ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Empirical Approach for Security Enhancement in Agile Development at Process Level using Fuzzy Logic

Amit Sharma
Department of Computer Engineering
Punjabi University, Patiala,
Punjab, India

Dr. R.K Bawa
Department of Computer Science
Punjabi University, Patiala,
Punjab, India

Abstract: In recent times agile software development has evolved world widely with a huge impact. The major advantage of agile development is a faster development time with flexible structure that helps handling rapid changes to new requirements easily than the more rigid older processes. Although the agile approach is becoming popular, but it is reported to have certain disadvantages related to secure software development. For building a secure software, there is a need for security enhanced processes and practices. The growing trend towards using agile methods for developing the software and increase in security breaches over the last few years has concluded that it becomes essential to integrate existing high profile security engineering processes with agile processes. This paper is addressing this huge concern of security requirements for projects using agile development approach. It provides a framework to introduce security at process level with the help of fuzzy logic. This framework is implemented in Java language with user friendly graphical user interface. It uses a lightweight approach to add the security features by integrating security activities from Security engineering processes that too without compromising the agility of agile process.

Keywords: Agile Development, Security Engineering, Agile Security, Fuzzy logic.

I. INTRODUCTION

A. Agile Development

Agile development process is based upon iterative and incremental software development approach. Close customer interaction and collaboration is the key through which the actual product evolves, starting from the first step of initial requirements till the finished product and the customer is made satisfied by providing him deliveries of the working software time to time which can be used to conform to his requirements. The term agile development can be completely described by the agile manifesto as well as the twelve principles of agile development [1]. Agile development is not at all based upon the hard core principles of plan driven approach which are based upon intensive control and formalization, rather opposing to this it follows an approach where the focus is on maximizing the productivity and profit by fast release and simplified documentation. The word agile has several meanings on the basis of varied practices, which are basically a collection of well defined methods which also further differs in their implementation. Alistair Cockburn, the pioneer of agile methodology has defined agile as “Being agile means flexibility and effective. It is both abundant and light. The lightness means, being flexible and abundant is a matter of staying in the game”.

Since the beginning of 1980s, several agile development methods have come into existence and this process of evolution has not ceased so far [2]. These popular methods which are being widely used by different organizations worldwide are Scrum [3] [4], Extreme Programming (XP) [5], Crystal Clear, Feature-driven Development (FDD) [6], Dynamic Software Development Method (DSDM) [7] and Lean development [8]. These methods in their roots differ in practices and principles, and the detailed elaboration of these agile development methods is out of the scope of this paper [9].

B. Fuzzy Logic

Fuzzy logic was introduced by Lotfi A. Zadeh [10] and it is actually an extension to Boolean logic and is based on the mathematical theory of fuzzy sets, which in turn is actually a generalization of the classical set theory. It introduces the notion of degree in the verification of a condition, thus putting a condition to be in a state other than just true or false. Fuzzy logic thus provides a very valuable and useful flexibility for reasoning, which provides the mechanism to take into account uncertainties and inaccuracies. The most obvious limiting feature of bivalent sets that can be observed clearly is that they are mutually exclusive as it is not possible to have membership of more than one set for example opinion would widely vary as to whether 50 degrees Fahrenheit is 'cold' or 'cool' and hence we need to define our system mathematically at odds with the humanistic world. Clearly, it is not precise and accurate to define a transition from a quantity such as 'warm' to 'hot' by merely applying one degree Fahrenheit of heat. In the realistic world a smooth drift which is unnoticeable from warm to hot would occur. Such natural phenomenon can be described more precisely by Fuzzy Set Theory.

C. Security in agile development

In literature, there is a major concern on whether agile development methods with their underlying principles are appropriate to develop secure software [11] [12]. One reason could be that the agile development proponents did deliberately not target high risk software development. Kent Beck rather stated that XP in itself is not suitable for high reliability requirements. However, security is not only a concern for only high reliability projects, but it also affects most software that is being developed.

The main challenge with agile development concerning security is that the more team emphasizing, dynamic and implied knowledge driven methods usually conflicts with the assurance activities as implemented by traditional secure software development methods [13]. However, there are also indications that agile development inherently improves quality [14] [15]. Moreover, the plan driven development also poses implicit challenges to secure software development that might be less critical in agile development [16] [17]. Early upfront planning of security requirements may conflict with the dynamic and changing requirements in practice, which agile development is rather better prepared for. Also, to address the major challenges to security in agile development, various changes and enhancements have been proposed to agile methods [18] [19].

II. SECURITY AT PROCESS LEVEL

A. Extraction of Security Activities

The increasing trend towards the use of agile methods for developing software and in turn the increase in security breaches over the past years indicates that it is need of the hour to integrate existing high profile Security engineering (SE) processes with agile processes. However, as there are no security engineering processes developed specifically for agile methods, thus organizations have used existing waterfall security engineering processes in their agile processes. But the reliability and suitability of the security engineering processes commonly used in the waterfall model have not yet been tested in an agile development setting. Thus, for current approach existing security activities within plan driven security engineering processes used in current agile processes are investigated. We have investigated four high profile waterfall security engineering processes which are Microsoft SDL, CLASP, Common Criteria and Cigital Touchpoints. Based on these security engineering processes, the following security activities are obtained which are used for further investigation shown in Table I.

Table I. **Agile compatible and beneficial security activities.**

Pre-Requirement	**Requirement**
Initial Education	Security Requirements
Design	Agree on Definitions
Risk Analyses	Role Matrix
Quality Gates	Identify Trust Boundary
Secure Design Principles	Specify Operational Environment
Counter Measure Graphs	**Implementation**
Testing	Security Tools
Vulnerability & Penetration Testing	Coding Rules
Security Testing	**Release**
	Signing the Code
	Operational Planning and Readiness

Below is given the definitions of these 15 security activities:

-Initial Education: The basic requirement of any development project is that everyone should be aware of the importance of security and the basics of security engineering which includes; types of security breaches, teaching the security concepts, possible solutions etc.

-Security Requirements: This includes assigning security experts, identifying and enumerating privacy and security functionality for a software process.

-Agree on definitions: The first and the foremost task for any organization are to define the stakeholders and thereafter to agree on a common set of security definitions, which includes the defining the security policies of the software company with the clients as a part of the stakeholders' security vision.

-Role Matrix: This includes identification of all the possible user roles and level of their access to the software.

-Identify Trust Boundary: Describing the architecture of the system from the network perspective and identifying data resources that may be used by a program and also denote where the trustworthy and untrustworthy entities could interact.

-Specify Operational Environment: This includes documenting assumptions and all the requirements about the operating environment, so that its impact on the security can be assessed.

-Risk Analyses: It includes finding and prioritizing the architectural flaws by security analysts so that appropriate mitigations can begin.

-Quality Gates: Creating appropriate privacy and security quality measures for the overall software development project, including even those activities that need to be performed for a fulfillment of the requirement.

-Security Design Principles: This includes making the application design robust and harder by applying security design principles. It also includes identifying security risks in third party components.

-Countermeasure Graphs: It is a risk analyses method which focuses on identifying security features and then prioritizing them.

-Security Tools: It defines and publishes a list of approved and verified security tools to assist the project, this includes commercially available, in-house developed, open source and associated security checks.

-Coding Rules: This defines the determination of the list of all unsafe functions and replacing those unsafe functions with safer and tested alternatives.

-Vulnerability & Penetration Testing: This type of testing provides a good understanding of the software in its real environment. This task is done by simulating real world working conditions and the associated attack patterns.

-Security Testing: This is done to find security problems which are not found by implementation review and catching failures in specification, design and implementation.

-Signing the Code: This provide the stakeholder with a formal way to validate the origin and to maintain the integrity of the software.

-Operational Planning and Readiness: This includes the documenting the security architecture, writing of user manuals and so on.

B. Fuzzy Integration of Security Activities

As mentioned in earlier also that there are some guidelines, best practices, methods and other techniques in Security engineering that can be used by the organization to produce secure software products [20]. To integrate agile methods with security features, it is quite acceptable to use these experienced and tested activities for secure software development. On the other hand, the major concern is that by integrating these heavy weight activities with light agile processes may lead to a process that may not be agile and possibly will be unacceptable. In order to overcome this reduction of agility nature, a proper empirical method has to be used. At the first step security activities are extracted from Security engineering processes as discussed in previous section and then agility degree of activities is defined to measure their lightness. Integration of the agile and security activities is handled very carefully and the flowchart shown in figure demonstrates the process to integrate security activities with organization's agile process. The proposed method is also using linguistic variables to show the compatibility of security activity with the agile activity that might be medium, might be moderately low, low etc. The linguistic variable used are extremely low, low, moderately low, medium, moderately high, high, extremely high commonly known as fuzzy logic shown in Table II. . By using these fuzzy logic values the subjective behavior can be compensated which is the basis for the approximate reasoning [21]. These fuzzy vales provide more realistic approach for human reasoning than the binary values. The observations based upon the input of five security experts for the integration of security with agile activities are presented in the Table III. .

Table II. **Fuzzy Numbers representation for Linguistic Variable**

Linguistic Variable	**Fuzzy Number**
Very Low(VL)	(0,0.05,0.15)
Low(L)	(0.1,0.2,0.3)
Fairly Low(FL)	(0.2,0.35,0.5)
Medium(M)	(0.3,0.5,0.7)
Fairly High(FH)	(0.5,0.65,0.8)
High(H)	(0.7,0.8,0.9)
Very High(VH)	(0.85,0.95,1.0)

Agile Activity	Security Activity	Security Expert				
Planning		SE1	SE2	SE3	SE4	SE5
	Initial Education	FH	FH	FH	FH	H
	Security Requirement	H	H	H	H	H
	Identify Trust Boundary	H	VH	H	H	H
	Role Matrix	H	H	H	H	VH
	Risk Analysis	H	VH	VH	VH	H
	Threat Modeling	M	FH	FH	H	H
	Static code Analysis	L	L	L	L	L
	Coding Rules	L	VL	VL	VL	L
	Security Testing	M	M	M	M	M
	Vulnerability Testing	M	FH	M	M	M
	Operation Planning	FH	H	H	H	FH

Table III. **Observation on the basis of Linguistic Variable for Agile Activity Planning**

III. INTEGRATION METHOD

To integrate security activities different approaches have been used by using comparison [22] or by calculating the functionality [23]but here we are integrating security activities which are selected from security engineering processes through the steps shown in the flowchart shown in Figure 1. This flowchart provides a series of steps through which security activities can be integrated with agile activities that too without compromising the agility of the process [24]. The overall approach is divided in following steps:

- Decision Making using Analytic Hierarchy Process (AHP) and Artificial Neural Network (ANN) by calculating Agility Indicator [25] [26].
- Selection of most appropriate Agile Development method.
- Extraction of agile characteristics.
- Extraction of security activities from Security Engineering.
- Extraction of agile activities.
- Calculation of MAV for the security activities and agile activities.
- Formation of Fuzzy Value Compatibility Table.
- Calculation of Influencing Factor Value.
- Algorithm for the integration of security activities with agile methodology.
- Design of framework for the integration.

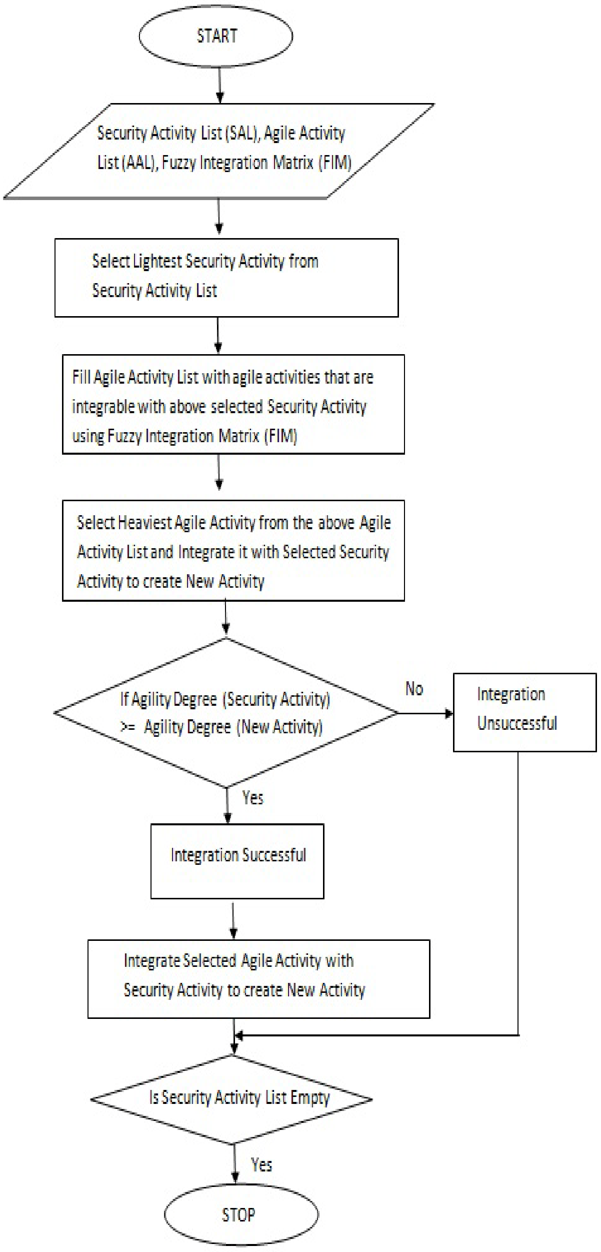
START
Security Activity List (SAL), Agile Activity List (AAL), Fuzzy Integration Matrix (FIM)
Select Lightest Security Activity from Security Activity List
Fill Agile Activity List with agile activities that are integrable with above selected Security Activity using Fuzzy Integration Matrix (FIM)
Select 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

Figure 1. **Flow Chart Showing the steps involved in integration of Security with agile Activities**

IV. Implementation

The above described algorithm is implemented in Java and the output is plotted in Figure 2. , which shows the extent of the compatibility of agile activities with Security activities.

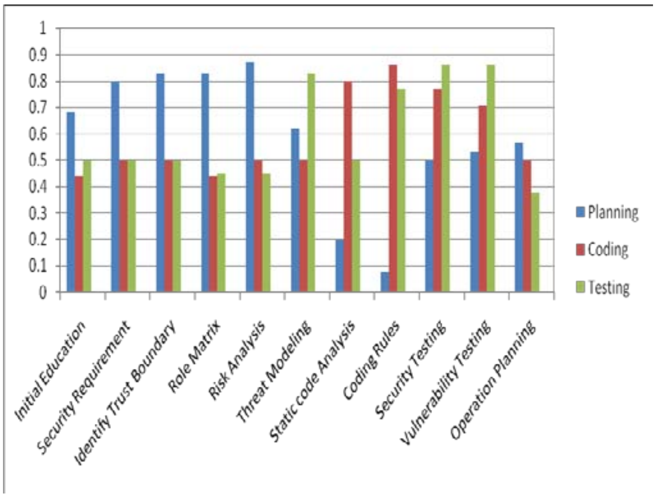

Figure 2. **Compatibility of agile activities with Security activities**

V. CONCLUSION

This work provides an initial roadmap that would serve as a starting point for developing a secure agile development approach and would enable the regeneration of more fruitful research results from this field. Using the proposed method, security activities can be integrated to agile processes using fuzzy logic to enhance the security of software product without compromising the overall lightness of the project.

In addition, as the selected security activities are basically developed for waterfall development approach in original, thus some of the security activities might need further modification in order to adapt with an agile process. We have not investigated into new agile security engineering processes because of lack of published research work. Therefore, the directions for the future work primarily would include evaluating these security activities which are selected as compatible to an agile process in a real agile industry setting. This study will add value to the current findings and will gain acceptance in the real agile industry.

VI. REFERENCES

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

[2] Highsmith, J. “Agile software development ecosystems”, Boston, M.A., Pearson Education, 2002.

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

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

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

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

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

[8] Poppendieck, M and Poppendieck T “Lean Software Development An Agile Toolkit” Boston: Addison Wesley, 2003.

[9] Sharma, A. and Sharma, R. “A Systematic Review of Agile Software Development Methodologies” In Proceedings of the

National Conference on Innovation and Developments in Engineering and Management, 2015.

[10] L.A. Zadeh, Fuzzy Sets, Information and Control, 1965.

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

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

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

[14] N. Ramasubbu and 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, p. 125.

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

[16] 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.

[17] Begel , A. , and 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, 255–264. Washington, DC: IEEE, 2007 .

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

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

[20] Sharma, A. "A Comprehensive approach for Agile Development Method Selection and Security Enhancement",. In Proceedings of the International Journal of Innovations in Engineering and Technology, Vol. 6, pp 36-44, 2016.

[21] Sharma, A. " An Integrated Framework for Security Enhancement in Agile Development using Fuzzy Logic", In Proceedings of the International Journal of Computer Science And Technology, pp 150-153, Vol. 7 (2016).

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

[23] Nasr-Azadani B., MohammadDoost 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.

[24] Sharma, A. and Bawa, R.K.: An Empirical Approach to Measure Agility for Secure Agile Development using Fuzzy Analytic Hierarchy Process and Artificial Neural Network. In Proceedings of the International Journal of Control Theory and Applications, pp 9283-9290, Vol. 9 (2016).

[25] Sharma, A. and Bawa, R.K.: A Framework for Agile Development Method Selection using Modified PROMETHEE with Analytic Hierarchy Process. In Proceedings of the International Journal of Computer Science and Information Security, pp 846-854, Vol. 14 (2016).

[26] Sharma, A. and Bawa, R.K.: A Roadmap for Agility Estimation and Method Selection for Secure Agile Development using AHP and ANN. In lecture notes of the book Data Engineering And Intelligent Computing, Springer, Chapter No 22 (2017).