An Integrated Framework for Security Enhancement in Agile Development using Fuzzy Logic

[1]Amit Sharma, [2]Ruchi Sharma, [3]Dr. R.K Bawa

[1]Dept. of Computer Science and Engineering, The ICFAI University, Solan, Himachal Pradesh, India
[2]Dept. of Computer Science and Engineering, Universal Group of Institutions, Lalru, Punjab, India
[3]Dept. of Computer Science, Punjabi University, Patiala, Punjab, India

Abstract

Agile methods are widely employed to develop high-quality software, but theoretical analyses argue that agile methods are inadequate for security-critical projects. However, most agile-developed software today needs to satisfy baseline security requirements, so that we need to focus on how to achieve this level for typical agile projects. Software grows up through its life cycle, so software development methodologies should pay special attention to security aspects of the product. This paper addresses the major concern of security requirements of projects using an agile approach. It provides an integrated framework developed in Java which uses a lightweight method to enhance the security features by integrating security activities from Security engineering processes without compromising the agility in the agile approach.

Keywords

Agile Development, Security Engineering, Scrum, XP, DSDM, FDD, Crystal Clear, Agile Security.

I. Introduction

Agile development follows an opposing approach from the formalization and control in plan-driven development. Instead, agile development formalizes processes only where necessary and emphasizes informal and intensive interaction to craft systems with high business-value. Agile development refers to a set of values shared by related development methods, such as SCRUM and XP. The Agile Manifesto [1] lists the overarching values in abstract terms:

- Individuals and interactions over processes and tools,
- Working software over comprehensive documentation,
- Customer collaboration over contract negotiation,
- Responding to change over following a plan.

In agile development, responsiveness is emphasized over the reliability of standardized development processes. The process is more likened to learning than to the application of prior knowledge. Agile methods can be described as "generative" instead of "adaptive" learning, applying double-loop learning. Also, the "command and control" approach of plan-driven models is exchanged for a more democratic model to profit from the tacit knowledge of the individuals in the team. More specifically than the Agile Manifesto, the principles of agile development are to "deliver something useful," "rely on people," "encourage collaboration," "technical excellence," "do the simplest thing possible," and "be adaptable."

Popular methods in the agile ecosystem have a broad range of characteristics. More liberal methods, such as Crystal Clear and Scrum, formalize the development process to a lesser degree than more heavy-weight methods, such as Feature-Driven Development. Two distinct philosophies can be contrasted here: The liberal methods with an optimistic view that development teams are able to tailor the process to fit their particular development environment in the most efficient way. Conversely, the pessimists rather specify the process in detail to prevent failures from adaptation or problems in large or high-reliability projects.

A. Security in Agile Development

In literature, there is a discussion on whether agile development methods and the underlying principles are appropriate to develop secure software. One reason is that the agile development proponents did explicitly not target high-risk software development. Kent Beck rather states in his XP book that XP in itself is not suitable for high-reliability requirements. However, security is not only relevant for high-reliability projects, but affects most software that is being developed.

The main issue with agile development concerning security is that the team-emphasizing, dynamic and tacit-knowledge-driven methods conflict with the assurance activities as demanded by traditional secure software development methods [22]. However, there are indications that agile development improves quality. Moreover, plan-driven development also poses challenges to secure software development that might be less critical in agile development. Early planning of security requirements may conflict with the changing requirements in practice, which agile development is better prepared for. Also, to address the challenges to security in agile development, various enhancements have been proposed to agile methods [22].

II. Framework for Security Enhancement

This section covers the various steps followed to develop a framework for security enhancement.

A. Integration of Security in Agile Development Method

The growing trend towards the use of agile techniques for building software and the increase in security breaches over the past few years means that it is essential to integrate existing high-profile Security engineering (SE) processes with agile processes. Moreover, as there are no SE processes developed specifically for an agile setting, organizations have used existing waterfall SE processes in their agile processes. However, the reliability of the SE processes commonly used in the waterfall model has not yet been evaluated in an agile development setting. Accordingly, existing security activities within water-fall SE processes used in current agile processes are investigated. Four high-profile waterfall SE processes (CLASP, Microsoft SDL, Cigital Touchpoints and Common Criteria) are investigated [20]. Based on these SE processes, a total of 41 security activities are obtained which are used for further investigation.

Table 1: Security Activities

Security Activities	
Requirement	**Implementation**
Security Metrics (CLASP)	Static Code Analyses (SDL, CT)
Initial Education (CLASP, SDL)	Security Tools (SDL)
Security Requirements (CLASP, SDL, CT, CC)	Coding Rules (SDL)
Abuse Cases (CLASP, CT)	Pair Programming (O)
Agree on Definitions (CC)	
Role Matrix (CLASP, SDL)	**Testing**
Design Requirements (SDL)	Vulnerability & Penetration Testing (CT)
Identify Trust Boundary (CLASP)	Red Team Testing (CT)
Identify Global Security Policy (CLASP)	Risk Based Testing (CT)
Specify Operational Environment (CLASP)	Dynamic Analysis (SDL)
Identify Attack Surface (CLASP)	Fuzzy Testing (SDL)
	Code Review (CLASP, SDL)
Design	Security Testing (CLASP)
Risk Analyses (CT, CC)	
Assumption Documentation (CT)	**Release**
Critical Assets (CC)	External Review (CT)
UMLSec (CC)	Repository Improvement (CC)
Quality Gates (SDL)	Incident Response Planning (SDL
Cost Analysis (SDL)	Signing the Code (CLASP)
Attack Surface Reduction (SDL)	Operational Planning and Readiness (CLASP)
Security Architecture (CLASP)	Final Security Review (SDL)
Secure Design Principles (CLASP)	
Security Activities	
Countermeasure Graphs (O)	
Requirements Inspection (CC)	
Threat Modeling (CLASP, SDL)	

B. Integration of Security Activities with Agile Development Methods

As mentioned in earlier section, there are some guidelines, best practices, methods and other materials in Security engineering (SE) that can be used by project's team to produce secure software products [20]. To arm agile methods with security features, it is acceptable to use these experienced and proposed activities for secure software development. On the other hand, integrating some heavy weight activities with agile processes may lead to a process that cannot be named agile and possibly will be unacceptable for project's team. In order to restrain reduction of agility nature, a proper method has to be used. First security activities are extracted from Security engineering (SE) processes, and then agility degree of activities is defined to measure their nimbleness. Integration issues of agile and security activities are handled and a flowchart to integrate security activities with organization's agile process is introduced.

C. Integration Method

To integrate security activities selected from Security engineering (SE) processes the following steps shown [2] in the flowchart have to be followed. This flowchart provides a method through which security activities can be integrated with agile activities without compromising the agility of the process as shown in fig. 1.

D. Implementation

The above mentioned method [2] is implemented in Java and the different steps mentioned are implemented in different modules as shown in figures. Figure 2 shows the addition of security activities and the same is done in case of agile activities. Fig. 3 shows Fuzzy Integration matrix which decides the membership value of whether a security activity can be integrated with the agile activity and to what extent. Fig. 4 shows the Integration of security activities with agile activities after running the program.

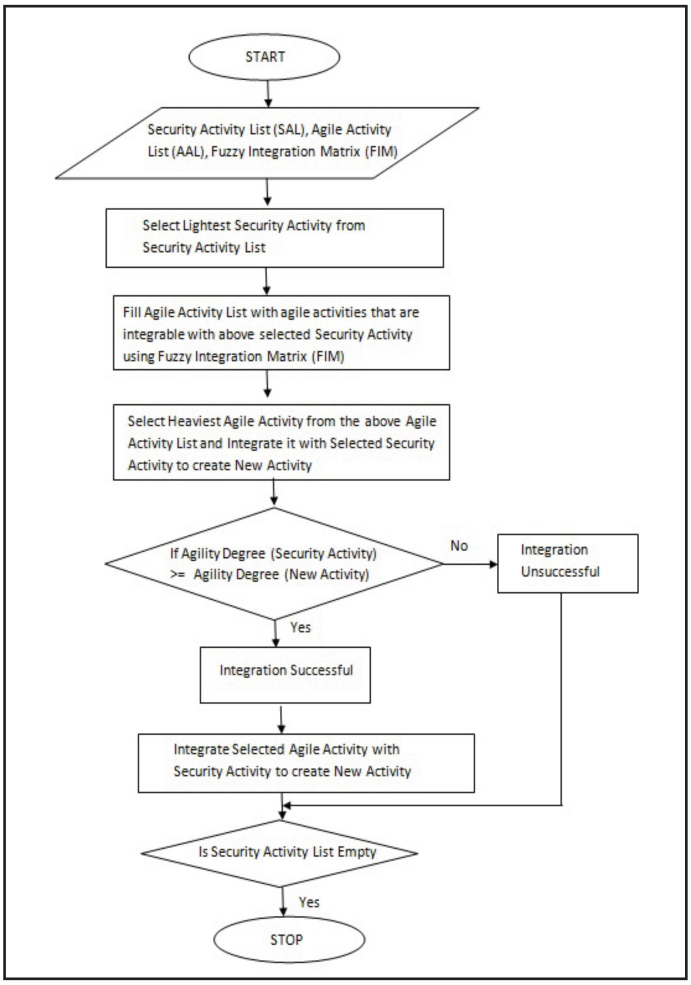
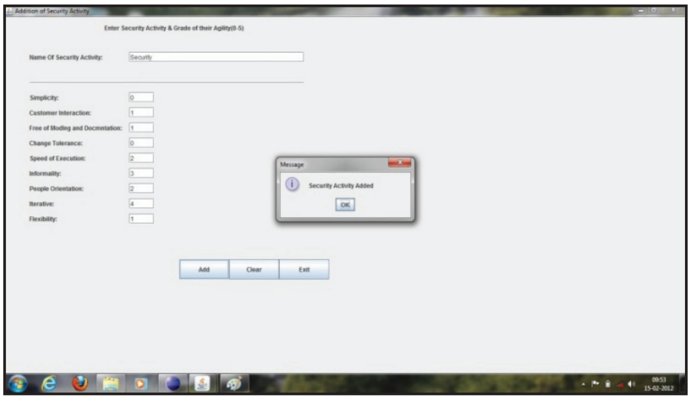

START
Security Activity List (SAL), Agile Activity List (AAL), Fuzzy Integration Matrix (FIM)
Select Lightest Security Activity from Security Activity List
Fill Agile Activity List with agile activities that are integrable with above selected Security Activity using Fuzzy Integration Matrix (FIM)
Select Heaviest Agile Activity from the above Agile Activity List and Integrate it with Selected Security Activity to create New Activity
If Agility Degree (Security Activity) >= Agility Degree (New Activity)
No
Integration Unsuccessful
Yes
Integration Successful
Integrate Selected Agile Activity with Security Activity to create New Activity
Is Security Activity List Empty
Yes
STOP

Fig. 1: Flowchart for Integrating Security Activities



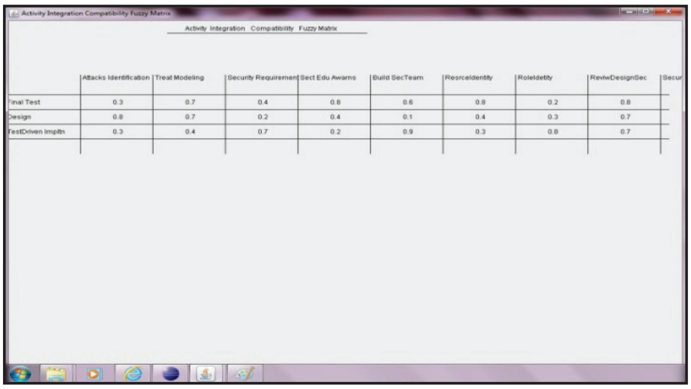
Enter Security Activity & Grade of their Agility(0-5)

Name Of Security Activity: Security

Simplicity: 0
Customer Interaction: 1
Free of Moding and Docmntation: 1
Change Tolerance: 0
Speed of Execution: 2
Informality: 3
People Orientation: 2
Iterative: 4
Flexibility: 1

Message
Security Activity Added

Fig. 2: Addition of Security Activities

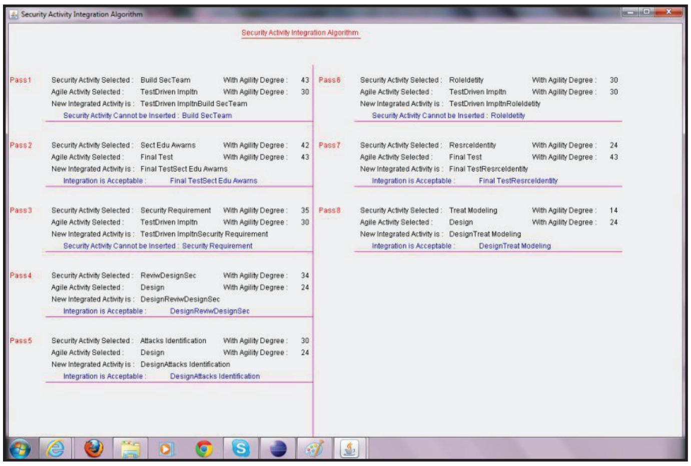


Activity Integration Compatibility Fuzzy Matrix

Activity Integration Compatibility Fuzzy Matrix

	Attacks Identification	Treat Modeling	Security Requirement	Sect Edu Awarns	Build SecTeam	ResrceIdentity	RoleIdetity	RevIwDesignSec	Secur
Final Test	0.3	0.7	0.4	0.8	0.6	0.8	0.2	0.8	
Design	0.8	0.7	0.2	0.4	0.1	0.4	0.3	0.7	
TestDriven Impltn	0.3	0.4	0.7	0.2	0.9	0.3	0.8	0.7	

Fig. 3: Fuzzy Integration Matrix



Security Activity Integration Algorithm

Security Activity Integration Algorithm

```
Pass1   Security Activity Selected :   Build SecTeam            With Agility Degree :   43
        Agile Activity Selected :      TestDriven Impltn        With Agility Degree :   30
        New Integrated Activity is :   TestDriven ImpltnBuild SecTeam
            Security Activity Cannot be Inserted :  Build SecTeam

Pass2   Security Activity Selected :   Sect Edu Awarns          With Agility Degree :   42
        Agile Activity Selected :      Final Test               With Agility Degree :   43
        New Integrated Activity is :   Final TestSect Edu Awarns
            Integration is Acceptable :        Final TestSect Edu Awarns

Pass3   Security Activity Selected :   Security Requirement     With Agility Degree :   35
        Agile Activity Selected :      TestDriven Impltn        With Agility Degree :   30
        New Integrated Activity is :   TestDriven ImpltnSecurity Requirement
            Security Activity Cannot be Inserted :  Security Requirement

Pass4   Security Activity Selected :   ReviwDesignSec           With Agility Degree :   34
        Agile Activity Selected :      Design                   With Agility Degree :   24
        New Integrated Activity is :   DesignReviwDesignSec
            Integration is Acceptable :        DesignReviwDesignSec

Pass5   Security Activity Selected :   Attacks Identification   With Agility Degree :   30
        Agile Activity Selected :      Design                   With Agility Degree :   24
        New Integrated Activity is :   DesignAttacks Identification
            Integration is Acceptable :        DesignAttacks Identification

Pass6   Security Activity Selected :   RoleIdetity              With Agility Degree :   30
        Agile Activity Selected :      TestDriven Impltn        With Agility Degree :   30
        New Integrated Activity is :   TestDriven ImpltnRoleIdetity
            Security Activity Cannot be Inserted :  RoleIdetity

Pass7   Security Activity Selected :   ResrceIdentity           With Agility Degree :   24
        Agile Activity Selected :      Final Test               With Agility Degree :   43
        New Integrated Activity is :   Final TestResrceIdentity
            Integration is Acceptable :        Final TestResrceIdentity

Pass8   Security Activity Selected :   Treat Modeling           With Agility Degree :   14
        Agile Activity Selected :      Design                   With Agility Degree :   24
        New Integrated Activity is :   DesignTreat Modeling
            Integration is Acceptable :        DesignTreat Modeling
```

Fig. 4: Integration of Security Activities With Agile Activities

III. Conclusion

This work provides a preliminary roadmap that serves as a starting point for creating a secure agile development approach and enables the generation of more fruitful research results from the field. Using introduced method in this paper, security activities can be integrated to agile methodologies to enhance security of software product without compromising the overall agility of the project.

In addition, since the selected security activities are originally developed for waterfall development approach, some of the security activities might need modification in order to adapt with an agile process. We have not investigated new or pure agile SE-processes (but a selection of existing/modified security activities as a base for the agile development). Therefore, the directions for future work primarily include evaluating these security activities that are selected as compatible and beneficial to an agile model in a real agile industry setting. These steps will add value to the findings and gain acceptance in the real agile industry.

References

[1] Beck K., et al.,"Manifesto for Agile Software Development", February 2001.

[2] Sharma A., Bawa R.K,"A Comprehensive approach for Agile Development Method Selection and Security Enhancement", International Journal of Innovations in Engineering and Technology, Vol. 6, Issue 4, pp. 36-44, 2016.

[3] Blitz, D.C., van Vliet, P.,"Global Tactical Cross-Asset Allocation", Journal of Portfolio Management, Vol. 35, No. 1, pp. 23-38, 2008.

[4] Boehm, B.,"A Spiral Model of Software Development and Enhancement", Journal: Computer, Vol. 21, No. 5, pp. 61-72, 1988.

[5] Beck, Kent, Andres, Cynthia,"Extreme Programiming: Embrace Change (2nd ed.)", Addison Wesley Professional, Boston, 2004.

[6] N. Ramasubbu, R. K. Balan,"Globally distributed software development project performance: An empirical analysis", in Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering - ESEC-FSE 07, 2007, pp. 125.

[7] Concas, G., Francesco, M., Marchesi, M., Quaresima, R., Pinna, S.,"An agile development process and its assessment using quantitative object-oriented metrics", Agile Processes in Software Engineering and Extreme Programming, pp. 83- 93, 2008.

[8] Ramasubbu, N., Balan, R.K.,"The impact of process choice in high maturity environments: An empirical analysis", In the proceedings of the 31st IEEE International Conferences on Software Engineering (ICSE 2009), Vancouver, British Columbia, Canada. pp. 529–539, May 16–24, 2009.

[9] Begel , A., Nagappan, N.,"Usage and perceptions of agile software development in an industrial context: An exploratory study", In ESEM '07: First International Symposium on Empirical Software Engineering and Measurement, pp. 255–264. Washington, DC: IEEE, 2007.

[10] Parsons, D., H. Ryu, R. Lal,"The Impact of Methods and Techniques on Outcomes from Agile Software Development Projects", IFIP International Federation for Information Processing, Springer Boston: pp. 235- 249, 2007.

[11] Fitzgerald, B., Harnett, G., Conboy, K.,"Customizing Agile Methods to SoftwarePractices", European Journal of Information Systems, Vol. 15, No. 2, 2006.

[12] Kniberg, H., Skarin, M.,"Kanban and Scrum - making the most of both," C4Media Inc., USA, 2010.

[13] Highsmith, J.,"Agile software development ecosystems", Boston, M.A., Pearson Education, 2002.

[14] Ken Schwaber, Mike Beedle,"Agile Software Development with Scrum (Prentice Hall, 2001)".

[15] Poppendieck, M, Poppendieck T,"Lean Software Development An Agile Toolkit", Boston: Addison Wesley, 2003.

[16] J. Stapleton,"DSDM: The Method in Practice", Second ed: Addison Wesley Longman, 2003.

[17] S. R. Palmer, J. M. Felsing,"A Practical Guide to Feature-Driven Development", Upper Saddle River, NJ: Prentice Hall PTR, 2002.

[18] Abrahamsson, P., Warsta, J., Siponen, M., Ronkainen, J., "New directions in agile methods: Comparative analysis". In Proceedings of the 25th International Conference on Software Engineering, 244–254, 2003.

[19] Keramati, H., Hassan, S., Hosseinabadi, M.,"Integrating software development security activities with agile methodologies", pp. 749-754, IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2008, March 31-April 4 2008.

[20] Baca D., Carlsson B.,"Agile Development with Security Engineering Activities", ACM International Conference on Software Engineering ICSE '11, May 21–28, 2011.

[21] Nasr-Azadani B., Mohammad Doost R.,"Estimation of Agile Functionality in Software Development",Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008.

[22] Sharma A., Sharma R., "A Systematic Review of Agile Software Development Methodologies ", NCIDEM-2015 National Conference on Innovation and Developments in Engineering and Management, April 28, 2015.

Amit Sharma received his B.Tech degree in Computer Science and Engineering from Kurukshetra University in 2004, the M.Tech degree in Computer Science and Engineering from Punjabi University, Patiala, India, in 2008 and pursuing Ph.D. degree in Agile Security from Punjabi University, Patiala, India. He is working as faculty in The ICFAI University, Himachal Pradesh, India. His research interests include Agile development, software engineering, security engineering. At present, He is engaged in security enhancement in agile development.