Journal of Theoretical and Applied Information Technology
10th April 2015. Vol.74 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# SECURE DYNAMIC SYSTEM DEVELOPMENT METHOD (SDSDM): A SURVEY ABOUT ITS SUITABILITY

**[1]IMRAN GHANI, [2]ABDULLAHISANI, [3]NAGHMEH NIKNEJAD, [4]MANNIR BELLO, [5]SHAHID KAMAL, [6]MUHAMMAD WASEEM CHUGHTAI, [7]SEUNG RYUL JEONG**

[1, 2,3,4,5,6] Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Bahru 81310, Malaysia
[7]School of Management Information Systems, Kookmin University, Seoul, Korea
E-mail: [1] imransaieen@gmail.com, [2] abdulsani2@yahoo.com, [3] n.niknezhad@gmail.com, [4] mannirbello2@yahoo.com, [5] skamaltipu@gmail.com, [6] mr.chughtai88@gmail.com, [7]srjeong@kookmin.ac.kr,
Corresponding author: Seung Ryul Jeong

**ABSTRACT**

Building secure software is about taking security into account during all phases of software development. However, the major problem in agile methods is the lack of basic security elements in their phases and practices. One of such method is Dynamic System Development Method (DSDM).Based on this study, we have observed that the original/traditional DSDM does not help guide the agile to develop secure software. In order to address this issue, we introduced additional phases and sub-phases to the original/traditional DSDM to integrate security. The proposed model is named Secure Dynamic System Development Method (SDSDM), which has six phases. These phases and sub-phases are feasibility study, functional model iteration, secure functional model iteration, secure design, design and build iteration and implementation. Our findings highlight an improved agility in DSDM after integration of security. However, the study focuses on a questionnaire (survey) where the subject matter experts' opinion has been used to validate our model. Based on the experts' opinion, we can say that it is possible to develop secure software using SDSDM model without affecting its agility negatively.

**Keywords:** *Agile Methodology, Software Security, Dynamic System Development Method, Degree of Agility*

## 1. INTRODUCTION

Software security is one of the critical concerns for a number of years [6] [7] [8] [9][10][11][12]. The agile software development processes are criticized to develop unsecure software. One such model is DSDM, which we have enhanced to support secure software development. The enhanced model is called SDSDM. The enhancement is based on guidelines of security principles [3] mentioned in the Section 2. This paper presents the survey results on the suitability of our proposed SDSDM[4] obtained from the respondents of Agile Symposium 2013 held in Melaka, Malaysia. For the sake of reader's convenience the proposed SDSDM is re-presented in Figure 1. In fact, the SDSDM proposes some enhancements to the original DSDM [1]. In Figure 1, the yellow color presents the existing DSDM phases and sub-phases whereas the blue color presents the enhanced security phases and sub-phases.

## 2. SOFTWARE SECURITY PRINCIPLES

Software security principles [3] seem to be the foundational tenets of the software security domain. These principles represent the experiential knowledge of software security. By leveraging them, we can gain access to the scalable wisdom necessary for assessing and mitigating the security risk. For a full understanding and treatment of the security principles, in this research, we considered these, which have been ignored in the agile software development methods. The principles are: Least Privilege, Failing Securely, Securing the Weakest Link, Defense in Depth, Separation of Privilege, Economy of Mechanism, Least Common Mechanism, Complete Mediation, Reluctance to trust, Never assuming that your secrets are safe,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Psychological acceptability, and Promoting privacy.

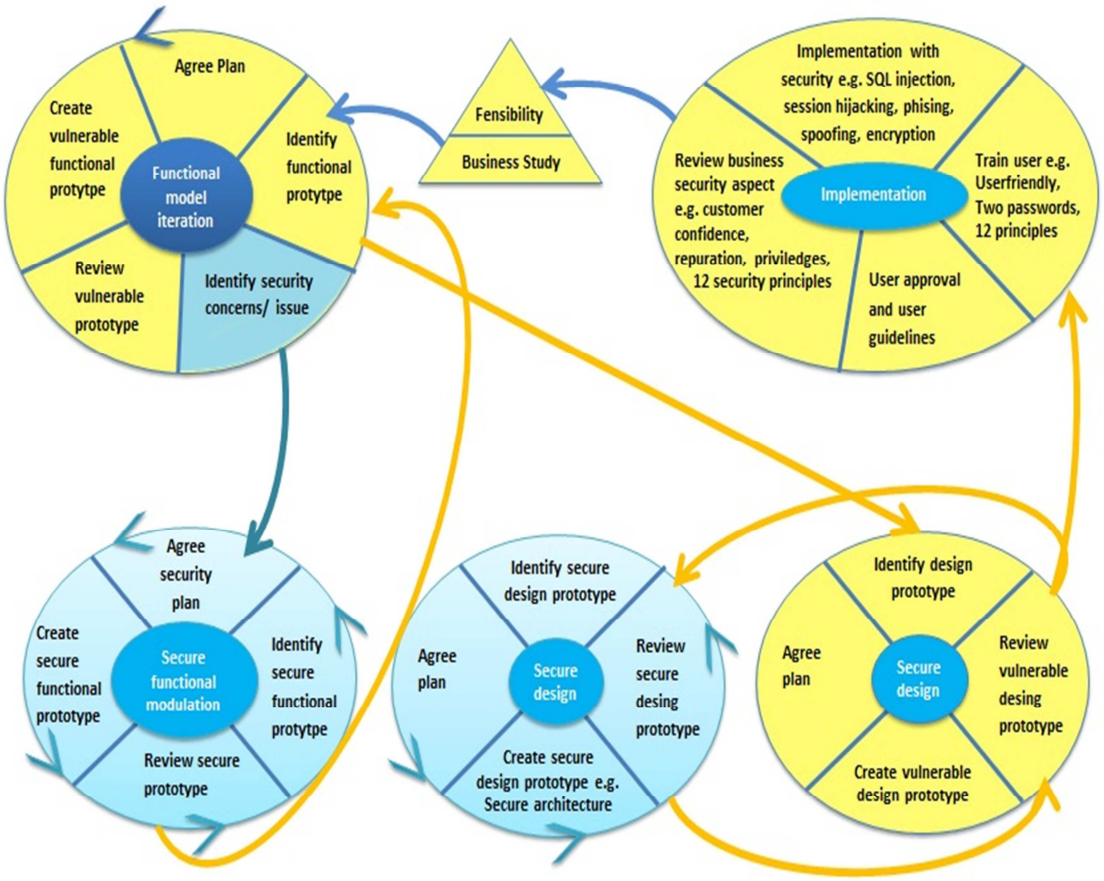


*Figure 1. Enhanced Secure Dynamic System Development Model (SDSDM)*

The security issues that we considered are SQL injection, session hijacking, spoofing, encryption and so on.

The objective of this survey was to validate the suitability of the proposed SDSDM based on the opinion of subject matter experts. This survey provides numerical description of opinions. In order to conduct the survey, the authors studied the existing samples of previous similar researches [2][5][13] that used the similar methodology to generalize the results. We believe that this survey will help the agile community to understand the problems being faced in software vulnerability particularly by using traditional DSDM. In addition, the survey agile teams are encouraged to adapt the DSDM in terms of their software security needs.

## 3. VALIDATION AND EVALUATION IN RELATION SECURITY PRINCIPLES

This section presents the opinion of respondents with respect to the application of security principles into SDSDM. The Table 1 below shows the opinion of respondents that whether the security principles have been properly implemented in SDSDM or not.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 1. Relationship Between Phases And Security Principles*

| Security Principles / SDSDM Phases | Least privilege | Failing securely | Securing the weakest link | Defense in depth | Separation of privilege | Economy of mechanism | Least common mechanism | Reluctance to trust | Never assuming that your secrets are safe | Complete mediation | Psychological acceptability |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Feasibility | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Functional model iteration | √ | √ | √ | √ | √ | √ | √ | X | X | √ | X |
| Secure functional model iteration | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Secure Design | √ | √ | √ | √ | √ | √ | √ | √ | | √ | √ |
| Design and build iteration | √ | X | √ | √ | √ | √ | √ | X | X | √ | √ |
| Implementation | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |

The √ symbol means that the certain phase has taken care of the respective security principle, while the X symbol means that phases do not take care of the respective security principle. Based on analysis and security principles in Table 1, we can see the security guidelines have been properly aligned to each related phases and sub- phases. The chosen factor is based on security principles stating that nearly all principles have provided guidelines for requirement, development and testing activities.

## 4. FINDINGS AND DISCUSSIONS BASED ON RESPONSES

The Figure 2 below presents the analysis of responses obtained from the Thirteen (13) participants of agile symposium in the survey. The survey contained 18 questions, which are shown on the horizontal axis of Figure 2.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

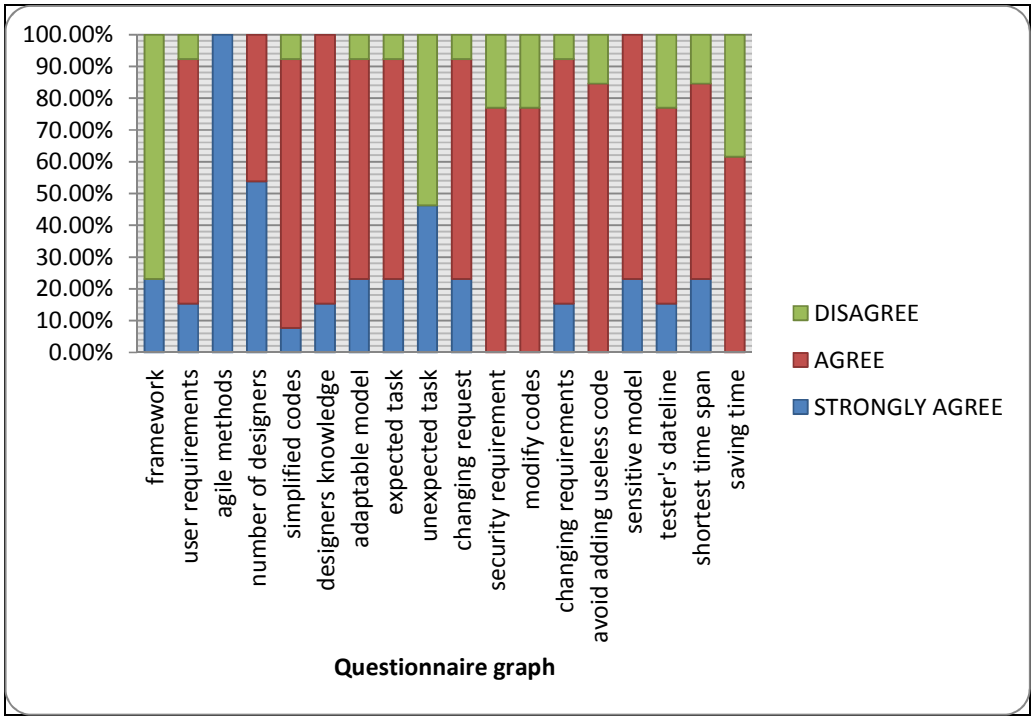


*Figure 2. The Graph Of Questionnaire And Responses*

Figure 2 illustrates that the percentage of positive opinion respondents who agreed (in red color) on the new improvement of security features in SDSDM has the highest proportion and those who strongly agreed (in blue color) followed that. The lowest proportion is in terms of disagreement (in green color) given by respondents which is negligible compared to those agreed with the SDSDM. Though the sample size of participants is quite small, however, we can deduce that the SDSDM should be applicable in terms security roles, practices and phases to develop secure software. As a future work, the SDSDM should be applied in real life agile environment by agile teams. Then the survey should be conducted in that environemnt which could provide more insights about the feasibility of SDSDM in real environment.

The next section presents the survey results about the degree of agility of SDSDM.

## 5. DEGREE OF AGILITY OF SDSDM

Qumer and Henderson-Seller [14] presented the following definition of the agility of any entity. "Agility is a persistent behavior or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal and external environment."

The aforementioned definition has been simplified into five key elements of agility that are flexibility, speed, leanness, learning and responsiveness. To validate the performance of an agile model, the degree of agility needs to be evaluated by measuring the following elements.

i. Flexibility: The agile method should be flexible enough to welcome a change in requirements during any phase. A lack of flexibility will create a serious crisis in the agile software process. In other words, we can say that this represents the ability to respond to any change at any given time.
ii. Speed: As the speed of software delivery is one of the most important elements in the agile manifesto, it needs to be considered as an element of agility [15].
iii. Leanness: This represents the elimination of waste or the doing of more with less. Through maximizing the utilization of all resources, and the elimination of unnecessary resources,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

all tasks are streamlined. At the same time, however, the level of quality should be maintained.

iv. Learning: Focuses on improvement during and after product development when software had been delivered to the client.

v. Responsiveness: This means responding to any change, either within the team, or in the requirements of the software itself [14]. Any software development process that implements the whole of agility, as stated above, can be considered to be an agile method. However, its degree of agility needs to be evaluated before it can be considered to be a suitable agile model.

### 5.1 Requirement Phase and Degree of Agility

In SDSDM, the requirement phase is conducted by the secure functional model iteration. After collecting the requirements, the selected features are populated in the functional model iteration. The functional model iteration is forwarded to the secure functional model iteration to evaluate and then to the secure design. The security design is evaluated and a conclusion is reached based on the agility provided. If agility is

ii. Flexibility: **F**lexibility refers to adaptability to expected change requirements. In the beginning, it is difficult to be flexible due to the lack of information within the security design. However, this information is also simple enough to analyze. Although flexibility can be affected in the beginning, with time it can be recovered however.

iii. Leanness: The security design can complete a task within the shortest available time span because of its simplicity. Due to this simplicity, the secure design can provide a proper understanding to the functional model iteration in a short amount of time. As a result, leanness will not be affected.

iv. Learning: If the description in a security design is user friendly, it can maintain or improve upon the current knowledge of the functional model iteration. Since the history of previous requirements can be maintained inside the secure design, it can also provide experience to new secure functional model iteration. As a result, the security design in the requirement phase will have a positive learning curve.

### 5.2 Development Phase and Degree of Agility

In SDSDM, the development phase is evaluated by the secure design and system

affected then the secure functional model iteration suggests improvements.

i. Speed: Overall, speed does not affect the delivery deadline when a secure design is implemented in the requirement phase. Since the secure design has an additional workforce available (such as security masters), they can identify if previously developed security libraries or classes can be reused. This information may be provided in the description of security requirements in the secure design. Another way in which the secure design can save time for the functional model iteration is by having guidelines without the need for research, or by finding a specific security requirement. Since the functional model iteration and secure functional model iteration have simple guidelines to follow in terms of security, the results of security requirements can be generated quickly. In conclusion, the security design would not affect the overall delivery deadline. It can also save time and produce swift security results.

developers in the team. The secure design illustrated is evaluated using their experience. If the agility is affected, they suggest improvements.

i. Speed: The secure design does not cause any slowdown in developing the system within the required timeframe. However, sometimes it depends on the flexibility or the complexity of a business case. Referring to the description provided in the secure design can also help the developers save time in tracing any errors or bugs that have occurred in the sub-modules. By using the secure design, the developers can also add or modify any new code to any of the sub-modules for implementation of security. As a result, the speed is generally not affected.

ii. Flexibility: The developer can accommodate expected changes or user requirements within any sub modules during the development period. For unexpected changes the secure design can be updated from time to time.

iii. Leanness: Since the secure design will have guidelines and standards for the developers, they can still maintain simplified codes during development to produce quality products through the help of secure design. Developers can avoid adding any un-used or useless codes when developing by using the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

secure design. This can be done in the beginning and can then be reused.

iv. Learning: Secure design can maintain or improve the current knowledge of the developers regarding the product or system that is being developed. Since the descriptions can help new developers develop new code, the secure design also provide experience to developers regarding the previously developed products.

### 5.3 Testing Phase and Degree of Agility

In SDSDM, the testing phase is evaluated by the team's testers. The secure design is consulted due to their experience. If agility is affected, the tester suggests improvements.

i. Speed: The implementation of secure design in DSDM development improves the tester's work flow. Use of the secure design can save time during the testing phase in selecting the appropriate method, since the requirements will be focused on testing SQL injection, buffer flow, session hijacking, phishing, and encryption and so on. The testing process will become faster and easier as the tester comes to more quickly understand the features "to be tested" from the design and build iteration. With this information provided in the secure design, all testing can produce quick results.

ii. Flexibility: The secure design is not rigid and offers the flexibility to add, modify or delete the security features scheduled to be tested.

iii. Leanness: The secure design helps the tester complete tasks and activities in the shortest amount of time by going through the major list.

iv. Learning: The tester gains new knowledge as information inside the secure design is stored and updated. This saved experience will allow a new tester to improve and learn more quickly.

### 5.4 Secure DSDM Evaluation about Responsiveness

The secure phase is the one responsible for evaluating the overall responsiveness of an agile development process using secure DSDM design. It has been noted that secure phase did not affect the responsiveness of the overall project. Secure phase is sensitive and can adapt to the environment. It has been applied on a small and large scale business application and accepted by an experienced team using the DSDM method. Their feedback is shown in Table 2.

*Table 2.Feeback Obtained In Agile Symposium Malaysia 2014*

| | | Security Questions | Questions for Developer | Questions for Designer | Question for Tester |
|---|---|---|---|---|---|
| Speed | Delivery on time | Agree | Agree | Agree | Agree |
| | Help saving time | Agree | Strongly Agree | Agree | Agree |
| | Produce result | Agree | Agree | Disagree | Agree |
| Flexibility | Method accommodate expected changes | Disagree | Agree | Agree | Agree |
| | Method accommodate unexpected changes | Strongly Agree | Agree | Strongly Agree | Disagree |
| Leanness | Shortest time span | Agree | Strongly Agree | Agree | Agree |
| | Economical | Agree | Strongly Agree | Agree | Agree |
| | Quality production | Agree | Agree | Agree | Agree |
| Learning | Update prior knowledge | Strongly Agree | Agree | Agree | Agree |
| | Provides experience | Disagree | Agree | Agree | Agree |
| Responsiveness | Sensitiveness | Agree | Strongly Agree | Agree | Disagree |
| | Adaptability in team, | Agree | Strongly Agree | Agree | Disagree |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Based on the agile symposium result shown in Table 2, it is safe to say that security can be implemented as part of the enhancement process in a dynamic system development method selected for security. This information has been used to calculate agility using the 4-DAT framework, in order to compare agility both before and after security implementation was performed according to our model [14].

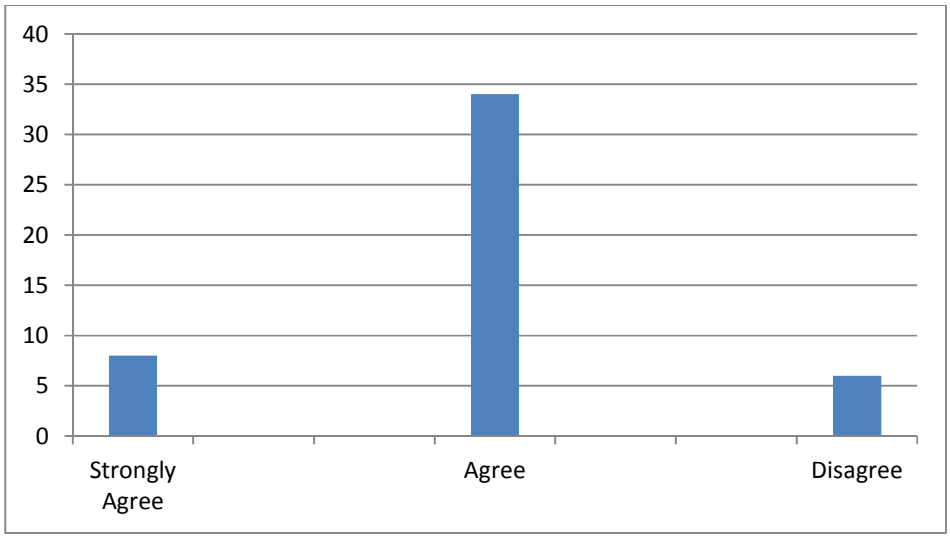


*Figure 3.Agile Symposium Feedbacks*

Figure 3 presents agile symposium feedback which shows that most respondents agree on integration of security in DSDM and few participants disagree on secure DSDM. On the left hand side is the number of participants against opinion. Thus, we can deduced that this integration of security into DSDM is worth doing.

### 5.5 Degree of Agility in Dynamic System Development Method

The 4-DAT dimension has been used to evaluate DSDM from an agility perspective and the degrees of agility have been measured in terms of the five variables (features) relating to Dimension 2: flexibility (FY), speed (SD), leanness (LS), learning (LG) and responsiveness (RS) that may exist at some specific level or lifecycle phase or as a result of the practices used in the phases of DSDM. If any phase or practice of DSDM supports a particular agility feature, then 1 point is allocated in that particular cell, otherwise 0; and so on.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 3.Degree Of Agility Before Enhancement*

| DSDM | Agility Features | | | | | |
|---|---|---|---|---|---|---|
| | **FY** | **SD** | **LS** | **LG** | **RS** | **Total** |
| ***Phases*** | | | | | | |
| Pre-project | 0 | 0 | 0 | 0 | 0 | 0 |
| Feasibility study | 0 | 0 | 0 | 0 | 0 | 0 |
| Business Study | 0 | 0 | 0 | 0 | 0 | 0 |
| Functional model iteration | 1 | 1 | 0 | 1 | 1 | 4 |
| Design and build iteration | 1 | 1 | 0 | 1 | 1 | 4 |
| Implementation | 1 | 1 | 0 | 1 | 1 | 4 |
| Post-project | 1 | 1 | 0 | 1 | 1 | 4 |
| Total | 4 | 4 | 0 | 4 | 4 | 16 |
| Degree of agility | 4/7 | 4/7 | 0/7 | 4/7 | 4/7 | 16/(7*5) |
| ***Practices*** | | | | | | |
| Active User involvement | 0 | 1 | 0 | 1 | 1 | 3 |
| Empowered teams | 1 | 1 | 0 | 1 | 1 | 4 |
| Frequent Product delivery | 1 | 1 | 0 | 1 | 1 | 4 |
| Iterative and incremental development | 1 | 1 | 0 | 1 | 1 | 4 |
| Reversible changes | 1 | 1 | 0 | 1 | 1 | 4 |
| Requirements are baselined at high level | 0 | 0 | 0 | 0 | 0 | 0 |
| Integrated testing | 1 | 1 | 0 | 1 | 1 | 4 |
| Collaboration and cooperation among stakeholders | 1 | 1 | 0 | 1 | 1 | 4 |
| Total | 6 | 7 | 0 | 7 | 7 | 27 |
| Degree of Agility | 6/8 | 7/8 | 0/8 | 7/8 | 7/8 | 27/(8*5) |

The degree of agility in DSDM can be seen in Table 3. Here we can also see that SDSDM, both at the process and practices level, supports all agility attributes except Leanness (LS). It is also noted that three DSDM phases (''The Pre-Project'', ''Feasibility Study'' and ''Business Study'') and one practice (''Requirements are base lined at High Level'') do not support any of the agility attributes at all and are marked as having ''zero'' agility in the cells. In reference to the practices in Table 3:

i. The 0 in column four means leanness was not achieved by the SDSDM. This means the secure design was not effective enough to practice security for the DSDM.
ii. The 1 in other columns means that the other agility features were achieved.
iii. 7/8 is the total of the values in columns 2, 4, 5. It means the relevant agilities in the columns were achieved.
iv. 0/8 is the total of the values in column 3. It means the relevant agility in the column was achieved.
v. 6/8 is the total of the values in column 1. It means the relevant agility in the column was achieved.

The following formula has been adopted in order to calculate agility in DSDM (Qumer and Henderson-Sellers, 2007).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$Degree\ of\ Agility\ Practices = \frac{A}{B*C} \quad (1)$$

Where **A** = Overall Sum of 1 for each Agility Feature in Each Practice , **B**= No of Agile Features and **C**= Number of Practices.
Thus, Degree of Agility of Practices = 27/(8*5) = 0.68
The formula, label 0 and 1 were also implemented in Tables 2 and 3 using the same technique. Table 4 shows the difference from Table 3, which includes the secure design. Questionnaire was distributed to agile symposium participants in which 13 were returned out of 25 we use feedback to evaluate the new model agree that the secure SDSDM can be implemented and does not affect the five criteria of agility i.e., flexibility, speed, leanness, learning and responsiveness. Based on the conclusion of Table 5.4's agile participants' evaluation, the SDSDM considered in Table 3 can be performed. Thus, number 1 should be put inside the blank space for all agility.

The data in Table 5 refers to the activity in practices for DSDM practices. The degree of agility has improved from 0.68 before implementation to 0.71 after implementation.

*Table 4. Degree Of Agility After Enhancement*

| **SDM** | Agility Features | | | | | |
|---|---|---|---|---|---|---|
| | **FY** | **SD** | **LS** | **LG** | **RS** | **Total** |
| ***Phases*** | | | | | | |
| Pre-project | 0 | 0 | 0 | 0 | 0 | 0 |
| Feasibility study | 0 | 0 | 0 | 0 | 0 | 0 |
| Business Study | 0 | 0 | 0 | 0 | 0 | 0 |
| Functional model iteration | 1 | 1 | 0 | 1 | 1 | 4 |
| Design and build iteration | 1 | 1 | 0 | 1 | 1 | 4 |
| Implementation | 1 | 1 | 0 | 1 | 1 | 4 |
| Post-project | 1 | 1 | 0 | 1 | 1 | 4 |
| Total | 4 | 4 | 0 | 4 | 4 | 16 |
| Degree of agility | 4/7 | 4/7 | 0/7 | 4/7 | 4/7 | 16/(7*5) |
| | | | | | | |
| ***Practices*** | | | | | | |
| Active User involvement | 0 | 1 | 0 | 1 | 1 | 3 |
| Empowered teams | 1 | 1 | 0 | 1 | 1 | 4 |
| Frequent Product delivery | 1 | 1 | 0 | 1 | 1 | 4 |
| Iterative and incremental development | 1 | 1 | 0 | 1 | 1 | 4 |
| Reversible changes | 1 | 1 | 0 | 1 | 1 | 4 |
| Requirements are base lined at high level | 0 | 0 | 0 | 0 | 0 | 0 |
| Secure Design | 1 | 1 | 1 | 1 | 1 | 5 |
| Integrated testing | 1 | 1 | 0 | 1 | 1 | 4 |
| Collaboration and cooperation among stakeholders | 1 | 1 | 0 | 1 | 1 | 4 |
| Total | 6 | 7 | 0 | 7 | 7 | 32 |
| Degree of Agility | 7/9 | 8/9 | 1/9 | 8/9 | 8/9 | 32/(9*5) |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 5. Calculate Degree of Agility*

| Process and Practices | DSDM (Before implementation) | DSDM (After implementation) |
|---|---|---|
| Phases | 16/35= 0.46 | 16/35=0.46 |
| Practices | 27/40=0.68 | 32/45=0.71 |

The degree of agility is illustrated in Figure 4, adds a comparison before and after the security phase has been added. The phases and practices have been calculated, and the result in Figure 4 shows the practices clearly.

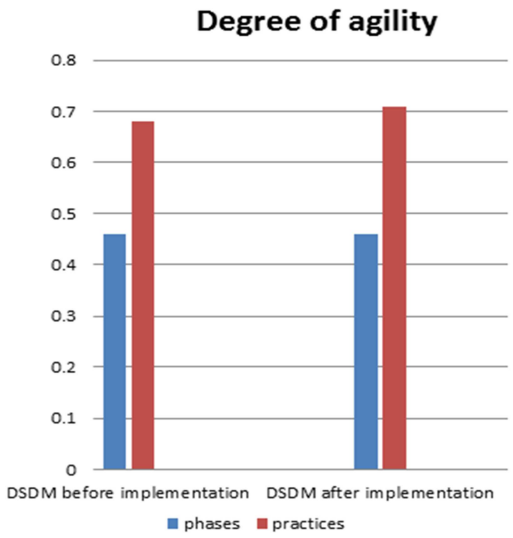


*Figure 4. Comparison Degree of Agility*

Based on Figure 4, the degree of agility has improved in the practices from 0.68 (before implementation) to 0.71 (after implementation). The improvement shown after implementation shows that the secure design does not add any delay to speed, flexibility, leanness, learning or responsiveness if the security applied is part of the DSDM. This shows that such implementation is relevant and can be performed without any fear of affecting agility negatively.

## 6. CONCLUSION

After analysis, comparison, and collection of literature such as journals, books, magazines and case studies, we were able to successfully identify the security issues in original/traditional DSDM model. Through further research, we were able to discover the relationship between the security principles and security in each of the DSDM phases. The second issue, then, was to enhance the DSDM model. The enhanced SDSDM model that we proposed has been evaluated in the requirement, development and testing phases. These research objectives were completed successfully. An agile team presented evaluations and feedbacks in regards to the enhancement model gathered from agile symposium participants. Based on their evaluations, the research was considered relevant, possible to try out and useful to implement. However, as mentioned earlier, there is a need for a survey that should be conducted in the real environment where secure software is a critical concern. That survey would provide more insights about the feasibility of SDSDM in real environment.

However, as mentioned earlier, there is a need for a survey that should be conducted in the real environment where secure software is a critical concern. That survey would provide more insights about the feasibility of SDSDM in real environment.

## REFERENCES

[1] DSDM Public Version, http://www.dsdm.com/products/dsdm_version_4_2.asp 4.2.(2013).

[2] Abdullahi, S., Adila, F.,Seung, R. J. and Imran, G. (2013). A Review on Software Development Security Engineering using Dynamic System Method (DSDM). *International Journal of Computer Applications*: *Published by Foundation of Computer Science, New York, USA*, **69**(25):33-44.

[3] Julia H. Allen, ed., Software Security Engineering: A Guide for Project Manager, *Addison wesley Professional*, 2008.

[4] AbdullahiSani, Imran Ghani,SeungRyulJeong,Secure Dynamic System Development Method (Sdsdm) Model For Secure Software Development,Sci.Int.(Lahore),**1059-64**,2013

[5] Imran Ghani; ZulkarnainAzham; SeungRyulJeong, Integrating software security into agile-Scrum method, *KSII Transactions on Internet and Information Systems*. 2014;**8**(2):646-663.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] Adila, F., Imran, G. and Nor I. M. (2013). Developing Websites using Feature Driven Development: A Case Study. *Journal of Clean Energy Technologies*, **4**(1):211-215.

[7] Alberts, J. C. and Allen, R. S. (2011). Risk-based measurement and analysis: Application to software security. *Software Engineering Institute, Carnegie Mellon University, Pittsburgh*.

[8] AntiVaha-Sipila, Marrying Scrum and Security http://blog.safecode.org/?p=45.ht. (2009).

[9] Bryan, S. Streamline Security Practices For Agile Development. MSDN Magazine. (2010).

[10] Chivers, H., Paige R. F. and Ge, X. Agile Security Using an Incremental Security Architecture.Department of Computer Science, *University of York*. (2007).

[11] Sipponen, M., and Baskerville, R. and Kuivalainen, T. Integrating security into agile development methods.*In Proceedings of the 38th Hawaii International Conference on System Sciences*. (2005).

[12] Torstein, N. and Richard, S. Agile Software Development.The Straight and arrow Path to Secure Software, *NTNU, Norway*. (2010).

[13] Azham, Z.,Ghani, I. and Ithnin, N. Security Backlog In Scrum Security Practices. *5th MySEC Malaysian Conference in Software Engineering*. (2011).

[14] Qumer, A. and Henderson-Sellers, B. (2007). An Evaluation of the degree of agility in six agile methods and it applicability for method engineering. Science Direct, 280-295.

[15] Walker, R. (2009). Improving Software Economics white paper, IBM.