication between the class owner.	3
Could not handle sudden changes and it will consume time	5
FDD requires too much documentation before the real coding starts	7

Based on the Table I, most of the feedbacks from question 1, stated that the weakness of FDD requires too much documentation before the real coding starts and that will consume so much time to finish up the system. The students also thought that FDD requires too much documentation and when they started the coding, and changes came in between, they need to update the documentation and also developed the code.

TABLE II: FEEDBACKS FROM QUESTION

Feedbacks	No. of Students
No	
The security feature is necessary to be considered as part of the requirements.	2
Yes	
Consume time since they are not prepared the right tool that they are using can be easily create the security features	1
They need time to learn how to implement the security feature inside the system	1
It will slow down since they need time to adapt the changes of the addition on security features	2
Deviate from the original schedule.	3

Most of the students stated that sudden changes in requirements in security features made their development process slowed down due to deviation from the original schedule. However there two students stated that additional security features does not slowed down their progress since they already include the security feature as part of their early requirements.

TABLE III: FEEDBACKS FROM QUESTION 3

Feedbacks	No. of Students
Consider security features as the functional requirement that must be treated fairly like other requirements of the system	1
Lessen the unnecessary documentations	1
Any additional changes on the system especially in security feature must be made parallel with the main system development progress.	1
The team must always report to each other to keep track the progress among each other	2
Must use or specify tools that can be used to create any security features	2
Add security specialist among the roles in FDD	2
Expose the team members with the security knowledge	3
Any requirements that included security features must be clearly stated	4
Must be stated in early stage of the development process	4
The existing FDD modeling must add more iteration between the phases.	5

Majority of the students stated that to improve the existing FDD modeling is by adding more iteration between the phases inside FDD. For instance, the iteration must be repeated since the first phase not in design and build phase only. However just minor of the students proposed that the FDD team must consider security features as the functional requirement that must be treated fairly like other requirements of the system, lessen the unnecessary documentations and any additional changes on the system especially in security feature must be made parallel with the main system development progress.

TABLE IV: FEEDBACKS FROM QUESTION 4

Feedbacks	No. of Students
Mind map is better than gantt chart because gantt chart:	7
Do not show individual progress	
Do not show the detail of the whole project in one shot	
Does not show relation between the task	
Does not show the whole progress of the development	
Hard to understand the progress	
Only focus on duration of the task and dateline	
Need to weekly update the progress	
Only show the progress in parallel	
Gantt chart is better that mind map because mind map:	1
Do not specify the date	
Day and time and have too much information in one short that need time to understand the whole picture.	

Based on Table IV, seven out of eight students agreed that using mind maps is better than use the conventional gantt chart due to the complete view of planning documentation. While only one student think that gantt chart is better than mind map and from his explanation, mind map Do not specify the date and have too much information in one short that need time to understand the whole picture.

VI. Conclusion and Future Works

In this experimental case study toward the secure software engineering, the third year students used FDD model. We have found out that FDD has the potentials to adapt with the secure development life cycle. Based on our observations, we can say that it is possible that FDD could manage well with security of security of software if this model is carefully analyzed and implemented with some of the security practices, plan the right tool, add more iterations between the phases of FDD and use mind map tool, to get a clearer view of the whole progress in the planning phase. We noticed that, even if the students that were still new in developing a secure system and never heard off or familiarized with the secure software development could pull off with little difficulty. However, we noticed that FDD needs to be remodeled to clearly define the process towards secure software development. Sometimes, the current defined roles in FDD are not sufficient. After these improvements, we think that the large scale secure systems (including in the industry sector) would have no problem in developing secured software. Thus, in our future we intend to propose security phase and security master role for FDD.

Acknowledgement

This project is supported by Ministry of Science, Technology and Innovation (MOSTI), Universiti Teknologi Malaysia (UTM), and Research Management Center (RMC), under the Vot Project Number: J.130000.7928.4S026. The project is leading by Dr. Imran Ghani, Senior Lecturer, Software Engineering Department, Faculty of Computer Science & Information Systems, Universiti Teknologi Malasia, Skudai, Johor, Malaysia

References

[1] J. Highsmith, *What Is Agile Software Development*? in Boston: Crosswalk, 2002.
[2] W. Royce, *Improving Software Economics*, in white paper, IBM, 2009.
[3] O. Lau, "The Ten Commandments of Security," *Computers and Security*, vol. 17, pp. 119-123, 1998.
[4] Rakkhis. (2012). [Online]. Available: http://www.rakkhis.com/2011/06/agile-security.html
[5] Z. Azham, I. Ghani, and N. Ithnin, "Security Backlog in Scrum Security Practices," in *Proc. 5th Malaysian Conference in Software Engineering (MySEC),* 2011.
[6] E. Rhoden, "People and processes — The Key Elements to Information Security," *Computer Fraud and Security*, pp. 14-15, 2002.
[7] B. S. Musa, N. M. Norwawi, M. H. Selamat, and K. Y. Sharif, "Improved Extreme Programming," *IEEE Symposium on Computers & Informatics*, 2011.
[8] M. E. M. Spruit and M. Looijen, "IT security in Dutch practice," *Computers and Security*, vol. 15, no. 2, pp. 157-170, 1996.
[9] R. Riley, X. Jiang, and D. Xu, "An Architectural Approach to Preventing Code Injection Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, 2010.
[10] J. Ren, R. Taylor, P. Dourish, and D. Redmiles, "Towards An Architectural Treatment of Software Security: A Connector-Centric Approach Software Engineering for Secure Systems – Building Trustworthy Applications," 2005.
[11] A. Jones, "A framework for the management of information security risk," *BT Technology*, 2007.
[12] V. S. Sharma and K. S. Trivedi, "Quantifying software performance, reliability and security: An architecture-based approach," *The Journal of Systems and Software*, vol. 80, pp. 493–509, 2007.
[13] M. E. Attar, "A framework for improving quality in misuse case models," *Business Process Management Journal*, vol. 18, no. 2, 2012.
[14] R. G. Epstein, "Getting Students to Think About How Agile Processes Can Be Made More Secure," in *21st Conference on Software Engineering Education and Training*, 2008.
[15] W. Neugent, "Teaching Computer Securitv: A Course Outline," *Computers and Security*, vol. 1, pp. 152-163, 1982.
[16] N. Salleh, E. Mendes, and J. Grundy, "Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review," *IEEE Transactions on Software Engineering*, vol. 37, no. 4, pp. 509-525, 2011.
[17] S. R. Palm, Feature-Driven Development—Practices. A Practical Guide to Feature-Driven Development, Prentice Hall PTR**,** 2002, ch. 3, pp. 35-54.
[18] J. H. Allen, "Software Security Engineering: A Guide for Project Manager", in *Addison Wesley Professional*, 2008.
[19] M. Siponena, R. Baskervilleb, and T. Kuivalainena, "Integrating Security into Agile Development Methods," in *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.
[20] R. Shumba, J. Walden, and C. E. Frank, "Teaching Software Security with Threat Modeling," *Journal Science Computer*, pp. 22, 2006.
[21] S. Bartsch, "Practitioners' Perspectives on Security in Agile Development," in *Sixth International Conference on Availability, Reliability and Security*, 2011.

Adila Firdaus Binti Arbain was born in April, 1990, Adila Firdaus was born in the Subang Jaya Medical Centre that known as Sime Darby Medical Centre nowadays. She gets her first education at Sekolah Rendah Subang jaya then she entered her secondary school at Sekolah Menegah Subang Jaya. Then, she got to Matriculation in Perlis for her Pre-U level education. Lastly she continued her degree and Master in software engineering in Universiti Teknologi Malaysia (UTM) in Skudai, Johor, Malaysia. In May 2012 she graduated first class honor in Degree of software engineering and currently, she furthers her Master of software engineering in the same local university. Back then, Adila used to work as an English Mathematics and Science teacher at Sekolah Kebangsaan Pusat Bandar Puchong 2 for three months while waiting results for Pre U level Matriculation application. Afther she finished her matriculation year, in 2008 she worked as the Clerk Assistant at Syarikat Batu Hampar for 6 months before she entered UTM. During her study in UTM, she has to attend Industrial training at Mesiniaga SDN BHD for one semester. Currently, she is doing her first research on security and Agile model especiall in feature driven development. She is planning to develop an Agile Security Model that could help in enhancing and developing a secure system for the future usage. Ms Adila is one of the Software Engineering research group members for her faculty in UTM, SERG. She used to present her work with Prof. Dr. Ina Schieferdecker, from the Technical University Berlin that came for a visit to Utm.

Imran Ghani is a Senior Lecturer at Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Campus. He received his Master of Information Technology Degree from UAAR (Pakistan), M.Sc Computer Science from UTM (Malaysia) and Ph.D. from Kookmin University (South Korea). His research focus includes to study semantics techniques, content-based, collaborative filtering techniques, semantic web services, semantics-based software testing, security in agile software development practices, enterprise architecture and software architecture.