International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 1
ISSN: 2088-5334

Case Study on Non-Functional Requirement Change Impact Traceability for Agile Software Development

Adila Firdaus Arbain[#], Dayang Norhayati Abang Jawawi[*], Wan Mohd Nasir bin Wan Kadir[*], Imran Ghani[+]

[#]*Faculty of Science Computer and Information Technology, Universiti Tun Hussein Onn Malaysia,86400 Parit Raja, Johor, Malaysia*
E-mail: adila@uthm.edu.my

[*]*Faculty of Computing, Universiti Teknologi Malaysia, Jalan Iman, 81310 Skudai, Johor, Malaysia*
E-mail: dayang@utm.my, wnasir@utm.my

[+]*Indiana University of Pennsylvania, 1011 South Drive, Indiana, PA 15705, USA*

***Abstract*— Currently, it is crucial to develop a complex software on time. Agile software development methodologies provide methods to develop a system in term of time and cost-saving but it has been criticized for software quality management. In this paper, a case study is used to find out the need of NFR change impact traceability approach in most of Agile software methodology. This case study was conducted in an undergraduate course that trained the students on how to develop software using Agile process model. This case study has been conducted for 4 months in an undergraduate-level course, Application Development. The samples of this case study are among Year 3 undergraduate students. The case study shows the lack of traceability techniques in the existing Agile process model (SFDD- Secured Feature Driven Development) that result to non-awareness of NFR change impact during development. Based on the case study mentioned the main objective of the case study conducted in survey is to empirically test the theoretical constructs and the hypothesized relationships of the research issues that concern on the lack of change impact management towards NFR in Agile Software Methodology. TANC (Traceability for Agile Non-Functional Requirement Change Impact) model offered techniques in tracing change impact during the agile development process. Therefore, the result of the case study, a traceability process model needs to design in order to tackle the NFR change impact issues in Agile software development.**

***Keywords*— agile methodologies; scrum; feature driven development; traceability; non-functional requirement.**

I. Introduction

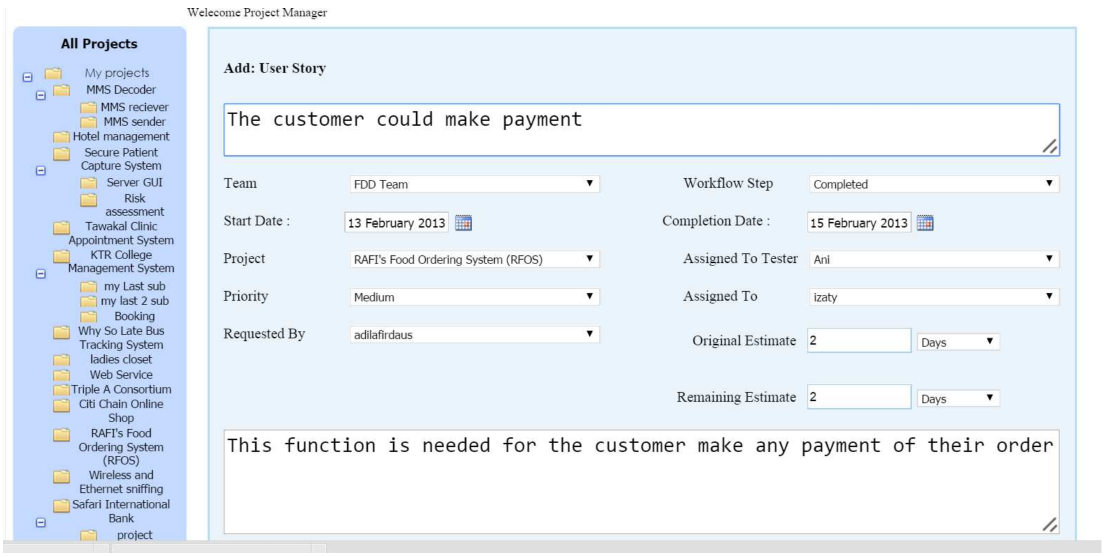
Traceability approaches and methods have been applied in traditional software development process such waterfall [1], [2] model-driven [3] and started to be introduce in Agile software development projects [4]–[6]. As a matter of fact, many researchers have done their research on agile and traceability [7]. There are some researches that have started to create traceability models and techniques in various Agile software development model such as Scrum [8], FDD [9], AUP [10] and other Agile software development model. However, these established traceability techniques in Agile only support the functional requirements, not the NFRs [11]. There are even some researches state that traceability does not compatible with XP processes [12] due to the heavy documentation and architectural solutions, and that do not go well with XP lightweight processes. Therefore, a case study conducted in an undergraduate class where they were using Agile software development method and asked to check the change impact after some requirement changes. By using survey techniques which most of researchers use in order to justify any type of issues that related to Agile methodology [13]–[15] this case study was to testify whether the existing Agile software development method not only equipped with the Agile software development change management [16], [17] but also covers the NFR change impact management. The rest of the paper is organized as follows: Sections 2 presents a case study that experimented on NFR Change Impact Management in the existing Agile Software Development Process Models. Lastly, Section 3 presents the conclusion of this study.

II. Materials and Method

The course begins by introducing a variety of Agile software development models to the students. Then, one of the best students' group project which used SFDD were chosen to be documented in this case study. In the middle of their software development project, they were asked to implement one of the NFR features that is security. They must document the security features using the SAgile tool. Then, they were asked to check any affected functional features or NFR features by asking them to check for the performance of the functional features already embedded with security codes and compare them with the ones without security codes. Then, they were asked if the affected performance is too much for the clients to handle or is barely acceptable. The flow and results of this case study are explained in the next sections. The process details for the case study will be explained based on SFDD phases and generic traceability process model phases order. Lastly, the study continues by collecting and discussing the students' survey feedbacks on the NFR change impact management in the existing Agile process model.

A. Rafi Food Ordering System (RFOS) in Plan by Feature concerning Strategic Trace Phase

Since this team has chosen to use the FDD development process, they are recommended to use SAgile during the Plan by Feature phase. Here are the results of this experiment. First, the students fill out the details about the features. For example, in this case, study, one of the features in the developed system, Manage Payment feature is shown in Fig 1.


Welecome Project Manager

All Projects

My projects
MMS Decoder
MMS reciever
MMS sender
Hotel management
Secure Patient Capture System
Server GUI
Risk assessment
Tawakal Clinic Appointment System
KTR College Management System
my Last sub
my last 2 sub
Booking
Why So Late Bus Tracking System
ladies closet
Web Service
Triple A Consortium
Citi Chain Online Shop
RAFI's Food Ordering System (RFOS)
Wireless and Ethernet sniffing
Safari International Bank
project

Add: User Story

The customer could make payment

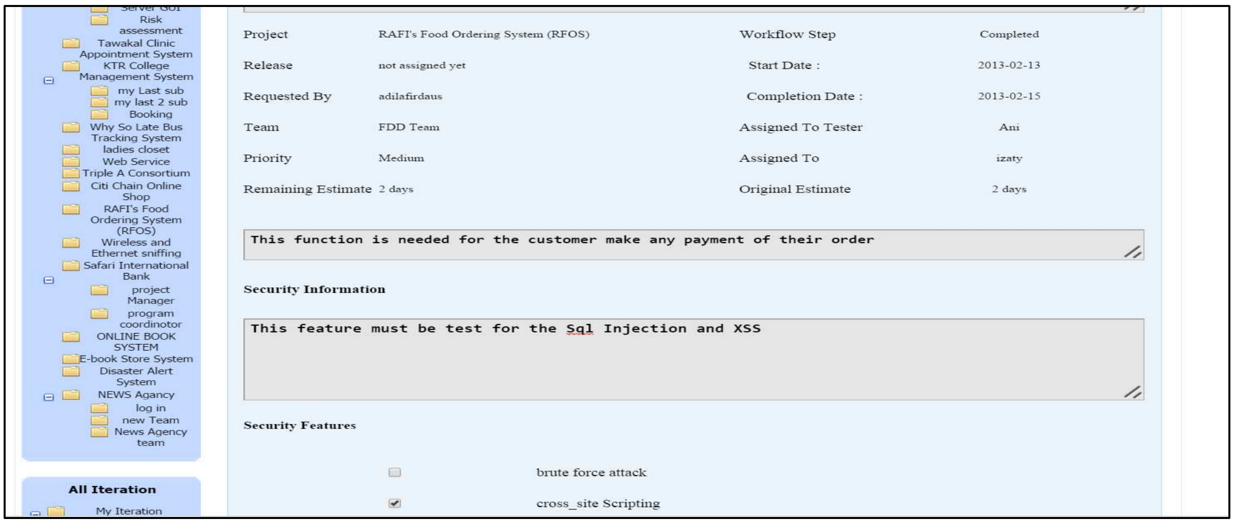
Team	FDD Team	Workflow Step	Completed
Start Date :	13 February 2013	Completion Date :	15 February 2013
Project	RAFI's Food Ordering System (RFOS)	Assigned To Tester	Ani
Priority	Medium	Assigned To	izaty
Requested By	adilafirdaus	Original Estimate	2 Days
		Remaining Estimate	2 Days

This function is needed for the customer make any payment of their order

Fig. 1 A sample line graph using colors which contrast well both on screen and on a black-and-white hardcopy

The figure above demonstrates that all details regarding the Manage Payment feature are present including start and end dates, which developer of this feature is assigned to, who is the tester, etc. This part completes the phase of creating the functional requirements phase in FDD. Next, NFR features are assigned to this feature.

In the first part of assigning NFR, the security elements need to be assigned first (this is based on the Strategic phase plan when the team decided on how they were going to prioritize the rank of the NFR. For example, if the system is security-based, they must check for security elements first or make the security itself as the main feature, then assign other NFR that seems relevant towards the system). The team has assigned a few security elements for this feature.


Risk assessment
Tawakal Clinic Appointment System
KTR College Management System
my Last sub
my last 2 sub
Booking
Why So Late Bus Tracking System
ladies closet
Web Service
Triple A Consortium
Citi Chain Online Shop
RAFI's Food Ordering System (RFOS)
Wireless and Ethernet sniffing
Safari International Bank
project Manager
program coordinator
ONLINE BOOK SYSTEM
E-book Store System
Disaster Alert System
NEWS Agancy
log in
new Team
News Agency team

All Iteration

My Iteration

Project	RAFI's Food Ordering System (RFOS)	Workflow Step	Completed
Release	not assigned yet	Start Date :	2013-02-13
Requested By	adilafirdaus	Completion Date :	2013-02-15
Team	FDD Team	Assigned To Tester	Ani
Priority	Medium	Assigned To	izaty
Remaining Estimate	2 days	Original Estimate	2 days

This function is needed for the customer make any payment of their order

Security Information

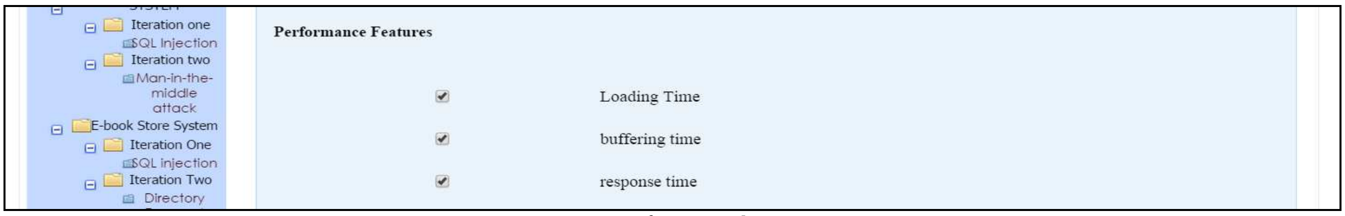
This feature must be test for the Sql Injection and XSS

Security Features

☐ brute force attack
☑ cross_site Scripting

Fig. 2 Security elements

Based on Fig 2, this team has assigned two types of security threats that are Cross-Site Scripting (XSS) and SQL injection. This indicates that this feature must have these two types of mitigation codes embedded in the feature codes for handling the types of security threats stated. After assigning security NFR to the feature, the team assigned performance features to the main feature. However, assigning performance features is a little bit different than the normal assigning process because the performance feature is assigned to detect any impact of the performance feature after the security features have been added. In this case, the team has added loading time, response time, and the buffering time as in Fig 3.

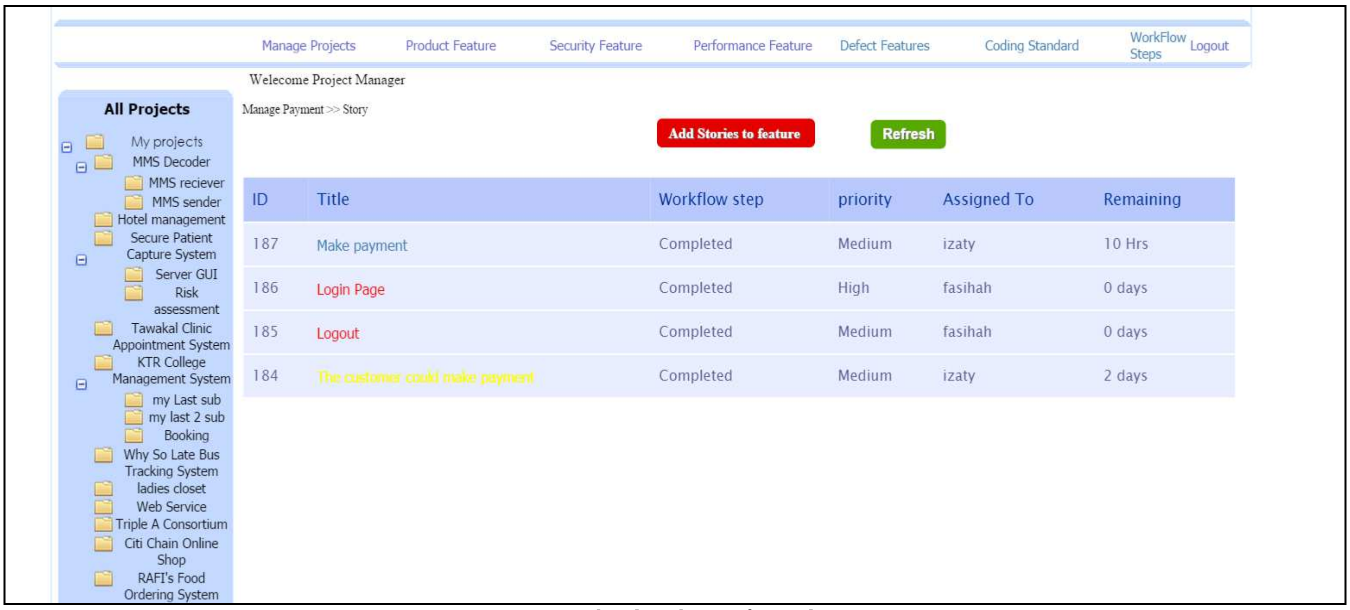


Iteration one
SQL Injection
Iteration two
Man-in-the-middle attack
E-book Store System
Iteration One
SQL injection
Iteration Two
Directory

Performance Features

✓	Loading Time
✓	buffering time
✓	response time

Fig 3: Performance elements



Welecome Project Manager

All Projects

My projects
MMS Decoder
MMS reciever
MMS sender
Hotel management
Secure Patient Capture System
Server GUI
Risk assessment
Tawakal Clinic Appointment System
KTR College Management System
my Last sub
my last 2 sub
Booking
Why So Late Bus Tracking System
ladies closet
Web Service
Triple A Consortium
Citi Chain Online Shop
RAFI's Food Ordering System

Manage Payment >> Story

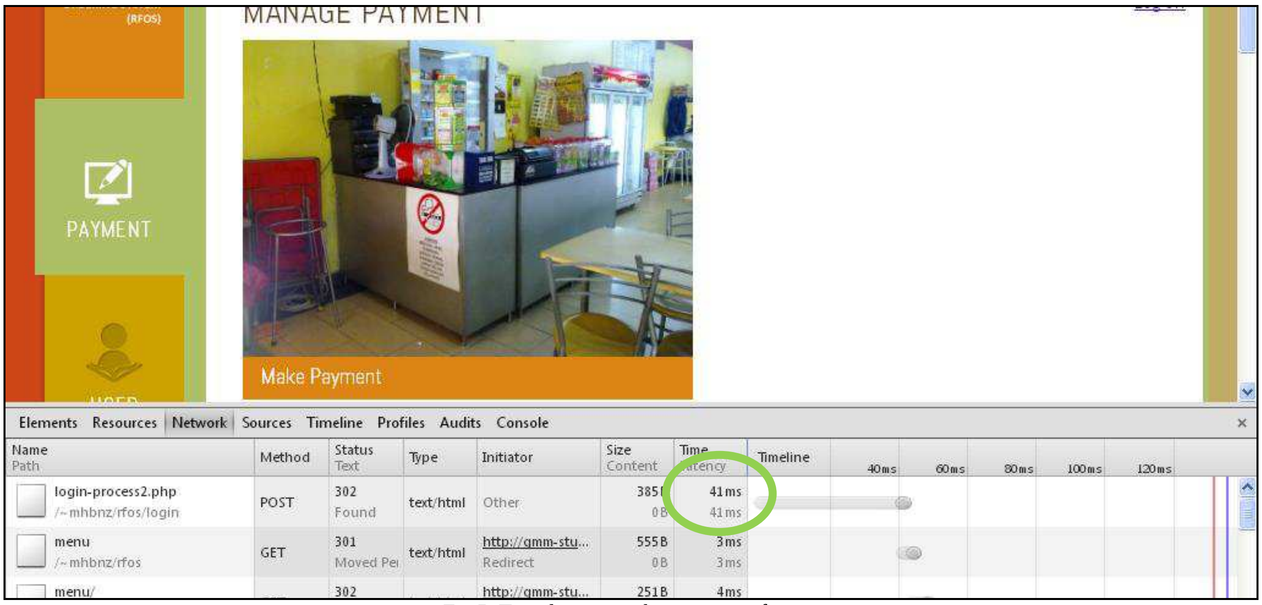
ID	Title	Workflow step	priority	Assigned To	Remaining
187	Make payment	Completed	Medium	izaty	10 Hrs
186	Login Page	Completed	High	fasihah	0 days
185	Logout	Completed	Medium	fasihah	0 days
184	The customer could make payment	Completed	Medium	izaty	2 days

Fig 4: Colored marking in feature list

Due to security and performance features have been assigned simultaneously, SAgile marked these two features in yellow. If the feature is marked in red, then it means that the feature is only assigned to security features and if it is blue, it means that the feature has not been assigned to any NFR feature. Fig 4 shows these colored markings. This marking is a very important feature of SAgile tool in order to notify the development team in making sure that they have assigned adequate NFR features to each feature. Without this marking, it is difficult for the team members to be aware of which features that have not been assigned with which NFR feature.

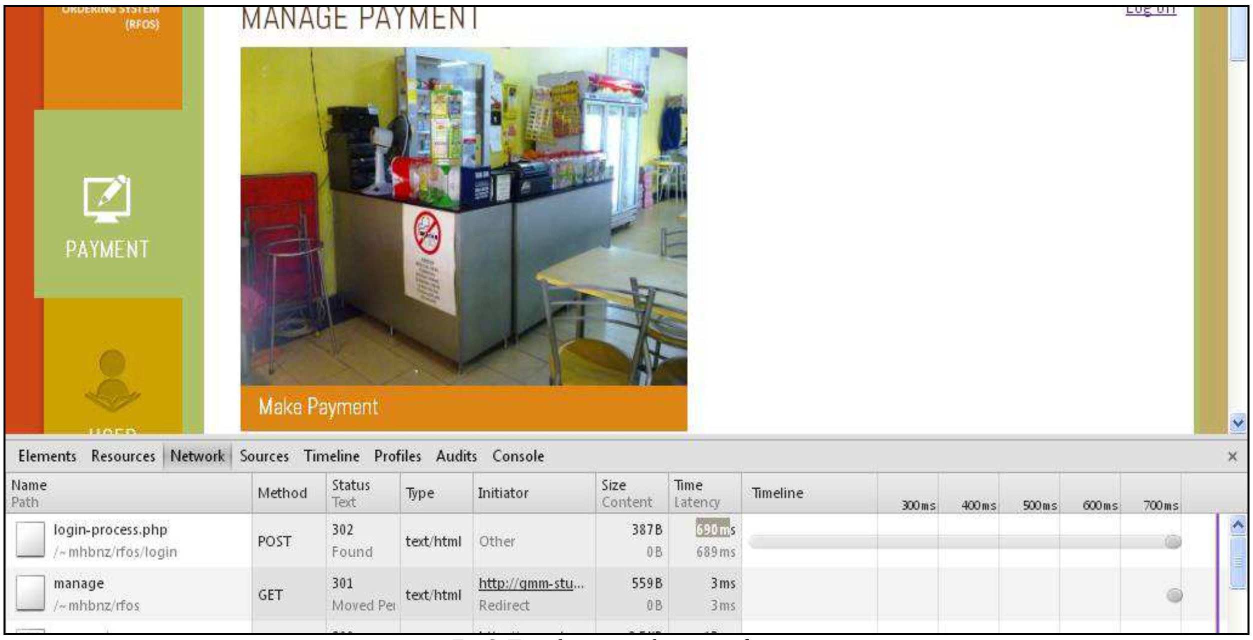
B. RFOS in Test by Feature concerning Use Trace Phase

This section will report the results of the Test by the Feature phase. First, they performed stress testing on Make Payment feature, where they have around 50 users simultaneously making payment to the Make Payment feature and recorded the response time (based on time latency) and loading time (duration taken by the page to go to the next page). Fig 5 shows the result of the time latency. Fig 5 shows that it took 41 ms to respond to 50 simultaneous user queries and took around 5 seconds to load the next page. Next, the security codes were added on the same feature (Make Payment) and the same stress testing was conducted. Fig 6 shows the results of the experiment. Based on Fig 6, the time latency with the security feature is increased to 690 ms and the loading time increased by 45 seconds. In these two cases, the number of users simultaneously using the system is the same. This shows that security features do impact the performance of the system. If TANC was not used during the development phases, they will not be aware of this issue and might cause vulnerability to their system. They are unable to check whether the time difference is too significant for the system to operate smoothly or is it acceptable by the users' standards. The next step is for the tester to report to the developer either the bugs that have been found can be ignored or should be dealt with. Fig 7 shows the test report of the bugs that were found during the tests depicted in Fig 5 and Fig 6. The bug that causes a little delay on the response and loading time of the feature with the addition of SQL and XSS is then reported to the developer team. However, the bugs are considered as acceptable because the delay is tolerable and not too long. This step is critical to make sure that the feature is thoroughly tested and does not need to be fixed and retested. Besides, this shows a change impact on the system NFR has been traced.



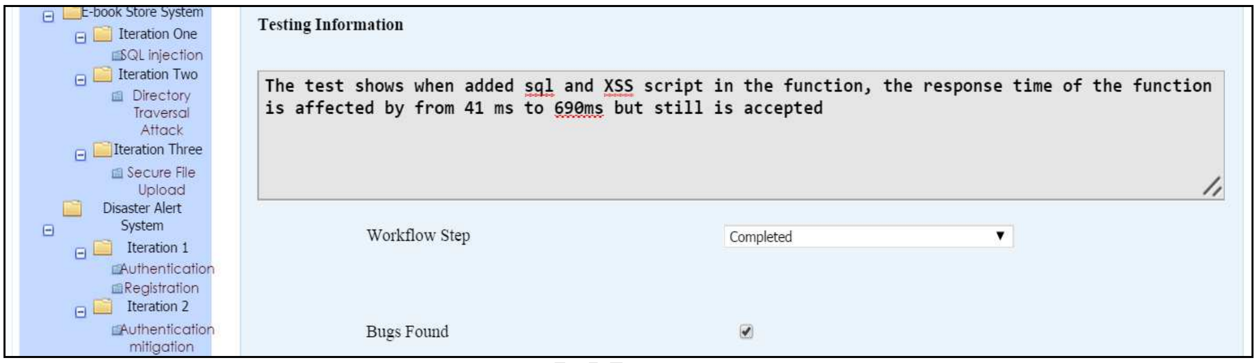
Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline
login-process2.php /~mhbnz/rfos/login	POST	302 Found	text/html	Other	385B 0B	41ms 41ms	
menu /~mhbnz/rfos	GET	301 Moved Per	text/html	http://qmm-stu... Redirect	555B 0B	3ms 3ms	
menu/		302		http://qmm-stu...	251B	4ms	

Fig 5: Time latency without security feature



Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline
login-process.php /~mhbnz/rfos/login	POST	302 Found	text/html	Other	387B 0B	690ms 689ms	
manage /~mhbnz/rfos	GET	301 Moved Per	text/html	http://qmm-stu... Redirect	559B 0B	3ms 3ms	

Fig 6: Time latency with security feature



Testing Information

```
The test shows when added sql and XSS script in the function, the response time of the function
is affected by from 41 ms to 690ms but still is accepted
```

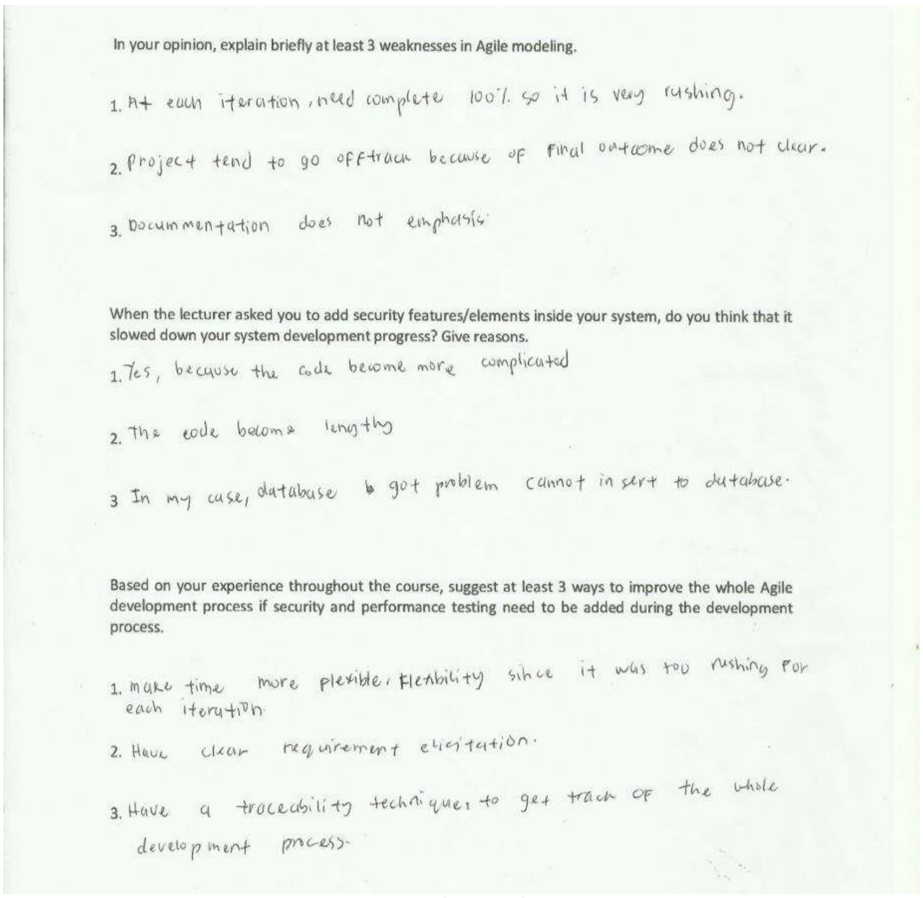
Workflow Step	Completed
Bugs Found	✓

Fig 7: Test report

III. Results and Discussion

This section reports the survey feedback which was collected during the case study. The case study is conducted as a cross-sectional survey where the unit of analysis is the individual sample that is involved in the practice of Agile methods [18], [19]. Our survey questionnaire asks for the students' opinions [20] and they agree with us. When asked about, in the whole issue in Agile modeling in managing change impact on system NFR (Question 1 & 2), and what they consider to be the most suitable solution in handling the issues (Question 3).

This feedback was collected from 24 samples (four of them performed the Rafi's Food Ordering system) that have just finished their project using FDD, Scrum, and XP. The survey contains four questions and some samples offer multiples answers based on their understandings. Fig 8 provides a sample of answered survey questionnaires. The rest of the feedbacks is presented on tables below based on the respective questions.



In your opinion, explain briefly at least 3 weaknesses in Agile modeling.

1. At each iteration, need complete 100% so it is very rushing.
2. Project tend to go offtrack because of final outcome does not clear.
3. Docummentation does not emphasis.

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 reasons.

1. Yes, because the code become more complicated
2. The code become lengthy
3 In my case, database got problem cannot insert to database.

Based on your experience throughout the course, suggest at least 3 ways to improve the whole Agile development process if security and performance testing need to be added during the development process.

1. make time more plexible, Flexibility since it was too rushing for each iteration
2. Have clear requirement elicitation.
3. Have a traceability techniques to get track of the whole development process.

Fig 8: Example of Survey feedbacks

Question 1: In your opinion, explain briefly at least 3 weaknesses of Agile modeling

TABLE I
FEEDBACKS FOR QUESTION 1

Suggestions	No. of Subjects
Suitable to use for a small team. Need tool support if the number of a team member is increasing	4
Product delivery is too frequent before the system finalized making the system need much modifying before the final product delivery.	13
Each iteration delays the project's completion time and extends the date for the new iteration phase	8
The final product is always delayed because the feature keeps on changing	13
Do not emphasis on documentation and management especially in the early stage	14
Needs to simultaneously develop the system as well as update the system's documentation	2
Every member needs to complete their tasks in order to complete the whole system	4
Do not employ any traceability technique	8

Based on the survey done, the problems or issues regarding the existing Agile methodology are listed above. As can be seen in Table 1, 75% of the samples' result shows that Agile does not emphasize on managing and updating the documentation of the project progress. The next major issue is that the requirement of the final product keeps changing. It means that as the project starts to progress, requirements provided by the clients are always changing and a slight change can impact the whole system. Due to this, the program cannot be finished according to the expected completion date.

To sum it up, Agile software is not good at managing changes. However, this does not mean that this method cannot accept changes, but it is weak in managing changes and the impact brought by changes toward the developed system. Out of the 24 samples' results that were collected during the case study, 6 students gave only 2 suggestions (25% of the total samples) and the rest gave 3 suggestions

(75% of the total samples). In total there are 66 responds were recorded.

Question 2: When the lecturer asked you to add security features/elements inside your system, do you think that it will slow down your system development progress.

TABLE II
FEEDBACKS FOR QUESTION 2

Suggestions	No. of Subjects
Yes	23
The number of codes needs to be increased too	17
Must check the system security and performance after security enhancements were made	16
The code becomes more complicated	13
More technical problems occur after changes, for example, database error	11
The process of development becomes more complicated	6
No	1
It is part of the iteration	1

Next, in the second question, the students were asked on the instruction to insert security features inside their projects which lead to delay or to increase the project cost. If this instruction is not given, they will most likely ignore this feature or did not give their best effort in implementing it. The results of this question are given in Table 2. Majority of them (96 percent) said that the addition of security elements in the middle of their project do interrupt the project progress. The top reason given to this issue is due to the increase in the number of codes needed in the system. Extra codes mean increased effort and time needed to develop the system and this, in turn, drags the development process. Moreover, they also need to check the overall system performance after the security features have been added, resulting in the need for extra time and effort.

However, one of the students viewed this matter in a different light. He stated that this process does not affect the completion duration because security features should be considered as part of the iteration from the beginning. Finally, based on the feedbacks, general issues in the Agile development process and problems that arise in handling NFR changes during the development phase will be investigated. Several suggestions were given on solving the problems of the Agile development process. Out of 24 samples' results, 20 of them show that Agile needs a proper traceability technique and its system progress should be rechecked so that if any change happens, the change can be traced and parts impacted can be determined. By doing this, the students do not need to check that everything is in order and the whole system is not affected by the changes made. In tackling this, 11 of them proposed to implement security features provided in the HTML tag. In other words, they recommended that the development tools should be able to build, check, and trace NFR automatically. Fig 9 shows the number of samples that provide one, two, or three suggestions. This bar chart analyses the majority numbers of respondents that give feedback. Referring to the chart, for Question 3, all samples provided three different suggestions, and for Question 2, only one sample provided one suggestion which that sample provided reason why sudden changes during the development phase do not affect the duration of the project.

Question 3: Based on your experience throughout the course, suggest at least 3 ways to improve the whole Agile development process if security and performance testing need to be added during the development process.

TABLE III
FEEDBACKS FOR QUESTION 3

Suggestions	No. of Subjects
Use Built-in Security Features Provided in The Html Tag	11
Use Simple Security Features	8
Use A Built-in System in Input Form	3
Allocate More Flexible Duration for Each Iteration	6
Have Clear Requirements in The Earlier Stage	8
Have A Proper Traceability Technique and Recheck the System Progress	20
Have A Complete Design During the Early Development Stage	2
Use Commercial Tool for Independent Testers	3
Perform Static Analysis	2
Provide an in-depth explanation on Agile Methodology	1
Use proper software development management lifecycle	3
Specific Roles for Tester	5

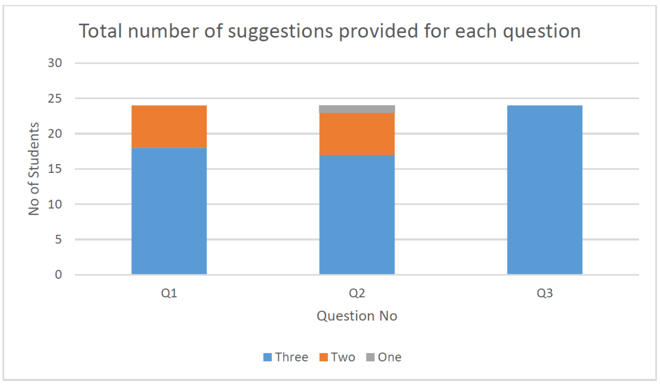


Fig 9: Total number of suggestions provided for each question

TABLE IV
PERCENTAGE OF FEEDBACKS AND TOP SUGGESTIONS

Questions	**Q1**	**Q2**	**Q3**
Total number of respondents	24	24	24
Percentage of samples that respond with three suggestions	75%	70.7%	100%
Percentage of samples that respond with two suggestions	25%	25%	0%
Percentage of samples that respond with one suggestion	0%	4.3%	0%

Table 4 shows the percentage of feedbacks based on the number of suggestions that they provided. This table is directly related to the bar chart before. Out of the 24 samples, 58% of them said the main problem with Agile software development in handling change impact is that it does not emphasize documentation and management aspects, especially in the early stages. Next, 95.8% of them stated that a sudden change during Agile development does affect the project duration. Out of that 95.8 %, 70.8% said that adding security features increases the number of codes in the system, and simultaneously increases the testing and development time for the entire process. Lastly, 83.3% of the samples shows that Agile software development needs a proper traceability approach in handling this issue.

Based on the discussion above, the majority of the students understand the goals and objectives of the project. They can grasp the purpose and problem presented in the case study given to them. This results in their understanding of the purpose of the case study and the problem related to the issues that this case study is trying to investigate. Besides that, feedbacks obtained from the survey found that there is a consensus between the issues presented by the case studies and feedback obtained from the students (Table 4 and Fig 9). Therefore, it is safe to say that Agile software development method needs traceability techniques to manage change impact on NFR of the system. Therefore, the reason behind the execution of this case study has been well justified that is to improve Agile software development, specifically in the area of change impact analysis.

IV. Conclusions

Based on this case study, the need for the Traceability approach for tracing change impact in Agile software development methodology, which offers better techniques in tracing change impact during the agile development process. The first main issue is the challenges of tracing the NFR change impact in the existing Agile Development. Then, this case study has investigated whether the Agile software development model, FDD could handle the NFR change impact management. This case study has proven that there are issues in tracing change impact especially in the term of NFR in the existing agile software development model. FDD was applied and they could not identify the impact of the system performance when they added security features in certain functional features in their system. Based on Fig 5 and 6, there are some changes that they did not expect to happen where the time latency was affected when they inject XSS mitigation code on the manage payment feature. Each case study strengthens the justification for the change impact issues in Agile software development methodology.

Acknowledgment

This research is supported by Grant Tier 1, Vot no: H130, Universiti Tun Hussein Onn Malaysia (UTHM).

References

[1] M. Martínez Pérez, C. Dafonte, and Á. Gómez, “Traceability in Patient Healthcare through the Integration of RFID Technology in an ICU in a Hospital.,” *Sensors Basel Sensors*, vol. 18, no. 5, May 2018.

[2] S. Kim, H. Kim, J. A. Kim, and Y. Cho, “A study on traceability between documents of a software R&D project,” in *Advanced multimedia and ubiquitous engineering*, vol. 354, J. J. Park, H.-C. Chao, H. Arabnia, and N. Y. Yen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 203–210.

[3] Q. Lu and X. Xu, “Adaptable Blockchain-Based Systems: A Case Study for Product Traceability,” *IEEE Softw.*, vol. 34, no. 6, pp. 21–27, Nov. 2017.

[4] F. Furtado and A. Zisman, “Trace++: A traceability approach to support transitioning to agile software engineering,” in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 2016, pp. 66–75.

[5] T. Vale, E. S. de Almeida, V. Alves, U. Kulesza, N. Niu, and R. de Lima, “Software product lines traceability: A systematic mapping study,” *Inf. Softw. Technol.*, vol. 84, pp. 1–18, Apr. 2017.

[6] R. Elamin and R. Osman, “Towards requirements reuse by implementing traceability in agile development,” in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, 2017, pp. 431–436.

[7] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki, and T. Grechenig, “Systematic literature review on agile practices in global software development,” *Inf. Softw. Technol.*, vol. 96, pp. 161–180, Apr. 2018.

[8] B. Fitzgerald, K. Stol, R.O. Sullivan, and D.O Brien, Scaling Agile Methods to Regulated Environments: An Industry Case Study, In: 35th International Conference on Software Engineering (ICSE), pp. 863-872.2013.

[9] L. Passos, C. Krzysztof, A. Sven, W. Andrzej, K. Christian, and G. Jianmei, Feature-oriented software evolution, In: Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems.2013,

[10] J. Zhang. "The software development process methodology of resource-based access control." In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 4, pp. 111-117. IEEE, 2010.

[11] L. K. Roses, A. Windmöller, and E. A. do Carmo. "Favorability conditions in the adoption of agile method practices for Software development in a public banking." *JISTEM-Journal of Information Systems and Technology Management* 13, no. 3, pp. 439-458, 2016.

[12] N. Spasibenko, and A. Besiana. "Project Suitability for Agile methodologies." 2009.

[13] J. Holvitie, S.A. Licorish, R.O. Spínola, S. Hyrynsalmi, S.G. MacDonell, T.S. Mendes, J. Buchan, and V. Leppänen. “Technical debt and agile software development practices and processes: An industry practitioner survey”. *Information and Software Technology*, *96*, pp.141-160, 2018.

[14] Al-Zewairi, Malek, Mariam Biltawi, Wael Etaiwi, and Adnan Shaout. "Agile software development methodologies: survey of surveys." *Journal of Computer and Communications* 5, no. 05 pp. 74-97.2017.

[15] Rigby, Darrell K., Jeff Sutherland, and Hirotaka Takeuchi. "Embracing agile." *Harvard Business Review* 94, no. 5, pp. 40-50.2016.

[16] Anwer, Faiza, Shabib Aftab, Usman Waheed, and Syed Shah Muhammad. "Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey." *International journal of multidisciplinary sciences and engineering* 8, no. 2 pp. 1-10. 2017.

[17] S. Gayer, A. Herrmann, T. Keuler, M. Riebisch, and P. O. Antonino, “Lightweight traceability for the agile architect,” *Computer*, vol. 49, no. 5, pp. 64–71, May 2016.

[18] P. Gregory, L. Barroca, H. Sharp, A. Deshpande, and K. Taylor, “The challenges that challenge: Engaging with agile practitioners’ concerns,” *Inf. Softw. Technol.*, vol. 77, pp. 92–104, Sep. 2016. M. Senapathi and A. Srinivasan, “An empirical investigation of the factors affecting agile usage,” in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE ’14*, New York, New York, USA, 2014, pp. 1–10.

[19] M. A. Brito and F. de Sá-Soares, “Assessment frequency in introductory computer programming disciplines,” *Comput. Human Behav.*, vol. 30, pp. 623–628, Jan. 2014.