http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (4): 769-778, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.769.778

Secure E-commerce Web Development Framework

[1]Bala Musa S., [1]Norita Md Norwawi and [2]Mohd Hasan Selamat
[1]Faculty of Science and Technology, Universiti Sains Islam, Malaysia
[2]Faculty of Computer Science and Information Technology
University Putra Malaysia, Malaysia

Abstract: This study presents a framework for safety critical e-commerce application development based on an Extreme Programming methodology with inbuilt security across the development lifecycle to mitigate security lapses. This approach tightens security checks and balances at every stage of development process, minimizes any vulnerability that will manifest in production environment and avoids an unnecessary extension of the life cycle.

Key words: E-commerce, development framework, extreme programming, web security, web engineering

INTRODUCTION

It is estimated that 75% attacks on web are done at the application layer, XSS and SQL injection are first and second reported vulnerabilities respectively (Adrian, 2008). Also browser vulnerabilities are potential avenue to infiltrate a malware (Wang, 2010).

Furthermore, a report of Sysadmin Audit Network Security SANS/FBI (2009) on the top cyber security risk threats covering March to August 2009 which summarizes vulnerability, attack trends and those threats that have the greatest potential to negatively impact on network and businesses, SQL Injection, Cross-site Scripting (XSS) and PHP File Include attacks continue to be the three most popular techniques used for compromising web sites.

Similarly, a survey conducted by Danny (2007) shows that there are 80% potential threats to web applications using cross-site scripting, 62% potential threat using Structure Query Language SQL Injection, 60% Parameter Tempering and 37% Cookies Poisoning. As a result of this, many web companies try to verify what they receive and from what source the content is originating from Ilyas *et al*. (2004).

Hence, it is apparent that most of web application insecurity has nothing to do with Secure Socket Layer but the application layer whereby safe programming methods have been identified as major ways of avoiding security threats (Stuttard and Pinto, 2008) and also the root causes of insecurity and vulnerabilities in web application lies with the development life cycle (Stijn, 2004). Therefore, security controls at early stage of life cycle of web development are necessary for mitigation (Caleb, 2007).

Since attacks are possible as a result of improper coding practices such as SQL injection that can be avoided by proper parameterized queries for database access; there is the need to select the appropriate development methodology that will allow for the insertion of checks and balances in order not to see a repeat of those attacks.

Therefore, this paper presents a framework for safety critical e-commerce application development based on an Extreme Programming (XP) methodology by inserting tight security controls across the development lifecycle, reviewing of developers' codes by their pairs from security perspective at various stages and the prioritizing and handling of vulnerabilities.

OVERVIEW OF WEB DEVELOPMENT FRAMEWORK

Agile methods have become widely acceptable and essential in web application development (Pressman, 1986) due to their agility nature and provision for changing requirements. They as well emphasis on client collaboration and satisfaction, defect rate reduction and most importantly their short iterative nature within a time frame.

Therefore, researchers such as Boehm and Turner (2003) and Nerur *et al*. (2005) have compared agile methods with Plan-driven software development in an attempt to evaluate the two blocks of methods, they found based on the comparison that agile methods are geared towards customer satisfaction, lower defects rates, faster development time and solution to rapid changing

Corresponding Author: Bala Musa Shuaibu, V1A-02-09, Block 1 Sri Palma Villa, Bandar Universiti Technology, Legenda 71700 Mantin, Negeri Sembilan, Malaysia Tel: +60149443697

requirements, whereas Plan-driven such as Classroom, Personal Software Process are geared towards predictabilities, stability.

Also, agile methods have proven successfulness and are preferred over traditional due to their short iterative process and there design to capitalized on each team and individual unique strength (MacCormack, 2001). The methods measure success in terms of getting requirements in a good time and better than anticipated. Whereas traditional methods looks at success in terms of getting requirements delivered within budget (Fowler, 2005).

However, despite the success shown by agile methods over traditional or heavy weight methods, researchers have delved further into comparison of the various agile methodologies using different frameworks and tools in order to give an insight on which agile method is appropriate for which home ground. Some of the most systematically guided comparisons like Qumer and Henderson-Sellers (2008) selected XP and Scrum as the two well regarded agile methods for comparison. They used 4-Dimensional Analytic Tool mainly built for agile process to carry out the comparison while Abrahamsson Warsta (2003) compared nine agile methods using Lenses Analytical tool that measures the methods based on software development life cycle, project management, abstract principles/concrete guidelines, universal pre-defined/situation appropriate and empirical evidence.

Another comprehensive analysis using framework for agile methods that incorporates most of the criteria used by Abrahamsson *et al*. (2003) and other additional criteria was described by Strode (2007). The comparisons have revealed the strength of different agile methods within the itemized perspective.

Extreme programming methodology (XP): Although the parameters for determining what methodologies are appropriate for what web application project are still inadequate (Hadjerrouit, 2001; Altarawneh and Shiekh, 2008) safety critical e-commerce applications needs absolute involvement of all stakeholders in defining security requirements. In addition, agile methodologies have been widely considered in web application development (McDonald and Welland 2003) due to their rapid delivery and accommodation in terms of change request. Specifically, the principle behind agile manifesto stipulates that customer be satisfied through timely delivery, provision for changing requirements and most importantly daily interaction between developers and business representative (Alliance, 2001). More so, Highsmith and Cockburn (2001) asserted that the customer collaboration is a key issue for any cost sensitive project leader to consider. Hence, in this approach, Extreme Programming methodology XP in which its key strength is the involvement of whole team including business representatives, project managers and technical teams in its iteration that allows easy adaptation to changes and who's emphasis is on coding rather than processes (Grisham and Perry, 2005) which is also a vital stage to introducing defects and exposing application to vulnerabilities and attacks if not fully considered is improved with inbuilt security checks and balances for critical e-commerce web development.

Therefore, the proposed approach allows for developers to review their pair's code which in essence increase retention rate, reduces gap in retention among pairs and produces qualitative codes as shown by McDowell *et al*. (2003) and Cockburn and Williams (2001) in regards to the benefit of pair review. It also further serves as a double check and walkthrough for vulnerabilities which Beck (1999), Hanks *et al*. (2004), McDowell *et al*. (2006) and Van De Grift (2004) have all attested to the fact that pair programming has great achievement in meeting desire goal. Therefore, the benefit of XP of typically involving the customer primarily during the early and late phases of the development process, specifically requirements elicitation and analysis, budget and contract negotiation, and acceptance testing (Grisham and Perry, 2005) is harnessed.

Other gains such as reduction of time spent on documentation that has made XP to gain productivity over other document-centric development process (Maurer and Martel, 2002) and its approach to bring together all developers to work closely so they can communicate informally rather than spending time documenting designs and decisions with Code-and-fix lifecycle described by Schach (2001) is utilized.

RELATED WORK

Threat modeling: Previous works focusing on security of information have touched on Threat Modeling such as Sodiya *et al*. (2007) designed a fuzzy logic-based threat modeling technique using fuzzification of input variables based on STRIDE- Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Perhaps comparison would have been done using other models. Threat modeling is an essential process that informs the degree of threats and risk during the development of software and how these threads should be addressed. The threat modeling document helps in proper understanding of risk to application and its degree so that proper measures be put in place.

Therefore, considering security and functionality as priority, there is need for threat modeling to be performed not just at the beginning of the life cycle of web development, but at various stages and in an iterative form that will reveal threats during the iterations.

There are basically six stages to be performed in threat modeling detailed by Mark (2003):

- Identifying protected valuable assets
- Create an architectural overview of the application
- Decompose the architecture of the application to a security profile uncover vulnerabilities in the design, implementation, or deployment configuration
- Identify the threats in the application while keeping the goals of an attacker in mind
- Document the threats
- Rate the threats based on preference starting with those that poses the biggest risk

Therefore, threat modeling produces document that prioritize threats and how to deal with it at an earlier stage. It is a pre-emptive approach to dealing with vulnerabilities.

In the same vein, Stijn (2004) uses the same STRIDE Threat model developed by microsoft to perform upon the WE Rock 24/7 web application, he organized the WE Rock 24/7 data point, assets, trust level, DFD, threat trees and vulnerabilities using a GUI-based.Net tool to a threat model document.

Similarly, Olzak (2006) made a recommendation for a high-level methodology for threat modeling that will allow security analysts to develop documentation necessary to make the right choices. But in terms of mitigating the attack risk, there are differences from the cost of eliminating the conditions necessary for an attack.

Adopted engineering methodologies: The absence of an established web engineering methodology has widely opened the field to adopt or adapt from software engineering method of Altarawneh and Shiekh (2008) and Jazayeri (2007) these methodologies are based on design, model or code. The Agile methodologies have been argued by Pressman (1986) as significantly necessary when developing a web application.

Hence, Souza and Falbo (2005) in their Agile modeling approach to web system engineering uses the conventional software engineering process comprising of requirement specification, analysis, design and implementation to propose a framework for Web Engineering. In their approach, the four phases has to be applied iteratively to allow for obtaining users feedbacks.

In addition, they assert that agile modeling principle should be integrated in the requirement specification and analysis stages. However, the major difference here is that they included Agile Modeling at the first two stages, while using at the design stage, software architecture for web applications which advocates for the usage of web application extension of UML. At the implementation stage, framework for web application development is used. But the framework is short of comparison to various combinations of frameworks in order to prevent over-fitting.

Similarly, a methodology involving structural design, detail and implementation design for web application was proposed by Hsieh (2003) in which emphasis was made on small student web development that is non-commercial application. His approach captures first of all the structural design that views a system at web page level, then the detail design which extend to intra-pages and finally the implementation of the visual interface, client-side scripting and task without visual interface.

Also proposed by Altarawneh and Shiekh (2008) is a theoretical agile process framework for web application development in small software firms. It involves three logical stages: small web project, adopt modified XP process methodology, apply XPMM as quality model for software process improvement, Education and training, internal assessment and light formal review, and external assessment. They contended that their approach will go a long way in reducing management overhead while keeping the customer's interest.

Meanwhile, Menchaca *et al.* (2005) introduces a framework that combines design, analysis and implementation tool for collaborative virtual environment in web. The methodology uses software engineering, HCL (Human Computer Interaction) and Java Technology. They also proposed tools such as: a model for the conceptualization of the virtual world, a graph-based high level notation, and a Java-based software framework to facilitate the implementation of the CVE.

Onto weaver Methodology is a modeling methodology for customization in web application presented by Lei *et al.* (2003) that allows for the target web application to be represented in an exchangeable format. They argued that it will enable management and maintenance of web application to be done at conceptual level. It involves specifying customization requirement rules for individual users. It also uses ontology to abstract all aspect of target web application. While WebHelix is a work in progress methodology presented by Whitson (2006) based on design methodology that is quite small for student comprehension of method in school.

Furthermore, a template meta-data based code generation is also proposed. This is a code based approach to web methodology presented by Zhang *et al.* (2004) that benefits the developers and designers of web application. It follows the general flow:

Data model → Navigation model → Presentation model

Therefore, in their approach, a WebGen is used to generate a prototype of a web application which they argued that it makes it a fast-prototyping methodology.

Secure development methodologies: Previous works on security such as Keramati and Mirian-Hosseinabadi (2008) focuses on the agile methodology been integrated with software development to equip the security. They restrain reduction of agile nature of organization's current process by means of agility using their approach, with security features to increase product's trustworthiness. Similarly, Meledath (2006) described the application of use cases and misuse cases which was incorporated into software design and implementation to identify security threats and requirement. Although it provides an early thought into security requirement in software life cycle, but this is centered on novice developers or students. However, to mitigate for industrial or commercial consideration, Eduardo (2004) presented a methodology to build secure software using object-oriented design, the Unified Modeling Language (UML) and patterns but short of define guidelines to apply the methodology more precisely at each level. All the works discussed are based on software development process. Therefore considering the works on security in web applications in this context, Ge Paige *et al.* (2006) investigated general-purpose information system development methods particularly the feature driven development and risk analysis, and integrate them to address the development of secure web applications. In their approach, risk assessment is fused into agile processes which is an engineering method that satisfy functional requirement for web application. The hand-shake of these two mechanisms is aimed at mitigating security in web development process. But the FDD approach focuses mostly on the design and building phase rather than the entire development process (Palmer and Felsing, 2001). Also, the key issue addressed in their approach includes the decreasing life-cycle time which in a safety critical web application has to be compromised or trade-off.

Sengupta *et al.* (2005) put forward a framework for security in e-commerce using a general life cycle approach. They first of all highlighted some perceived threats and vulnerabilities associated to e-commerce, which afterwards proposed the framework for mitigation. While Scott and Sharp (2002) seek to address web application insecurity at a higher level and reducing development time and projects against application level attack. They developed an Object oriented API with security. It accommodates dealing with access control and secure database abstraction layer by using structured data type to manipulate SQL at the abstract syntax level.

METHODOLOGY

Key security factors: Different security web development models have being proposed. While some of these models inserts security checks in the existing software development models, others proposes new security models to suit the concept. However, all the models basically comprising of requirement, design, development, testing and implementation phases. Therefore, to scrutinize the existing security development models in order to ascertain that security has been fully addressed, the following key factors based on expert practitioners and researchers' views are used as analytic tool (Table 1). Putting security checks at the very beginning of a life cycle can reduce tremendously vulnerabilities and result to faster release cycle. Also, uncovering bugs at early stage is far less than at later stage or even when system is in production. Therefore, integrating security at early stage will guaranty robustness and minimize risk (Redspine, 2009). This can only be achieved when Stake holders collaborate in defining security thereby security professionals will familiarize themselves with business risk on one hand, and the business representatives will be informed on the impact of errors or bugs to the data of their customers on the other hand. This will allow for security trade-off (Diana, 2009).

Furthermore, Use and Misuse-cases are very essential in determining what control has to be in place at early stage of the development. Though use cases show how a user uses certain functionality of a system, misuse-cases will show how an attacker could exploit the system (Redspine, 2009; Diana, 2009). Also important is the Design security control that addresses expensive security problem at early (Diana, 2009). Though Security Infrastructure Definition is the technology needed for security policy and requirement enforcement (Sengupta *et al.*, 2005), in an iterative development, this will be taken care off at different phase of the life cycle.

While code Review serves as double check for errors introduced by developers during coding stage, a third party review ensures that coding standards are adhered to, and no vulnerability is further introduced.

Table 1: Key security factors to be address in the lifecycle

Key security Factor	Description and References
Stalk holders collaboration on security deffinition	Are there both business representives and development team defining security at requirement phase? (Diana, 2009)
Security at the very beginning	Is the requirement specification phase including security requirements? (RedSpine 2009), (SPI Dynamic Secure Protect and Inspect 2002)
Use and Misuse-cases at requirement and design	Is use and Misues-cases incalculated at the requirement and design? (RedSpine, 2009; SPI Dynamic Secure Protect and Inspect, 2002: Diana, 2009)
Design security control	Does the design phase present security control that relates to production environment such as risk analysis or threat modeling? (Diana, 2009)
Security infrastructure definition	Does it specify what kind of security infrastructure definition is needed? (Sengupta *et al.*, 2005)
Code review for vulnerability	Does the testing phase check and review codes? (Diana, 2009)
Prioritization for vulnerability	Does the testing phase have prioritization on vulnerability starting with the most critical? (Diana, 2009)
Monitoring production environment for leakages	Is there plan for monitoring leakages at production environment? (Diana, 2009)

Table 2: The performance of the frameworks based on key security factors

Security Factors	(Ge *et al.*, 2006) (Xiaocheng)	(Sengupta *et al.*, 2005)	Proposed model
Stalk holders collaboration on security definition	No detail of stake holders colloboration on security	No define stakeholder's collaboration	Whole team collaboration of stalk holders to define detail Of asset protection
Security at the very beginning	Security policy requirement defined at beginning	Security asset to be protected is defined at the requirement specification	Detail physical and intellectual property protection definition
Use and Misuse-cases at requirement and design	Only use cases in both requirement and design	No use and misues-cases	Use/Misues-cases for functionality guaranty Show how users interact with functions Show how new vulnerability may be introduced
Design security control	Has security control with risk analysis in iteration at design	No securoty control at design	Risk and architectural review Threat modeling Test plan docnment
Security infrastructure definition	No definition for securoty infrastructure	Security tools for asset protection	No tools define
Code review for vulnerability	Does not specify details of code review	Present testing without composite rview	Secure coding practice Secureinfrastructureenforcement Codes comply with security procedures Security best practices Immnne codes Use automated tool
Prioritization for vulnerability	Prioritization of vulnerability using threat modeling	Has prioritization of vulnerability	Describe functions with security implication metaphorically Integration Testing (spotting errors and their impact) Penetration Testing
Monitoring production environment for leakages	No plan for monitoring in production environment	Provide plan for monitoring in production environment	Concrete plan for monotoring using portfolio analysis

To ensure that most critical threat to business and security of the system are dealt with first before the least critical ones, prioritization of vulnerability (Diana, 2009) has to be done which at the same time a monitoring production environment for leakages is also a security necessity in order to give assurances that the pre-deployment and in-production risk assessment provides useful feedback on security (Diana, 2009).

RESULTS

Comparative analysis result: From the previous review, the two prominent frameworks that are centered on security in web development life cycle and that chooses specific development methodology for security purpose are those proposed by Sengupta *et al.* (2005) and Ge *et al.* (2006).

Apparently, Xiaocheng's model has a defined security in its requirement phase but lacks a definite stake holder's collaboration in defining these requirements. On the other hand, Sengupta's model has no description of security at very beginning but with defined stake holder's collaboration. Hence our proposed model combines both factors at its very beginning with detail intellectual property to be protected (Table 2).

The performance of the three models based on the use-cases and the misuse-cases in the requirement and

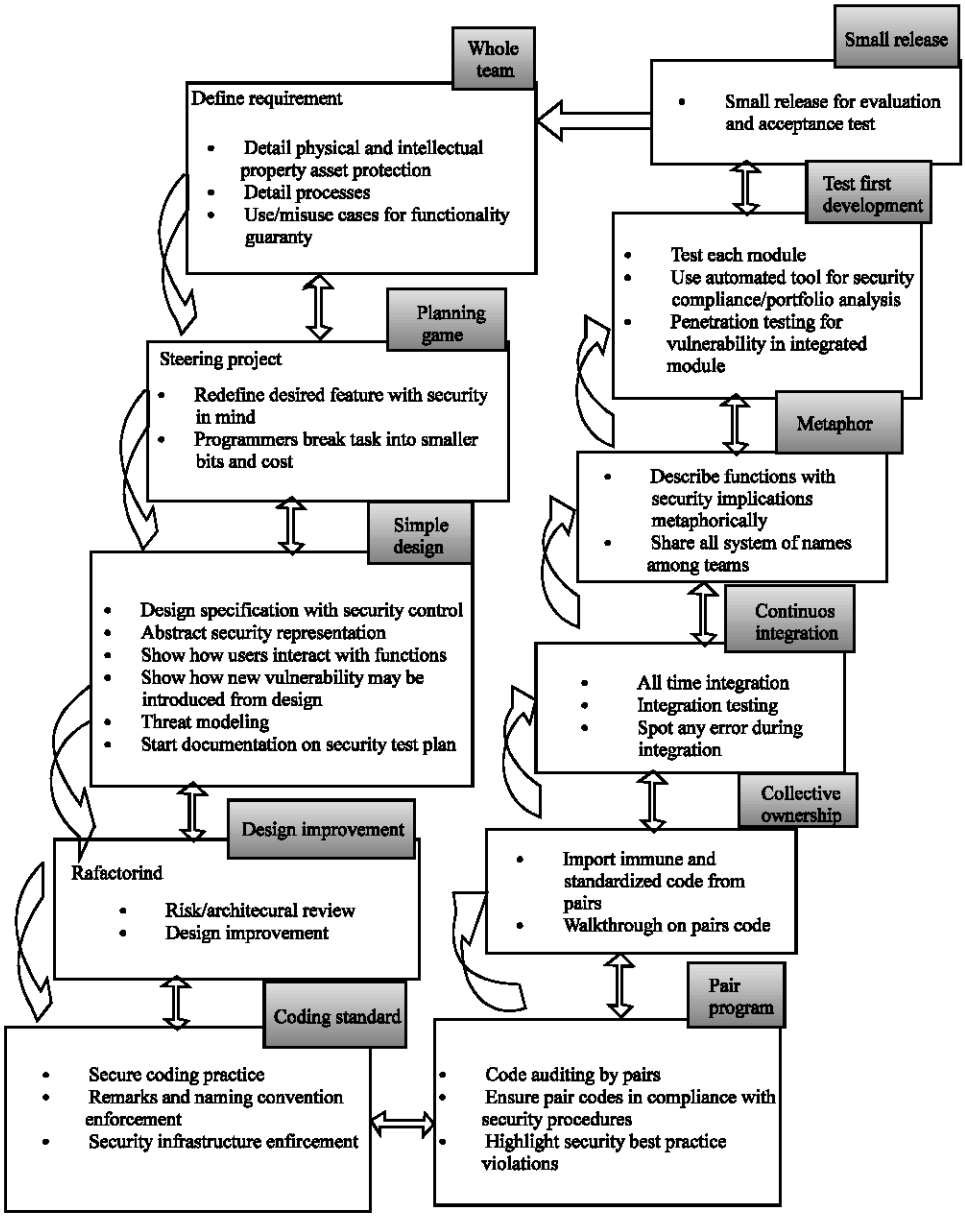

Fig. 1: Proposed secure E-commerce development frameworks

design phase has shown that Xiaocheng's model only included use cases at both requirement and design stage without reference to misuse cases at any of the stage. Sengupta's model neither included use cases nor misuse cases at the requirement and design stage. Our model inculcated this security aspect due to its enormous benefit as it uncovers threat at early stage.

Therefore, the proposed model has shown significant improvement based on the analytic factors that are key to security in web application development.

PROPOSED SECURE E-COMMERCE DEVELOPMENT FRAMEWORK

The major difference between conventional software system and security system is that, the former emphasis more on functional accuracy based on requirements while the later dwells more on the system properties that is immune from attack (Sengupta *et al*., 2005). This characteristic of security is highly necessary in e-commerce web application since fund transfer, web shopping, public retirements and a great deal of credit

card information or Pin numbers are involved. Thus, we present a secured web development framework based on an extreme programming methodology that involves all stakeholders in fashioning out security requirement which is changing with new vulnerabilities and tightens security controls across the development lifecycle with mechanism for developers to review their pairs code from security perspective.

Figure 1 depicts the proposed framework. Detail explanation with built-in security is as follows:

- **Whole team:** All stakeholders collaborate in a team to define the requirement at the very beginning of the project. This offers an opportunity for security requirements and controls to be discussed. Business representatives who may not understand security requirements and acceptance testing in line with the company policies and regulation may be guided by the analysts and testers. The managers provide resources and other logistics for the discussion
  Use/misuse case are also defined in order to start looking into possible ways the system can likely be compromised. Also, any changing requirement may lead to review of security policy as well as Use/misuse cases. This will ensure that the functionality is guaranteed and that information is secured from unauthorized users. Similarly, physical and intellectual property must be considered
- **Planning game:** Although this stage is basically predicting what will be accomplished by the due date and determining what to do next, the knowledge acquired during the first stage will guide the business representatives to go back to the drawing board and redefine the desired features with security in mind. The programmer's team will then break the task into smaller bits, estimate cost and what will be done in the next iteration
- **Simple design:** Design is not a one-time thing or up-front thing and hence a revision through refactoring is achieved until the end. For the purpose of security, a design specification that shows security controls and how such design may introduce vulnerability is done at this stage. Risk analysis and threat model gears the development of security test plan which will be documented and used during Test-First Development. All design must be abstracted with details of security in each representation
- **Design improvement:** This is the continuous improvement called refactoring while paying attention closely to the business requirement as well as doing risk assessment of possible attack tunnel. This will ensure the development of a good design. There is also improvement of those architectural constrains that introduces risk in the design at this stage
- **Coding standard:** For security implications, secure coding practice including remarks, naming convection etc must be ensured. Security infrastructure must also be enforced to establish a uniform coding for all team members who makes it easier for collective ownership
- **Pair programming:** Two or more programmers build the code which ensures auditing and review and as well tighten security. At this stage, pairs must ensure that any lack of compliance with security procedure or violation of security practice is highlighted
- **Collective code ownership:** Collective code ownership is when programmers have access to codes develop by other programmers without need for permission. The security alignment is the use of standardized and immune code from attack to be used a walkthrough must be accomplished also.
- **Metaphor:** Information must be fully shared among the team to facilitate adequate knowledge of where to find any functionality and its rightful position. Identify and separate with appropriate remark any function that has security implications to the development of the system
- **Test-first development:** Collate feedback at every stage of the code development. Developers can get the result of the test carried out on every single code they release before the final integration. Include security test on each module to give an iterative check for vulnerabilities. This gives some sort of security assurance to the development. Source code should also be check using automated tool for security compliance. The portfolio analysis checks the cross-enterprise risk that can have security impact
- **Continuous integration:** Keep the system fully integrated at all time to enable errors be visible at early stage that is not apparent during testing of the integrated system
- **Small release:** Perfume a small release to customer for evaluation of business requirement fulfillment and to accelerate the rapid development.

CONCLUSION AND FUTURE WORK

The challenges in delivering a secure web application methodology that will allow for secure web with less vulnerability remains at large, and the security in web applications especially e-commerce application is not afterthought issues were it is considered only at the end

of development life cycle or just before the application is made online for the purpose it is meant for. Therefore, there is an absolute need for security consideration from the requirement stage down to the deployment in order not to put the information of legitimate or authorized users, customer or clients at jeopardy.

Our approach presents a framework for safety critical e-commerce application development based on an Extreme Programming methodology with inbuilt security across the development lifecycle to mitigate security lapses. The approach inserts and tightens security checks and balances at every stage without necessarily extending the life cycle. Importantly, the approach allows for developers to review their pair's codes from security perspective which will serve as a double check and walkthrough for vulnerabilities and provides a mechanism for prioritizing and dealing with vulnerabilities.

The next line of action in this research will focus on improvement of the framework based on its applicability in different domain.

ACKNOWLEDGEMENTS

We wish to thank Dean, School of Post Graduate Studies for his support in providing the logistics and guidance, and also the reviewers for their profound support and scrutiny which assisted tremendously towards this research.

REFERENCES

Abrahamsson, P. and J. Warsta, 2003. New directions on agile methods: A comparative analysis. Proceedings of the 25th International Conference on Software Engineering, May 3-10, Portland, Oregon, IEEE Computer Society, pp: 244-254.

Adrian, O., 2008. Web application vulnerability and IBM rational appscan. Proceedings of the IBM Rational Software Development Conference 2008, (RSSDC'08), Orlando FL, pp: 79-88.

Alliance, A., 2001. Manifesto for agile software development. Retrieved February 13, http://www.agilemanifesto.org/principles.html

Altarawneh, H. and A.E. Shiekh, 2008. A theoretical agile process framework for web applications development in small software firms. Proceedings of the 6th International Conference on Software Engineering Research, Management and Applications, Aug. 20-22, IEEE Computer Society, Washington DC. USA., pp: 125-132.

Beck, K., 1999. Embracing change with extreme programming. Computer, 32: 70-77.

Boehm, B. and R. Turner, 2003. Observations on balancing discipline and agility. Proceedings of the Conference on Agile Development, June 25-28, IEEE Computer Society, Washington DC. USA., pp: 1-32.

Caleb, S. and L. Vincent, 2007. InforSecWriters. Effective controls for attaining continuous application security throughout the web application development life cycle. Retrieved August 29. http://www.infosecwriters.com/texts.php?op=display&id=583

Cockburn, A. and L. Williams, 2001. The Costs and Benefits of Pair Programming. Addison-Wesley Longman Publishing Co. Inc., Boston, MA. USA., ISBN: 0-201-71040-4, pp: 223-243.

Danny, A., 2007. Managing a growing threat: An executives guide to web application security. Web Application Security Executive Brief, New York, USA., pp: 1-8. ftp://ftp.software.ibm.com/software/rational/web/brochures/r_web_app_security.pdf.

Diana, K. and Security Curve, 2009. Practical approaches for securing web applications across the software delivery lifecycle. IBM White Paper 3-7, USA. http://viewer.media.bitpipe.com/1033409397_523/1268240390_67/White-Paper-3.pdf.

Eduardo, B., 2004. A methodology for secure software design. Proceedings of the 19th International Conference on Database and Expert Systems Application Turin, (ICDESAT'04), Boca Raton, FL., pp: 1-7.

Fowler, M., 2005. The new methodology. December 13, 2005. http://www.martinfowler.com/articles/newMethodology.html.

Ge, X., R.F. Paige, F.A.C. Polack, H. Chivers and P.J. Brooke, 2006. Agile development of secure web applications. Proceedings of the 6th International Conference on Web Engineering, July 11-14, Palo Alto, California, USA., pp: 305-312.

Grisham, P.S. and D.E. Perry, 2005. Customer relationships and extreme programming. Proceedings of the Workshop on Human and Social Factors of Software Engineering, July 2005, ACM, St. Louis, Missouri, pp: 1-6.

Hadjerrouit, S., 2001. Web-based application development: A software engineering approach. SIGCSE Bull., 33: 32-34.

Hanks, B., C. McDowell, D. Draper and M. Krnjajic, 2004. Program quality with pair programming in CS1. Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, June 28-30, ACM, Leeds, UK., pp: 176-180.

Highsmith, J. and A. Cockburn, 2001. Agile software development: The business of innovation. IEEE Comput., 34: 120-122.

Hsieh, S., 2003. Software engineering for Web application development. J. Comput. Small Coll., 19: 10-19.

Ilyas, Q.M., Y. Zongkai and M.A. Talib, 2004. A journey from information to knowledge: Knowledge representation and reasoning on the web. Inform. Technol. J., 3: 163-167.

Jazayeri, M., 2007. Some trends in web application development. Proceedings of the Future of Software Engineering, May 23-25, IEEE Computer Society, Washington DC. USA., pp: 199-213.

Keramati, H. and S.H. Mirian-Hosseinabadi, 2008. Integrating software development security activities with agile methodologies. Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications, March 31-April 4, Doha, pp: 749-754.

Lei, Y., E. Motta and J. Domingue, 2003. Design of customized web applications with OntoWeaver. Proceedings of the 2nd International Conference on Knowledge Capture, Oct. 23-25, ACM, Sanibel Island, FL. USA., pp: 54-61.

MacCormack, A., 2001. Product-Development practices that Work: How internet companies build software. MIT Sloan Manage. Rev., 42: 75-84.

Mark, C., S. Joel and O. Erik, 2003. Improving web application security: Threats and countermeasures. Version 1.0, Microsoft Corporation. http://www.cgisecurity.com/lib/Threats_Countermeasures.pdf.

Maurer, F. and S. Martel, 2002. Extreme programming: Rapid development for web-based applications. IEEE Internet Comput., 6: 86-90.

McDonald, A. and R. Welland. 2003. Agile web engineering (AWE) process: Multidisciplinary stakeholders and team communication. Proceedings of the International Conference on Web Engineering, (ICWE'03), Springer-Verlag, Oviedo, Spain, pp: 515-518.

McDowell, C., L. Werner, H.E. Bullock and J. Fernald, 2003. The impact of pair programming on student performance, perception and persistence. Proceedings of the 25th International Conference on Software Engineering, May 03-10, IEEE Computer Society, Portland, Oregon, pp: 602-607.

McDowell, C., L. Werner, H.E. Bullock and J. Fernald, 2006. Pair programming improves student retention, confidence and program quality. ACM. Commun. 49: 90-95.

Meledath, D., 2006. Secure software development using use cases and misuse cases. Information Systems. http://www.iacis.org/iis/2006_iis/PDFs/Damodaran.pdf

Menchaca, R., L. Balladares, R. Quintero and C. Carreto, 2005. Software engineering, HCI techniques and Java technologies joined to develop web-based 3D-collaborative virtual environments. Proceedings of the Latin American Conference on Human-Computer Interaction, Oct. 23-26, ACM, Cuernavaca, Mexico, pp: 40-51.

Nerur, S., R. Mahapatra and G. Mangalaraj, 2005. Challenges of migrating to agile methodologies. ACM. Commun., 48: 72-78.

Olzak, T., 2006. A practical approach to threat modeling. http://adventuresinsecurity.com/blog/wp-content/uploads/2006/03/A_Pr'actical_Approach_to_Threat_Modeling.pdf

Palmer, S.R. and M. Felsing, 2001. A Practical Guide to Feature-Driven Development. 1st Edn., Pearson Education, Mumbai, pp: 299.

Pressman, R.S., 1986. Software Engineering: A Practitioners Approach. 2nd Edn., McGraw-Hill, Inc., Boston.

Qumer, A. and B. Henderson-Sellers, 2008. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Inf. Softw. Technol., 50: 280-295.

Redspine, 2009. What executives need to know about web application development security. Redspin Inc., Capinteria. http://www.bitpipe.com/detail/RES/1257782304_345.html.

Schach, S.R., 2001. Object-Oriented and Classical Software Engineering. McGraw-Hill Higher Education, Boston.

Scott, D. and R. Sharp, 2002. Developing secure web applications. IEEE Internet Comput., 6: 38-45.

Sengupta, A., C. Mazumdar and M.S. Barik, 2005. E-Commerce security: A life cycle approach. Sadhana, 30: 119-140.

Sodiya, A.S., S.A. Onashoga and B.A. Oladunjoye, 2007. Threat modeling using fuzzy logic paradigm. Informing Sci. Inform. Technol., 4: 53-61.

Souza, V.E.S. and R.D.A. Falbo, 2005. An agile approach for web systems engineering. Proceedings of the 11th Brazilian Symposium on Multimedia and the Web, Dec. 05-07, Pocos de Caldas-Minas Gerais, Brazil, AC., pp: 1-3.

Stijn, V.K., 2004. Threat Model for Web Application Using STRIDE Model. Royal Halloway University, London.

Strode, D.E., 2007. The Agile Methods: An Analytical Comparison of Five Agile Methods and an Investigation of Their Target Environment. Palmerston North, Massey University, New Zealand, pp: 224.

Stuttard, D. and M. Pinto, 2008. The Web Application Hackers Handbook: Discovering and Exploiting Security Flaws. Wiley Pub., Indianapolis, IN.

Sysadmin Audit Network Security SANS/FBI, 2009. The top cyber security risk. http://www.sans.org /top-cyber-security-risks/.

Van De Grift, T., 2004. Coupling pair programming and writing: Learning about students perceptions and processes. Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, March 03-07, ACM, Norfolk, Virginia, USA., pp: 2-6.

Wang, L., H. Mu, L. Xu, J. Chen, X. Liu and P. Chen, 2010. Trojan URL detector: A statistical analysis based trojan detection mechanism. Inform. Technol. J., 9: 1124-1132.

Whitson, G., 2006. WebHelix: Another web engineering process. J. Comput. Small Coll., 21: 21-27.

Zhang, J., J.Y. Chung and C.K. Chang, 2004. Towards increasing web application productivity. Proceedings of the ACM Symposium on Applied Computing, March 14-17, ACM, Nicosia, Cyprus, pp: 1677-1681.

Sysadmin Audit Network Security SANS/FBI, 2009. The top cyber security risk. http://www.sans.org /top-cyber-security-risks/.

Van De Grift, T., 2004. Coupling pair programming and writing: Learning about students perceptions and processes. Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, March 03-07, ACM, Norfolk, Virginia, USA., pp: 2-6.

Wang, L., H. Mu, L. Xu, J. Chen, X. Liu and P. Chen, 2010. Trojan URL detector: A statistical analysis based trojan detection mechanism. Inform. Technol. J., 9: 1124-1132.

Whitson, G., 2006. WebHelix: Another web engineering process. J. Comput. Small Coll., 21: 21-27.

Zhang, J., J.Y. Chung and C.K. Chang, 2004. Towards increasing web application productivity. Proceedings of the ACM Symposium on Applied Computing, March 14-17, ACM, Nicosia, Cyprus, pp: 1677-1681.