A Comprehensive Approach for Embodiment of Security Activities with Agile Methodologies

Ajay Kumar Rangra
Department of Computer Science and Engineering,
Chitkara University
Himachal Pradesh
India

Manik Gupta
Department of Computer Science and Engineering,
Chitkara University
Himachal Pradesh
India

ABSTRACT
Agility among the software is seeking importance during the development phase, as it promotes adaptive planning, incremental and evolutionary development with many other features that are lightweight in nature. Security is one of the major issues in today's highly agile software development industry. More emphasize is on to produce a secure software, so as to minimize the amount of risk and damage caused by the software. Developing secure software with high agile characteristics is always a hard task to do because of heavy weight nature of security activities. This paper proposes a novel approach by which security activities can be integrated with agile activities by calculating the mean agility value of both activities i.e. agile as well as security keeping in mind the factors such as cost, time, recurrence, benefits affecting the agility of the activity. By using fuzzy value compatibility table (FVCT), extend of compatibility of embodiment of both the activities is done with fuzzy values.

Keywords
Mean Agility Values, Security Activities, Fuzzy Logics.

1. INTRODUCTION
Agile methodology has gained a significant popularity in the field of software development. The reason behind this popularity is various lightweight characteristics of agile methodology such as people orientation, iterative, collaboration, modularity, convergent [1].These characteristics help to speed up the development process. As the development speed increases, security of the software decreases. So it is always a concerned to secure the software that is developed by using agile methodology.

Security issue is a fundamental worry in software development industry. Because of software vulnerability major software are not able to stay in the competitive market. So it is important to develop secure software. Security activities are heavyweight process. As security is not the part of SDLC but it is needed in every phase of SDLC i.e. in requirement phase, analysis phase, design phase, implementation phase, test phase, deployment phase.

The injection of heavyweight security process to the popular agile methodologies which are lightweight may reduce the agility feature of agile methodologies, so special attention must be given for the injection of security in the agile method of software development.

A lot of research is carried out in the field of making the software secure. One of the area by which software can be secured is the integration of security activities with agile methodology. This paper is proposing an approach by which embodiment of agile methodology and security activities are achieved in a capable way.

In this paper, the proposed noble approach for the embodiment the security activity in agile method by calculating the mean agility value (MAV) i.e. compatibility level for the security activities as well as agile activities with various agile characteristics and also the various other factors such as time, cost, benefits and recurrence that affect the embodiment are considered. Moreover, the use of Fuzzy Value Compatibility Table (FVCT) which uses fuzzy value that shows extend to which a security activity can be embodiment into agile activity. In the last this paper proposed an algorithm for the efficient embodiment of security activities with the agile methodology's agile activities.

2. RELATED WORK
As the damage done to software product by various security issues the demand for developing secure software is mandatory for today's software industry. [2] is a process by which software is developed by keeping the security vulnerability in mind. This process add the security related activities in each phase of development cycle So the deliverable that has to be delivered after every phase are secure [3].

Over a few years a new method is gaining popularity in the software development industry i.e. Agile Methodology. The principal of agile methodology is set in Manifesto for Agile Software Development [4]. Most popular agile methods include Extreme programming, Scrum, Crystal Methodologies, and Feature Driven Development.

In [5], address the issue and make some proposals as to how security assurance activities could be merged into agile development methods; here the existing software assurance activities were divided into four categories: in first categories those activity is listed which are natural match for agile methods, in second category those activities are listed which are independent of any development methodology, in third category those activities are listed which can be integrated into agile methods, and in the last categories those activities are listed which fundamentally mismatched with agile methods.

Mikko Siponen [6] suggested the requirement of adding security activity into the agile methodology. Hossein Keramati, Seyed-Hassan Mirian-Hosseinabadi [7] suggests algorithm for the integration of security activity with agile methodology by calculating the agility degree and a variable

called ART. In his approach, the calculation of agility degree is based upon of agile characteristics having scale of 0 to 5, after the calculation of agility degree the algorithm is proposed which uses an AICM matrix to show the integration of agile and security activity by using binary values along with a tuneable parameter called ART.

Ching Torng Lin et al [8] proposed a way to calculate the agility with fuzzy logics. CLASP [9] later known as OWASP suggest a approach to add the security activity into the early phases of Software development phase. The main focus of CLASP is security metrics. CLASP includes a set of 24 activities and supplemental resources whose use should be adapted to the development process in use.

3. PROPOSED APPROACH

For the embodiment of security activities to agile methods paper proposed a noble approach. In proposed approach various short forms are used. The short form and there abbreviations is as MAV- mean agility value, FVCT- fuzzy value compatibility Table, SDPPM-software development proficient project managers, IFV-Influencing factor value, MCV- Mean Compatibility Value, CV-compatibility value, TV- Threshold value, DV-Delta Value.

The whole approach is divided in the following eight steps:

3.1. Assortment of Agile Characteristics

Agile methodology has gained huge popularity in software development industry. There are many reasons behind this popularity such as light weight, people oriented, adapt changes, frequent communication between team members, fast etc. The agile manifesto [4] gives twelve principles for the agile methodology. Granville G. Miller [1] listed out nine characteristics for the agile methodology.

3.1.1. Modularity

This characteristic focuses on the division of a large model into small modules for simple and efficient progress.

3.1.2. Iterative

It refers to the repeating of the small cycles. In each cycle a certain task is to be done.

3.1.3. Time Bound

Time bound is setting the time limit for the completion of activity.

3.1.4. Parsimony

In agile methodology there are no predetermined set of goals which are not able to be completed in limited amount of time so developer feel exhausted. So, agile focuses on minimum amount of activities to be completed in certain time limit.

3.1.5. Adaptive

The software development methods should be ready to tolerate changes at any stage of development.

3.1.6. Incremental

Agile process does not develop a process in one step only but it follows the approach of developing the software in increments, so that if there may any error at any stages that can be identified and removed.

3.1.7. Convergent

All the modules are developed in such a way that they achieve a common goal that is completion of project within specified time limits.

3.1.8. People Oriented

The main focus of agile is to keep in mind the people that are directly or indirectly relate to the software over the technology and processes.

3.1.9. Collaborative

There should be proper and frequent communication between the team members.

3.2. Assortment of Security Activities from Various Development Phases

Security activities are the activity that should be there in the software to make it secure. There are a lot of risks and threats are present in software, so by using security activities in the SDLC these risks and threats can be eliminated. Microsoft SDL [3], CLASP [9] is the software security process which deals with the various security activities that should be embodiment into software so that software should be secure. Based upon these above mentioned security processes paper listed some of the important security activities form every phase of security life cycles in Table 1.

Table 1. List of Security activities

Security Activities	Pre Phase	Initial Education and training
	Requirement Phase	Security Requirement
		Identify Trust Boundary
		Role Matrix
	Design Phase	Risk Analysis
		Threat Modeling
	Implementation phase	Static code Analysis
		Coding Rules
	Testing Phase	Security Testing
		Vulnerability Testing
	Planning Phase	Operation Planning

3.3. Calculation of MAV for Security Activities

Agility is a comprehensive response to the business challenges of profiting from rapidly changing, continually fragmenting, global markets for high quality, high-performance, customer-configured goods and services [12]. Agility is also the measure of quickness, flexible, dynamic, changes with requirements of customer, interactive. Based upon these parameters the level of compatibility of the security activity with the agile characteristics can be calculated. The paper is using scale of 0 to 5 to show the compatibility of activities with agile characteristics. Higher the value of, high is the compatibility, lesser value means lesser compatibility.

Based upon these values the proposed approach calculates the Mean agility value (MAVsa) of each security activity in such a way:

$$\text{MAVsa(i)} = \sum_{j=1}^{n} X(i,j)/n$$

where, n are total no of agile characteristics, $X(i,j)$ is the agile value of security activity 'i' for agile characteristics 'j', as calculated in table 2.

3.4. Assortment of Agile Activities from Popular Agile Methodologies

Agile methodology is a collection of agile activities that are present in every phase of agile methodology. So for the embodiment of security activities in to agile methodology one should firstly identify these activities. The proposed approach is not mentioning every activity from each phase but considering only three activities i.e. planning, coding and testing. In real scenario list of agile activity is not limited to these three, depends upon the desire of project manager that how much activities he want to analyze in his project

3.5. Calculation of the MAV Based upon the Characteristics of Agile Methodology

As calculated MAV for Security activities similarly calculation of MAV of Agile activities is done. Here paper also considering the scale of 0 to 5. 0 means less compatible whereas 5 mean more compatible with agile characteristics. More is the MAVaa, more the compatibility of activity with agile methods.

$$\text{MAVaa(i)} = \sum_{j=1}^{n} Y(i,j)/n$$

where, n are total no of agile characteristics, $Y(i,j)$ is the agile value of agile activity 'i' for agile characteristics 'j', as calculated in Table 3.

3.6. Formation of Fuzzy Value Compatibility Table (FVCT)

After calculating MAV for both agile and security activity, the formation of FVCT is done. FVCT illustrates the level of compatibility of security activities with agile activities. Values for the FVCT are calculated on the basis of observations of five software development proficient project managers based upon fuzzy values. L.A. Zadeh [10] proposed the fuzzy sets. This paper uses Fuzzy sets because of the following advantage, As proposed approach inspecting the compatibility of agile activity with security activity, judgment might not be based upon binary values true or false i.e. compatible and not compatible. The embodiment might be to a certain extend or range, some activity might not fully embodiment into other activity but to a certain extend embodiment might be possible. So, with the help of Fuzzy sets one can calculate relatively realistic values for the embodiment. Also in [11], L.A. Zadeh proposed the term linguistic variable meaning that a variable whose value is not a number rather it is word or sentence. 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 having values (0,0.10,0.15), 0.10,0.20,0.30), (0.25,0.35,0.45), (0.35,0.50,0.65), (0.55,0.65,0.75), (0.75,0.80,0.85), (0.80,0.95,1.00) respectively, These values are divided in three parts i.e. lower bound, probable value, higher bound. Based upon these fuzzy values the software development proficient project managers (SDPPM) give its own observations that how much is the compatibility of security activity with agile activity is possible. It might be possible that one SDPPM is not able to give the exact observation so, paper is using five SDPPM. This paper calculates the mean of observations of each SDPPM observations called Mean Compatibility Value (MCV) to fill the FVCT. The values in FVCT are known as Compatibility values (CV) shown in Table 4.

3.7. Calculation of Influencing Factor Value (IFV)

IFV table is based upon the various factors that affect the embodiment of current activity either be agile activity or security activity. The proposed approach taking care of four factors that affect the activities cost, benefits, time, and recurrence. Cost is factor which is calculated of the total expenditure during the development of the software. To calculate the cost factor scale of 1 to 3; where, 1 is hard, 2 is moderate and 3 is easy. If some activity has the values of cost factor as 1 that means it is very costly to embodiment this activity with other activity. If cost factor is as 3 than the cost of embodiment of this activity is very less. Benefit is other factor which effects the embodiment of activities. Scale of 1 to 3 to get the value of benefit is used . In this value 1 is low, 2 is average, 3 is high. If any activity has the influencing factor values as 1 than this activity is not much beneficial, whereas if value is 3 than activity is very beneficial. Time also affect the overall embodiment of activity. Time measure the priority of completion of each phase. This factor also uses 1 to 3 scales where 1 is least priority, 2 is average priority, and 3 is highest priority. If some any activity has time factor as 1 means timely completion of this phase is not least important, may be activity of that phase completed on time or not. If activity has value of time factor as 3 that means the activity in that phase should be completed in allowed time no delay is allowed. Recurrence of activity is also play an important role. Recurrence define the repetition of an activity in any phase weather that is least repeatable, average and highly repeatable scaled on 1 to 3 scale as 1 ,2 and 3 respectively. If any activity has recurrence value as 1 means it is less repeatable so its

Table 2. Calculation of MAV_{sa} for Security Activities on the Basis of Agile Characteristics

Security Activities \ Agile Characteristics		Modularity	Iterative	Time Based	Parsimony	Adaptive	Incremental	Convergent	People Oriented	Collaborative	Mean Agility Value (MAVsa)
Pre Phase	Initial Education and training	3	5	5	5	5	5	5	5	4	4.6
Requirement Phase	Security Requirement	4	4	4	5	4	3	3	4	4	3.8
	Identify Trust Boundary	2	5	2	5	4	2	4	2	4	3.3
	Role Matrix	3	4	3	4	3	3	2	4	4	3.3
Design Phase	Risk Analysis	2	5	2	5	4	3	4	2	3	3.3
	Threat Modeling	1	4	2	1	1	1	3	1	1	1.6
Implementation Phase	Static code Analysis	5	5	5	4	5	2	5	5	4	4.4
	Coding Rules	3	4	4	2	5	3	4	2	3	3.3
Testing Phase	Security Testing	1	5	3	3	5	2	4	4	3	3.2
	Vulnerability Testing	0	5	3	3	5	2	4	4	3	3.1
Implementation	Operation Planning	2	2	3	4	4	4	3	4	5	3.4

Table 3. Calculation of MAV_{aa} for Agile Activities on the Basis of Agile Characteristics

Agile Activities \ Agile Characteristics	Modularity	Iterative	Time Based	Parsimony	Adaptive	Incremental	Convergent	People Oriented	Collaborative	Mean Agility Value (MAVaa)
Planning	4	3	5	3	4	4	4	5	5	4.1
Coding	4	4	4	4	5	5	4	4	4	4.2
Testing	2	4	3	2	5	2	4	4	4	3.3

Table 4. Compatibility Values of Security Activities with Agile Activities

Agile Activities \ Security Activities	Initial Education and Training	Security Requirement	Identify Trust Boundary	Role Matrix	Risk Analysis	Threat Modeling	Static Code Analysis	Coding Rules	Security Testing	Vulnerability Testing	Operation Planning
Planning	0.68	0.80	0.83	0.83	0.87	0.62	0.2	0.08	0.5	0.53	0.57
Coding	0.44	0.5	0.5	0.44	0.50	0.50	0.80	0.86	0.77	0.71	0.50
Testing	0.5	0.5	0.5	0.45	0.45	0.83	0.5	0.77	0.86	0.86	0.38

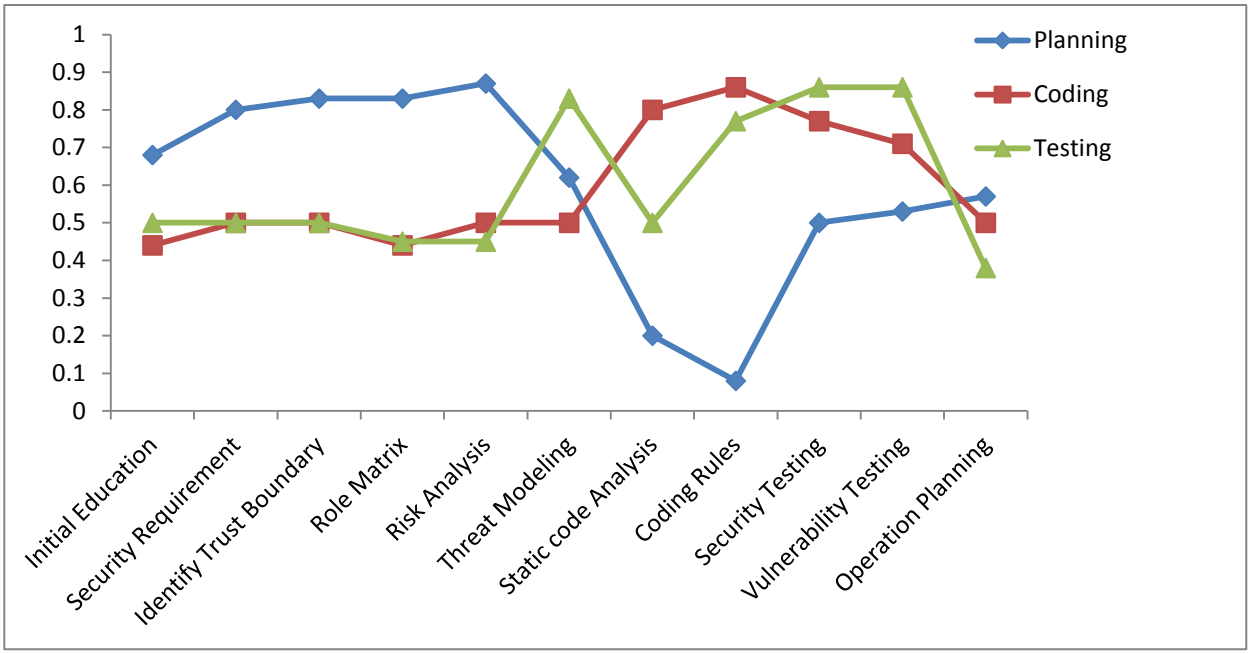

Fig 1: Showing the compatibility of various agile activities with security activities

affect on embodiment of activity is less or negligible. If value is 3 than recurrence is very frequent, so affect on embodiment is high. Based upon the values of above mentioned influencing factors proposed approach calculate the Influencing factor value IFV for security activity as well as agile activity.

3.8. Algorithm for the Embodiment

3.8.1. Select the security activity form security activity table having highest MAVsa.

3.8.2. From the FVCT, list out the agile activities having compatibility value (CV) greater than threshold value (TV) of 0.35

3.8.3. From this list select the agile activity having lowest MAV

3.8.4. Check out the IFV (influencing factor value) for selected agile activity and selected security activity in IFV table. Select the highest IFV among both the activities. If IFV and MAVaa for the selected agile activity is greater than delta value (DV), then the selected security activity can be embodiment in to selected agile activity

3.8.5. Remove the agile activity from the selected agility activity list. Repeat the step 3.8.3 and 3.8.4 until agile activities list is empty.

3.8.6. Remove the security activity from security activity table. Repeat from step 3.8.1 until security activity table is not empty.

Here, TV is the minimum threshold value that an activity must possess, so that it is able to embodiment. The algorithm set threshold value 0.35 means at least 35% activity should be compatible for the embodiment. If some activity have TV less than 0.35 means these activity does not have the capability to be integrated.

The DV depends upon the priority of the security activity embodiment. This value measures the importance of security activity in particular agile development phase. The DV depends upon the project manager expertise and can vary from development phase to phase, company to company depending upon other factor such as environment of the company, software delivery time, software security quality etc.

4. COMPARATIVE STUDY WITH PREVIOUS APPROACH

In [7], an algorithm is proposed for the integration of security activities with agile methodology. In his approach the calculation of agility degree to show the agile nature is depend upon value calculated by adding all the characteristic's values measuring on 0 to 5 grades. More the value of agility degree of the activity more is the compatibility with that feature, less the value showing the conflict on integration. Moreover the use of activity integration compatibility matrix (AICM) matrix uses the binary value to show the compatibility of security activity with agile activity. 1 value represents compatibility and 0 means incompatibility. In proposed approach the use of various security activities from every phase of development cycle is done. These activities are selected form the best activities from the various activities suggested by Microsoft SDL [3], CLASP [9] processes. This paper is calculating the Mean agility value (MAV) of every activity either is agile or security activity based upon the nine agile characteristics on the scale of 0 to 5. Higher the value more is agile less the value lesser is the agile nature of the activity. The benefits of calculating the mean is letting the agility value within the defined range of 0 to 5 which help in easy calculation. In the previous approach the use of AICM with binary value does not hold good for showing the compatibility of the activities. In that approach 1 value shows the compatibility whereas 0 denotes incompatibility, the reasons for the drawback of using the binary value is that might be a security activity is not fully compatible with agile activity but to certain extends it is compatible. There is no space in the previous approach for these kinds of integrations. So in proposed approach the use of fuzzy values to show the extend of capability ranging from 0 to 1 not exactly 0 and 1.By using this approach one can easily find out which security activity is most excellent integratable with which agile activity as shown in figure 1 .In proposed approach the various factors such as cost, benefits, time and recurrence that affect the compatibility of two activities which help the project manager in taking the decision that weather the security activity is embodiment with agile activity or not are also considered.

5. CONCLUSION AND FUTURE WORK

Making the software secure is the essence of today's software development industry. Securing the software is done with embodiment of security activities in every phase of software development life cycle not as in earlier approach in which software is secure after the development. In this paper, the proposed approach is calculating the MAV for both agile as well as security activities based upon the basic nine agile characteristics.The formulation of FVCT is done for estimating the compatibility of security activities with agile activities based upon the fuzzy values. The study considers various other factors that will affect the embodiment of security activities and agile activities; these factors include cost, benefits, time and recurrence based upon these factors the IFV is calculated which plays an important role in embodiment.

For future work, we can work upon the generality of different software environment for calculating the IFV that affects the embodiment. Other factor such as time-cost trade-off may also be considered while maintaining the embodiment of the agility based secured software.

6. REFERENCES

[1] Granville G. Miller, "The Characteristics of Agile Software Processes", *Proceedings of the 39th Int'l Conf. and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS'01)"s* 1530-2067/01, IEEE 2001.

[2] Lipner, Steve & Howard, Michael," *The Trustworthy Computing Security Development Lifecycle"* Microsoft Corporation, March 2005.

[3] Howard, M., Lipner, S., "The Security Development Lifecycle – SDL: A Process for Developing Demonstrably More Secure Software", Microsoft Press, 2006.

[4] "Manifesto for Agile Software Development," http://www.agilemanifesto.org

[5] Beznosov and Kruchten," *Towards agile security assurance'' NSPW '04 Proceedings of the 2004 workshop on New security paradigms, Pages 47-54, 2004.*

[6] Siponen, M., Baskerville, R., Kuivalainen,T, "Integrating security into agile development methods" in proceedings of the 38th Annual Hawaii International, 2005.

[7] Hossein Keramati, Seyed-Hassan Mirian-Hosseinabadi, "Integrating software development security activities with agile methodologies", IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, 2008.

[8] Ching-Torng Lin, Hero Chiu, Yi-Hong Tseng, "Agility evaluation using fuzzy logic", International Journal of Production Economics, Volume 101, , Pages 353–368, Issue 2, June 2006.

[9] Comprehensive, lightweight application security process. http://www.owasp.org, 2006.

[10] L.A Zadeh,"Fuzzy Sets" Information and Control Volume 8, Issue 3, June 1965, Pages 338–353

[11] L.A Zadeh,"The Concept of a Linguistic Variable and its Application to Approximate Reasoning-I" Information Sciences Volume 8, Issue 3, 1975, Pages 199–249

[12] Steven Goldman, Roger Nagel, and Kenneth Preiss, "Agile Competitors and Virtual Organizations", Chapter 3, Van Nostrand Reinhold, 1995.