Implementation of Fuzzy Logic for Introducing Security at Process Level in Agile Development

Ruchi Sharma
Department of Computer Science and Engineering
Universal Group of Institutions, Lalru, Punjab, India

Priyanka Mehta
Department of Computer Science and Engineering
Universal Group of Institutions, Lalru, Punjab, India

Abstract- Agile software development has had a huge impact on how software has evolved worldwide recently. The key aspect of the agile development is a flexible structure with faster development time that allows handling changes to new requirements easier than the older more rigid processes. Even though the agile approach is becoming popular, it is reported to have disadvantages related to secure software development. In order to build secure software, security enhanced processes and practices are needed. 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 SE processes with agile processes. This paper addresses this major concern of security requirements of projects using agile approach. It provides a roadmap to introduce security at process level using fuzzy logic. It is implemented in Java language with graphical user interface. It uses a lightweight method to enhance the security features by integrating security activities from Security engineering processes without compromising the agility of agile process.

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

I. INTRODUCTION

A. Agile development

Agile – denoting “the quality of being agile; readiness for motion; nimbleness, activity, dexterity in motion” (http://dictionary.oed.com) – software development methods attempt to offer once again an answer to the eager business community asking for lighter weight along with faster and nimbler software development processes. This is especially the case with the rapidly growing and volatile Internet software industry as well as with the emerging mobile application environment. The new agile methods have evoked a substantial amount of literature and debates. The Agile Manifesto gathered representatives from Extreme Programming (XP), Dynamics Systems Development Methods (DSDM), Adaptive Software Development (ASD), Scrum, Crystal Methods, Feature-Driven Development (FDD), and others who saw the need for an alternative to documentation driven, heavyweight Traditional software development processes [21].

The manifesto reads as follows (Agile Alliance, 2001): “We are uncovering better ways of developing software by doing it and helping others does it. Through this work we have come to value [14]

- Individuals and interactions over Processes and tool
- 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.”

B. Fuzzy Logic

Fuzzy logic is an extension of Boolean logic by Lotfi Zadeh in 1965 based on the mathematical theory of fuzzy sets, which is a generalization of the classical set theory. By introducing the notion of degree in the verification of a condition, thus enabling a condition to be in a state other than true or false, fuzzy logic provides a very valuable flexibility for reasoning, which makes it possible to take into account inaccuracies and uncertainties. The most obvious limiting feature of bivalent sets that can be seen clearly from the diagram is that they are mutually exclusive - it is not possible to have membership of more than one set (opinion would widely vary as to whether 50 degrees Fahrenheit is 'cold' or 'cool' hence the expert knowledge we need to define our system is mathematically at odds with the humanistic world). Clearly, it is not accurate to define a transition from a quantity such as 'warm' to 'hot' by the application of one degree Fahrenheit of heat. In the real world a smooth (unnoticeable) drift from warm to hot would occur.

This natural phenomenon can be described more accurately by Fuzzy Set Theory.

C. 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. 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.

II. SECURITY AT PROCESS LEVEL

A. Extraction of Security Activities

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 [22]. Based on these SE processes, the following security activities are obtained which are used for further investigation.

Table 1.Agile compatible and beneficial security activities.

Pre-Requirement (PRq)	**Requirement (Rq)**
Initial Education (CLASP, SDL)	Security Requirements (CLASP, SDL, CT, CC)
Design (D)	Agree on Definitions (CC)
Risk Analyses (CT, CC)	Role Matrix (CLASP, SDL)
Quality Gates (SDL)	Identify Trust Boundary (CLASP)

Secure Design Principles (CLASP)	Specify Operational Environment (CLASP)
Counter Measure Graphs (O)	**Implementation (I)**
Testing (T)	Security Tools (SDL)
Vulnerability & Penetration Testing (CT)	Coding Rules (SDL)
Security Testing (CLASP)	**Release (R)**
	Signing the Code (CLASP)
	Operational Planning and Readiness (CLASP)

Below the definitions of the 16 security activities are presented:

- Initial Education (PRq): Everyone on a development project should be aware of the importance of security and the basics of SE which includes; teaching the se- curity concepts, types of security breaches, possible solutions and so on.

- Security Requirements (Rq): Assign security experts, identify and enumerating security and privacy functionality for a given software process.

- Agree on definitions (Rq): The first task for an organization is to define the stakeholders and to agree upon a common set of security definitions, i.e. the def- inition of the security policies for a software company with the clients as part of the stakeholders' security vision of the IS.

- Role Matrix (Rq): Identifying all possible user roles and their access level to the software.

- Identify Trust Boundary (Rq): Describe the architecture of the system from the perspective of the network, identify data resources that may be used by a pro- gram and denote where trustworthy and untrustworthy entities interact.

- Specify Operational Environment (Rq): Document assumptions and require- ments about the operating environment, so that the impact on security can be as- sessed.

- Risk Analyses (D): Security analysts find and prioritize architectural flaws so that appropriate mitigations can begin.

- Quality Gates (D): Create appropriate security and privacy quality measures for the entire software development project, including activities that need to be done for a fulfillment of the requirement.

- Security Design Principles (D): Make the application design harder by applying security design principles and identify security risks in third-party components.

- Countermeasure Graphs (D): A risk analyses method that focuses on identifying security features and prioritizing them.

- Security Tools (I): Define and publish a list of approved security tools to assist the project, i.e. commercially available, open source and in-house developed,

  and associated security checks.

- Coding Rules (I): Determine the list of unsafe functions and replace those unsafe functions with safer alternatives.

- Vulnerability & Penetration Testing (T): Provides a good understanding of field- ed software in its real environment. This is done by simulating real world work- ing conditions and attack patterns.

- Security Testing (T): Find security problems not found by implementation re- view and catching failures in design, specification and implementation.
- Signing the Code (R): Provide the stakeholder with a way to validate the origin and the integrity of the software.
- Operational Planning and Readiness (R): This includes the writing of user man- uals, documenting the security architecture and so on.

B. Fuzzy Integration of Security Activities

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 [17]. 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. The Proposed approach is also using linguistic variables to show the compatibility of security activity and 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 2. By using these fuzzy logics one can provide the basis for the approximate reasoning [17]. These fuzzy vales offer more realistic approach for human reasoning than the binary values. The observation based upon the fuzzy logic of five security experts for the integration of security and agile activities are represented in the Table 3.

Table 2. 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)

Table 3. Observation on the basis of Linguistic Variable for Planning Agile Activity

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

C. Integration Method

To integrate security activities selected from Security engineering (SE) processes the following steps shown 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.

Step 1. Select the security activity form security activity table having highest Mean Agile Value.

Step 2. From the fuzzy matrix, list out the agile activities having compatibility value greater than threshold value of 0.35

Step 3. From this list select the agile activity having lowest Mean Agile Value.

Step 4. Check if the influencing factor plus Mean Agile Value for the selected agile activity is greater than the old Mean Agile Value, then the selected security activity can be integrated into selected agile activity

Step 6. Remove the security activity from security activity table. Repeat from Step 1 until security activity table is not empty.

D. Implementation

The above mentioned algorithm is implemented in Java and the output is plotted in figure 1, which shows the extent of the compatibility of agile activities with Security activities.

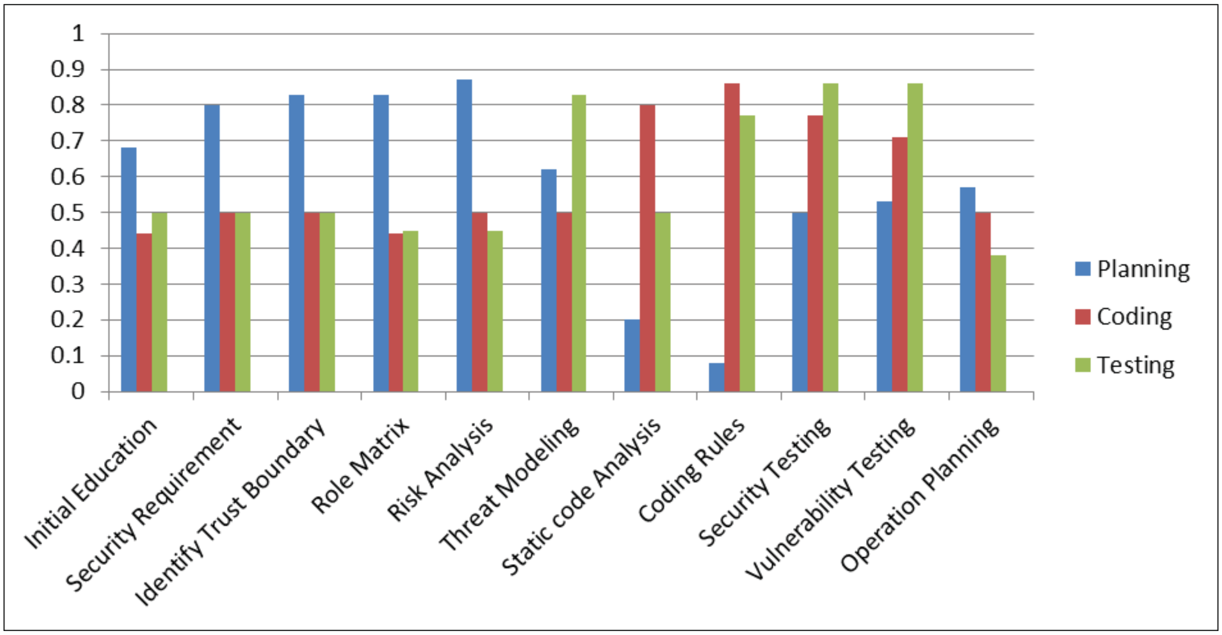

Figure 1. Showing the compatibility of agile activities with Security activities

IV.CONCLUSION

This work provides a 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 using fuzzy logic 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] Blitz, D.C., and van Vliet, P., “Global Tactical Cross-Asset Allocation”, Journal of Portfolio Management, Vol. 35, No. 1, pp. 23-38, 2008.
[3] Boehm, B. “A Spiral Model of Software Development and Enhancement”, Journal: Computer, Vol. 21, No. 5, pp. 61-72, 1988.
[4] Beck, Kent, Andres, Cynthia, Extreme Programiming: Embrace Change (2nd ed.). Addison Wesley Professional, Boston, 2004.
[5] 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.
[6] 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.
[7] 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.
[8] 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 .
[9] 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.
[10] Fitzgerald, B., Harnett, G., and Conboy, K. “Customizing Agile Methods to SoftwarePractices”, European Journal of Information Systems, Vol 15, No. 2, 2006.
[11] Kniberg, H., and Skarin, M. "Kanban and Scrum - making the most of both," C4Media Inc., USA, 2010.
[12] Highsmith, J. “Agile software development ecosystems”, Boston, M.A., Pearson Education, 2002.

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

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

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

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

[17] 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).

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[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.