Review of Secure Software Development in Various Agile Models

Imran Liaqat, Ahsan Raza Sattar, Nayyar Iqbal, Hafiza Sadia Hassan

Abstract— Software whether application or system are most valuable assets of a computer system. Hardware of computer is also dependent upon software. Initially software requirements, size, complexities and operating environment were limited. There are multiple development methodologies such as waterfall, spiral, rational unified process and now the most common agile. Scrum, XP, DSDM, FDD, and Crystal are some of commonly used agile software development methodologies being used for development of software in increments. Agile facilitates quick releases, iterations and customer involvement. Weak coding practices are major cause of software vulnerabilities. Secure programming rules should be followed for writing programs in order to manage security. Interactive Static Analysis integrates static analysis in Integrated Development Environment (IDE) it assists and indicate programmer such coding mistakes.

Index Terms— Comparison of Secure Agile, Secure Agile Development, Security, Vulnerabilities, Secure Software, Complexities, Security in Agile.

1 Introduction

Software has been a piece of current civilization for over fifty years. Software improvement began off as a mixed-up action regularly said as "code and fix". The product was composed without a lot of an arrangement, and the outline of the framework was resolved from numerous brief choices. This functioned very well for little frameworks yet as frameworks developed it turned out to be harder to include new components and errors were tougher to settle. Such form of improvement was utilized for a long time till an option was presented: Procedure. Techniques force a trained procedure upon software improvement with the point of building programming advancement more expected and more effective.

Traditional strategies are arrangement determined in which work starts with the collection and documentation of a comprehensive arrangement of necessities, trailed by building and abnormal state outline advancement and investigation. Because of these important angles, this procedure got to be to be known as heavyweight. A few specialists discovered this procedure driven perspective to software development unsolved and posture challenges when change rates are still moderately low. It includes the secure development processes for managers and developers [1].

In today's expanding unpredictability and vulnerability, skilled individuals need to effort in a relationship in to have better control over function and communication with companions, clients and administration. Issues are varying, individuals are altering and thoughts are evolving. Whereas there is quiet a requirement for arrangement determined style advancement and administration in a few circumstances the greater development deceits in deft and adaptable. This research examine different methods of software development in Agile and deft strategies for appropriateness in software improvement and audit episodic information given from experts to figure out which models suits superlative.

Due to the use of computer networks and internet the software products are exposed to the outer environment. So the need of developing secure software increases. It is an overhead and expansive work in terms of cost, time and other software development activities Security requirements must be gathered. Apart from this threat modelling and risk assessment must also be performed. These steps will help in developing more secure software systems [2].

Traditional/planned procedures are thought to be the routine method for creating software. These approaches are in light of a consecutive arrangement of steps, for example, basics definition, arrangement building, testing and sending. Heavyweight strategies oblige characterizing and archiving a steady arrangement of fundamentals toward the start of a venture. There are a wide range of heavyweight techniques such as: Waterfall, Spiral Model and Unified Process. Each methodologies has its own life cycle of software development in various stages. The brief description of these methodologies divided in different stages and part that shows the development process described as under.

2 Agile Methodologies

Agile – dedicating "the nature of being light-footed; status for movement; deftness, action, expertise in movement" as specified in the Oxford Dictionary – programming improvement systems are endeavouring to offer by and by a response to the excited business group requesting lighter weight alongside speedier and nimbler programming advancement forms. To give some examples of those created: Adaptive Software Development (ASD), Agile Modelling, Crystal Methods, Dynamic System Development, Lean Development and Scrum.

- *Imran Liaqat, Research Scholar, Department of Computer Science, University of Agriculture, Faisalabad, Pakistan. E-mail: imranliaqat111@gmail.com*
- *Ahsan Raza Sattar, Assistant Professor, Department of Computer Science, Unviersity of Agriculture, Faisalabad, Pakistan. E-mail: ahsan_raza@uaf.edu.pk*
- *Nayyar Iqbal, Lecturer, Department of Computer Science, University of Agriculture, Faisalabad, Pakistan. E-mail: nayyar_iqbal@hotmail.com*
- *Hafiza Sadia Hassan, Research Scholar, Department of Computer Science, University of Agriculture, Faisalabad, Pakistan. E-mail: sadiahassan60@gmail.com*

Every one of these philosophies recognized that astounding programming and all the more vitally consumer loyalty could just be accomplished by bringing "delicacy" to their procedures.
The old-style software development methodologies do not support daily changing requirements. These methodologies are plan driven that are applied to small scale projects. These methodologies are less flexible. Agile methods are flexible approaches that save time and cost. Agile methods are comfortable in managing constant changing requirements. These methods rely on strong customer communication between developers and customer. The basic rule of agile is to promote interactions, individuals, customer collaboration and comfort in changing requirements [3].

2.1 Extreme Programming (XP)

Extreme programming (XP) has advanced after the issues created by the extended improvement phases of conventional advancement models. The XP procedure can be portrayed by little advancement phases, incremental arranging, nonstop input, dependence on correspondence, and developmental outline. With all the overhead abilities, XP software engineers react to moving environment through a great deal more bravery. Further as per Williams, XP colleagues expend couple of minutes on programming, couple of minutes on venture administration, couple of minutes on configuration, couple of minutes on criticism, and couple of minutes on group building ordinarily every day. The expression "great" originates from taking these down to earth standards and observes to compelling stages.

The extension of XP to facilitate developer team to elicit security requirements in a better way. It was seven steps resulted in threat scenarios and security functionalities. Most important goal of this method was proactive approach to security requirements and conduction of risk analysis. Security Sensitive Asset Identification, Threat Scenario (Abuse Case Stories) Formation, Risk Assessment of Abuser Scenarios, Negotiation of User & Abuser Stories, Defining Security Related User Stories, Defining Coding Standards for Security and Cross matching of Abuser Stories were steps defined in this model [4].

2.2 Scrum

Scrum is an iterative, incremental procedure for building up any item or dealing with some work. Scrum focuses on in what way the colleagues ought to capacity so as to create the framework adaptability in an always showing signs of change environment. Toward the end of each emphasis it creates a possible arrangement of usefulness. The expression "scrum" began from a technique in the sport of rugger where it indicates "receiving an out-of-take care of business once again into the diversion" with collaboration.

Scrum does not oblige or give any particular programming improvement techniques/performs to be utilized. Rather, it involves certain administration practices and apparatuses in diverse periods of Scrum to keep away from the confusion by eccentrics and many-sided quality.

2.3 Feature Driven Development (FDD)

Feature Driven Development (FDD) was utilized interestingly as a part of the improvement of a substantial and complex managing an account application extend in the late 90's.Unlike alternate systems, the FDD methodology does not protection the whole software advancement transform but instead spotlights on the configuration and building stages.

The initial three stages are done toward the start of the venture. The previous two stages are the iterative piece of the procedure which bolsters the nimble improvement with speedy adjustments to late variations in necessities and business needs. The FDD methodology incorporates regular and unmistakable deliverables, alongside exact checking of the advancement of the report.

The weak coding practices were most important cause of software vulnerabilities. Secure programming rules were followed for writing programs in order to manage security. Interactive Static Analysis integrated static analysis in Integrated Development Environment (IDE) it assisted and indicated programmer such coding mistakes. If the coding did not right direction then the software cannot fulfil the requirements.

2.4 Dynamic System Development Method (DSDM)

The DSDM, Dynamic System Development Technique, was produced in the United Empire in the mid-1990. It is a mix of, and augmentation to, fast application improvement and Iterative advancement rehearses. Martin Fowler, one of the journalists of Agile Manifesto, accepts, "DSDM is remarkable for taking a great part of the foundation of more develop customary systems, while taking after the standards of the nimble routines approach". The essential thought late DSDM is to alter time and assets, and afterward change the measure of usefulness in like manner as opposed to settling the measure of usefulness in an item, and after that conforming period and assets to achieve that functionality.

3 Software Security

Security is an extreme issue in building up any product item. Much of the time the thought of what may happen if a product item is deliberately and vindictively assaulted, is disregarded. The product items nowadays are fragile to the point that they scarcely work appropriately when they have a primitive and predictable environment. At the point when nature, in which our product items run, gets to be forceful and noxious, the items fizzle drastically.

Security engineering procedures bolster software improvement designing in conveying arrangements that anticipate abuse and vindictive conduct. The procedures enhance the finished items security by including formal routines, prerequisites and predefined best practices for the designers. By having rules and solid steps the designers need to perform the procedures tries to compel the engineers to concentrate on security [5].
Software Security Touch focuses has regularly been depicted as a lightweight security designing process that incorporates centre exercises in a current advancement transform and enhances the quality and security part of the deciding item. Basic

Criteria is develop and all around utilized security designing primary that is ISO confirmed. Because of the utilization of PC systems and web the product items are presented to the external environment. So the need of creating secure programming increments.

4 Software Vulnerabilities

Vulnerabilities are a shortcoming in a product framework. Regularly, vulnerabilities have two conceivable roots, shortcomings in the execution of the product and imperfections in the product's configuration. Nonetheless, source code based vulnerabilities are still programming blames; the issues can simply spread into a particular kind of disappointment, security weakness.

Software improvement in disseminated dexterous system developing and dangers in such environment gets to be increment. Programming association was working in a tight conditions and very extreme environment. In conveyed light-footed improvement numerous elements impact the spry group on the grounds that the partners and colleagues of advancement group scattered in diverse territories. Appropriated Agile Development (DAD) creates programming with ease and association changes the business necessities effortlessly [6].

The idea of ahead of schedule helplessness location is like right on time flaw recognition. The aim is to recognize any peculiarity bringing about powerlessness in the item that would oblige push to be rectified after the item has been discharged to clients. Endeavours in recognizing a particular kind of defencelessness ought to additionally be engaged in the stage and strategy that is most financially savvy for that sort of helplessness. Studies in right on time shortcoming recognition have demonstrated that distinguishing the deficiency prior being developed decreases the improvement cost.

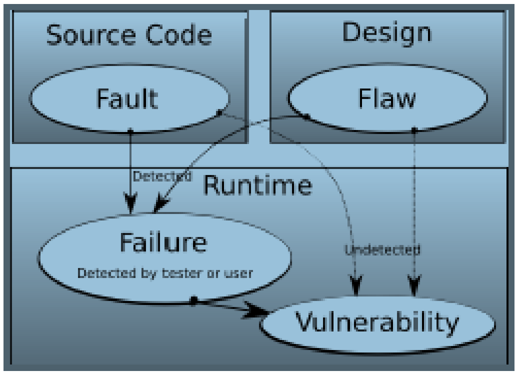


Fig. 1. Origin and Propagation of Vulnerabilities

4.1 Design Vulnerabilities

At the point when planning an item it is conceivable to add prerequisites to the outline that would influence the finished items security. On such normal configuration helplessness is the locking of clients after a foreordained measure of fizzled confirmation endeavours. This prerequisite is frequently named as a security necessity to avoid animal power secret key speculating.

4.2 Implementation Vulnerabilities

All product activities produce no less than one normal ancient rarity, the items' source code. At the code level, the emphasis is set on usage issues, particularly those that are noticeable by static investigation instruments. Nonetheless, knowing the execution blames that are not distinguished is additionally fascinating as it can be utilized to manage where other more costly location systems ought to centre their examination on.

4.3 Configuration Vulnerabilities

A completed item will in the end be conveyed on a client's system. The setup of the item can be essential for security. An item that has been composed sheltered and executed effectively may at present be powerless because of for setup security.

5 Secure Software Development In Agile

Agile advancement which is essentially iterated software improvement strategy has been the plan for creating software for various organizations amid the previous decade. The fast advancement of software these days requires the rapid programming item conveyance by improvement groups. So as to convey the item speedier, the advancement groups make a change from their customary software improvement lifecycle to light-footed improvement strategy which can empower them towards expedient conveyance of software adapting to the prerequisites change marvel. As agile advancement concentrates on opportune conveyance of software, the security issues and issues identified with that product marvel is not really considered. These issues can make software items extraordinarily defenceless and inclined to malignant assaults by people.

5.1 Secure Extreme Programming (SecureXP)

The concentrate on distinguishing the security related practices of XP. Keeping in mind the end goal to set up satisfactory level of security inside of a framework, each XP part needs to receive security centre practices as needs to decreasing the dangers or vulnerabilities.

There are five fundamental parts in XP, i.e., Managers, mentor, client, designer, and the analyser. Likewise, we present another part "Security Maser". Each of the parts has his/her own particular security centre to verify that the product is created in secure way.

Including some security components inside programming is not viable if there is no expert individual in the improvement of secure programming. To verify the security issues are altered in fitting way, we included another part called "Security Master" to give advices and lead different parts about security details. Here, our examination concentrates on the action of Security Master amid improvement [7].

As said before that we present Security Master as another part in XP who is master in security. This part gives a great deal of points of interest to XP group who needs to create secure programming utilizing XP hones. In XP, the part of security expert can be critical to give preparing to colleague, and sharing the data about sorts of assaults in diverse sorts of programming. Taking into account there are ten XP rehearses which are arranging amusement, illustration, coding standard, straightforward outline, little discharge, ceaseless incorporation, pair programming, aggregate code proprietorship, 40-hours every week and refactoring that are identified with security expert.

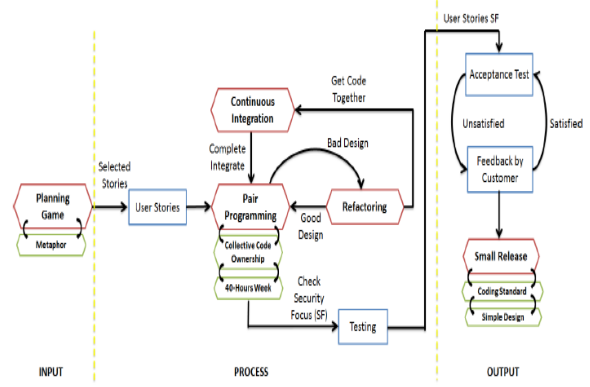


Fig. 2. Extreme Programming Incorporating Security

5.2 Secure Scrum (S-Scrum)

Scrum disregards exact documentation of improvement exercises to build the advancement speed. Regardless this methodology contrarily influences the nature of the web administrations through consolidating imprecision and absence of tractability into the advancement process. Then again security as a quality property has dependably been a standout amongst the most essential concerns of the web administration improvement. To administer to security of the web administration we generally need to consolidate security investigation and outline into the improvement life cycle. In spite of the fact that there have been a few endeavours to watch over examination exercises inside of the Scrum emphases, it is not clear yet how to accomplish this naturally through the Scrum forms. On alternate words watchful building of security into the general framework investigation and configuration is regularly dismissed. In this paper we propose a security-improved form of scrum i.e. Secure Scrum (S-Scrum) to oblige security examination and configuration exercises inside of the Scrum.

To guarantee security of the web administrations it is obliged to introduce stuffiest reports demonstrating fulfilment of security goals. Security certification is not legitimately upheld in Scrum because of the neglecting documentation of security contemplations and examination results. Dismissing the security documentation or de-underscoring on documentation initiates, is tranquil dangerous in the setting of creating security basic framework [8].

The significance of security is more substantial in terms of critical web administrations uncovered all around over the web. Scrum as a light-footed technique takes into account quick advancement of web administrations. Scrum likewise takes into consideration staying up to date with the changing prerequisites amid the product improvement. Transformative nature of Scrum process likewise encourages incremental advancement of web administrations taking into account client prerequisites and business sector request.

The agile method resulted in rapid development, quick response to customer changing requirements, high degree of customer satisfaction and flexibility. But certain agile methodologies like Scrum, XP and DSDM did not include security element in them. This exclusion resulted in vulnerable software [9].

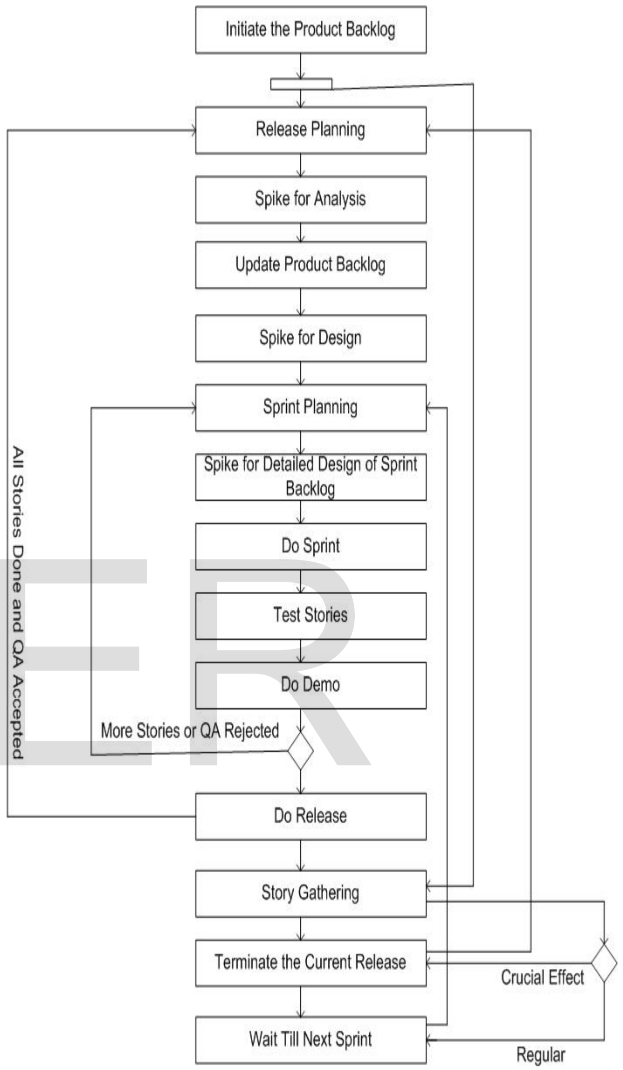


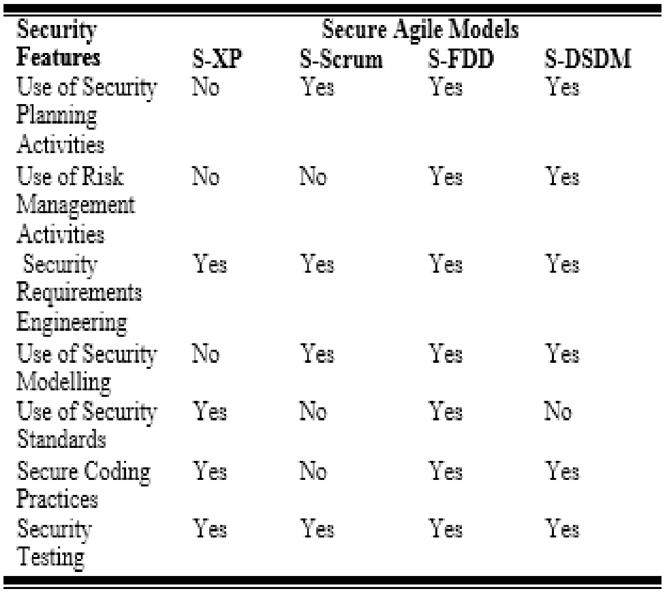
Fig. 3. Security in Scrum

5.3 Secure Feature Driven Development (SFDD)

Conventional agile software improvement practices considered that the genuine opponent to programming advancement procedure is many-sided quality and size of the product. Then again, various late studies showed a more discriminating component disregarded by spry systems, i.e., software creating software productively yet not safely has huge effects, for example, loss of information, loss of notoriety, loss of clients' certainty etc. In this manner, create secure programming in effective way, is a rising issue.

Keeping in mind the end goal to address this issue the current spry strategies should be returned to and improves so that they

TABLE 1
COMPARISON OF AGILE METHODS



Security Features	Secure Agile Models			
	S-XP	S-Scrum	S-FDD	S-DSDM
Use of Security Planning Activities	No	Yes	Yes	Yes
Use of Risk Management Activities	No	No	Yes	Yes
Security Requirements Engineering	Yes	Yes	Yes	Yes
Use of Security Modelling	No	Yes	Yes	Yes
Use of Security Standards	Yes	No	Yes	No
Secure Coding Practices	Yes	No	Yes	Yes
Security Testing	Yes	Yes	Yes	Yes

could give new stages, sub-stages, practices and parts identified with secure programming advancement. As of late, we upgraded Scrum show that could backing secure programming improvement. In the coherence of our examination, this paper concentrates on one of the coordinated advancement procedure, called Feature Driven Development [10].

Not at all like the In-Phase security, has the security viewpoint been characterized after a stage, called After-Phase security. For this, we proposed two new stages, Build Security by Feature and Test Security by Feature. These stages are entirely suitable for experienced and in addition new and less experienced group to create and test secure programming. Taking into account these alterations, there are six stages out and out that begin with Develop an Overall Model, Build and Design Features, Build Security by Features, Plan by Features and Build by Features and Test Security by Features.

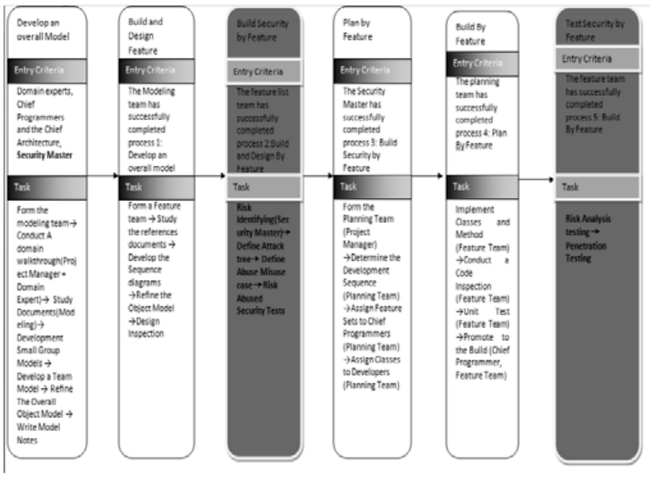

Develop an overall Model
Entry Criteria
Domain experts, Chief Programmers and the Chief Architecture, Security Master
Task
Form the modeling team → Conduct A domain walkthrough(Project Manager + Domain Expert) → Study Documents(Modeling) → Development Small Group Models → Develop a Team Model → Refine The Overall Object Model → Write Model Notes
Build and Design Feature
Entry Criteria
The Modeling team has successfully completed process 1: Develop an overall model
Task
Form a Feature team → Study the references documents → Develop the Sequence diagrams → Refine the Object Model → Design Inspection
Build Security by Feature
Entry Criteria
The feature list team has successfully completed process 2:Build and Design By Feature
Task
Risk Identifying(Security Master) → Define Attack tree → Define Abuse Misuse case → Risk Abused Security Tests
Plan by Feature
Entry Criteria
The Security Master has successfully completed process 3: Build Security by Feature
Task
Form the Planning Team (Project Manager) → Determine the Development Sequence (Planning Team) → Assign Feature Sets to Chief Programmers (Planning Team) → Assign Classes to Developers (Planning Team)
Build By Feature
Entry Criteria
The planning team has successfully completed process 4: Plan By Feature
Task
Implement Classes and Method (Feature Team) → Conduct a Code Inspection (Feature Team) → Unit Test (Feature Team) → Promote to the Build (Chief Programmer, Feature Team)
Test Security by Feature
Entry Criteria
The feature team has successfully completed process 5: Build By Feature
Task
Risk Analysis testing → Penetration Testing

Fig. 4. Secure Feature Driven Development

5.4 Secure Dynamic System Development Method (SDSDM)

Agile models are referred to think working software as the essential measure of accomplishment, and don't pay consideration on security highlights. Clients' association in the task is crucial in DSDM procedure model. Nonetheless, in its present structure, DSDM does exclude any specific part of security partners. In DSDM, there is unlucky deficiency of any stage or stages keeping in mind the end goal to cook the security issues while gathering/dissecting necessities, outlining, or actualizing of software.

In the new SDSDM various changes were made: The first transform we made, present sub-stage in practical model emphasis called distinguish security concerns/issues which will provide food for security exercise. Secondly, another stage was presented called secure utilitarian model emphasis and each of the sub stages security component is fused. Thirdly, secure configuration stage was presented in which security issues are mapped to the stages with the goal that it will be ensured against any vulnerabilities [11].

In two-week iterations differing the objectives and golden triangle for project. The author was using three different agile methods XP, Scrum and DSDM to achieve the success factor and golden angel of the product. Agile team was using four method of objective iteration like Schedule, Teem Satisfaction, Functionality and Quality [12].

DSDM like other agile methods did not present any phase for organization security. Dynamic System Development Method was intended for organization in different changing requirements. It compensated almost no notice to the security requirements. DSDM had purchaser involvement but no security master or security position. Analysis show if software were organism developed by means of DSDM, they would not exist secure.

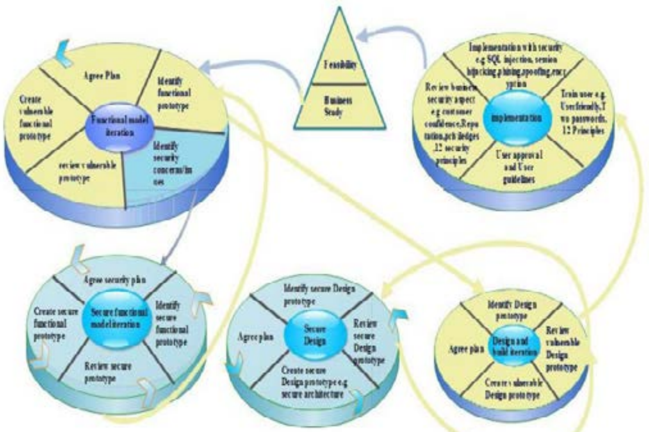

Feasibility
Business Study
Functional model iteration
Agree Plan
Identify functional prototype
Identify security concerns/issues
review vulnerable prototype
Create vulnerable functional prototype
Implementation
Implementation with security e.g SQL injection, session hijacking,phising,spoofing,encryption
Train user e.g. Userfriendly,Two passwords, 12 Principles
User approval and User guidelines
Review business security aspect e.g customer confidence,Reputation,pchledges 12 security principles
Secure functional model iteration
Agree security plan
Identify secure functional prototype
Review secure prototype
Create secure functional prototype
Secure Design
Identify secure Design prototype
Review secure Design prototype
Create secure Design prototype e.g secure architecture
Agree plan
Design and build iteration
Identify Design prototype
Review vulnerable Design prototype
Create vulnerable Design prototype
Agree plan

Fig. 5. Secure Dynamic System Development Method

6 COMPARISON OF VARIOUS SECURE AGILE METHODS

7 Conclusion And Future Work

According to table SFDD (Secure Feature Driven Development) is more complete method as compared to other method for developing secure software. This is because it covers large number of security management activities which are missing in other methods. It provides proper security planning and involves security risk analysis activities. It also considers security as important requirement and thus focuses on security requirement engineering activity. Security Modelling, Secure coding practices, security testing and use of security standards make it distinctive and effective.

Performance of various agile methods for secure software development can be evaluated.

Acknowledgment

I would like to thank my parents, brothers, sisters, teachers and friends for praying, helping and motivating me throughout my educational career.

References

[1] K. Gottipalla, N. M. S. Desai and M. S. Reddy, "Software Development Life Cycle Processes with Secure," *The International Journal of Scientific and Research Publications*, vol. 3, pp. 1-3, 2013.

[2] M. A. Hadavi, V. S. Hamishagi and H. M. Sangchi, " Security Requirement Engineering; State of the Art and Research Challenges," *Proc. Int. Multi Conf. of Engineers and Computer Scientists*, vol. 1, pp. 19-22, 2008.

[3] M. Almseidin, K. Alrfou, N. Alnidami and A. Tarawneh, "A Comparative Study of Agile Methods: XP and Scrum," *International Journal of Computer Science and Software Engineering*, vol.4, no. 5, pp. 126-129, 2015.

[4] G. Bostrom, J. Wayrynen and M. Boden, "Extending XP Practices to Support Requirements Engineering," *International Workshop Software Engineering for Secure Systems*, vol. 1, pp. 11-17, 2006.

[5] P.Salini and S. Kanmani, "Survey and Analysis of Security Requirement Engineering," *The Journal of Computers and Electrical Engineering*, vol. 38, pp. 1785-1797, 2012.

[6] S. V. Shrivastava and U. Rathod, "Risks in distributed agile development A review," *Journal of Social and Behavioral Sciences*, vol. 13, pp. 417 – 424, 2014.

[7] I. Ghani and I. Yasin, "Software Security Engineering in Extreme Programming Methodology: A Systematic Literature Review," *The Journal of Sci.Int.(Lahore)*, vol. 25, pp. 215-22, 2013.

[8] D. Mougouei, N. F. M. Sani and M. M. Almasi, "S-Scrum: a Secure Methodology for Agile Development of Web Services," *The World of Computer Science and Information Technology Journal*, vol. 3, pp. 15-19, 2013.

[9] C. B. Haley, R. Laney, J. D. Moffett and B. Nuseibeh, "Security Requirement Engineering: A Framework for Representation and Analysis," *IEEE Trans. on Software Engineering*, vol. 34, pp. 133-153, 2008.

[10] A. Firdaus, I. Ghani and S. R. Jeong, "Secure Feature Driven Development (SFDD) Model for Secure Software Development," *Int. Conf. on Innovation, Management and Technology Research*, vol. 129, pp. 546-553, 2013.

[11] A. Sani and I. Ghani, "Secure Dynamic System Development Method (SDSDM) Model for Secure Software Development," *Journal of Sci.Int.(Lahore)*, vol.2, pp.1059-1064,2013.

[12] D. G. Firesmith, "Engineering Security Requirements," *The Journal of Object Technology*, vol. 2, pp. 53-68, 2003.