International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015
ISSN 2091-2730

Analysis And Selection Of Appropriate Activities Based On Reliability And Security Engineering For Software Development

Rida Ghafoor Hussain, Muhammad Nadeem Majeed

Department of Software Engineering

University Of Engineering And Technology, Taxila, Pakistan.

rida_ghafoor@yahoo.com, nadeem.majeed@uettaxila.edu.pk

Abstract— The most significant phase in software progress process is Requirement engineering .The key subject that affects the accomplishment time and success of projects is defined as RE practices in software engineering. The improvement of client requirement for software reliability, the precision of reliability tools is rising in a new style with the unremitting development of software production technology. Many secure development efforts are carried out in direction of software development lifecycles like security specification languages and processes. This paper simply focuses on software engineering specifications that supports reliability and introduces its development trend.The analysis identifies lack of required properties. In the end, guidelines are provided for the development of secure software and reliability issues are identified.

Keywords— Reliability engineering , secure software development process ,Software Reliability, software testing, software engineering, Software security, software security requirements engineering, Reliability engineering.

INTRODUCTION

The software based systems are more and more dependent on projects in this IT industry. Such systems require specific standards to be followed for less failure and the requirement gathering must be strong enough leading to success of software .Of the most significant standards is reliability. Therefore research on reliability has become important as the problems related to reliability of software are increasing in development industry [1].

In SDLC, security vulnerabilities are at higher risk because software security is not considered as major quality standard in earlier stages of development [2]. The phenomenon of security based software development is to prevent software vulnerabilities by taking security measures throughout the software lifecycle i-e in requirement gathering, designing, development and testing. Secure development methods and procedures are developed for software production.SSD methods comprises of security assessment and assurance techniques, security software specification languages and processes. Security measures in SDLC are different from application security. By application security it means security measures after deployment of application. It normally includes, firewalls, intrusion detection and prevention, antivirus etc [2].The purpose is to develop an SDLC that satisfies security and reliability as development of patch for error removal can be upto 200 times more classy [3] than fixing the defect as soon as it is introduced. This paper highlights the basic contents of software reliability and projected some issues regarding software reliability in software engineering. The target is to introduce a software activities that covers all solutions of present security issues .The identified properties can be useful in conversion of one security language into another. Such a conversion is predominantly functional when a client of a language intends to apply security tools made for further languages.

LITERATURE REVIEW

The typical description of software reliability [5] is the prospect of no failures in precise accepted unit [4]. Reliability-based Software engineering is a practical discipline that decrease the risk of unsatisfied user requirement in software development[4]. Specification of availability ,veracity and privacy is the main objective of security requirements. Normally, security features are identified for software design and production [6].G. McGraw [2] in " Software Security: Building Security In" have briefly described Specification of abuse cases and security requirement in software development lifecycle. Abuse cases can be used to derive test cases. Static code analysis tools are utilized and secure design guidelines are provided. M. Howard and S. Lipner [7] in "The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software" analyze identification of interfaces, security objectives and required security features in software development lifecycle. Definition of exit criteria, Identification of critical components, attack surface, design methods, and completion criteria are also mentioned. I. Flechais, C. Mascolo, and M.A. Sasse [8], in "Integrating Security and Usability into the Requirements and Design Process," have done identification of assets in high level security requirements. A. Apvrille and M. Pourzandi [9],in "Secure Software Development by Example," proposed use of a secure

programming language in software development, avoiding buffer overflow formatting string vulnerabilities. L. Futcher and R.v. Solms [10], in "SecSDM: A Model for Integrating Security into the Software Development Life Cycle," explained Modeling using flexible modeling framework (FMF) considering security issues. security code scanning tools are also listed. A security checklist is provided describing potential items to guide development.(CLASP) [11] focused on identification of attackers and attack surface. The research also describes annotation of class diagrams with security information. M. Essafi, L. Labed, and H.B. Ghezala [12],in "S2D-ProM: A Strategy Oriented Process Model for Secure Software Development," have briefly described security modeling language and model checking in SDLC.N. Davis [13],in "Secure Software Development Life Cycle Processes: A Technology Scouting Report" have done state machine design and verification for use in software development and production. Renzuo Xu [14], in "Software Reliability Engineering" explained that software reliability plays a vital role as software quality feature. The software is based on the instructive attainment, understanding, mind behavior , cognitive skill, progress environment and specialized principles , so it indicates that failure can't be avoided. Xizi Huang[15],in Progress Review of Software Reliability have briefly described that SFMEA ,SFTA, SSCA, Petri net analysis, Failure-recover, fault-tolerant ,fault prevention etc can be analyzed through these issues in software reliability design [15].

METHODOLOGY

If Reliability is very important quality attribute of software development .As the production of software depends on experience, cognitive ability ,software production and development environment etc, so chances of failure are more[14].The problems that need to be solved are how to overcome these failures and what are the preventive measures .Software reliability can be defined as probability of software development without any failure[5].The contents of software reliability are as follows:

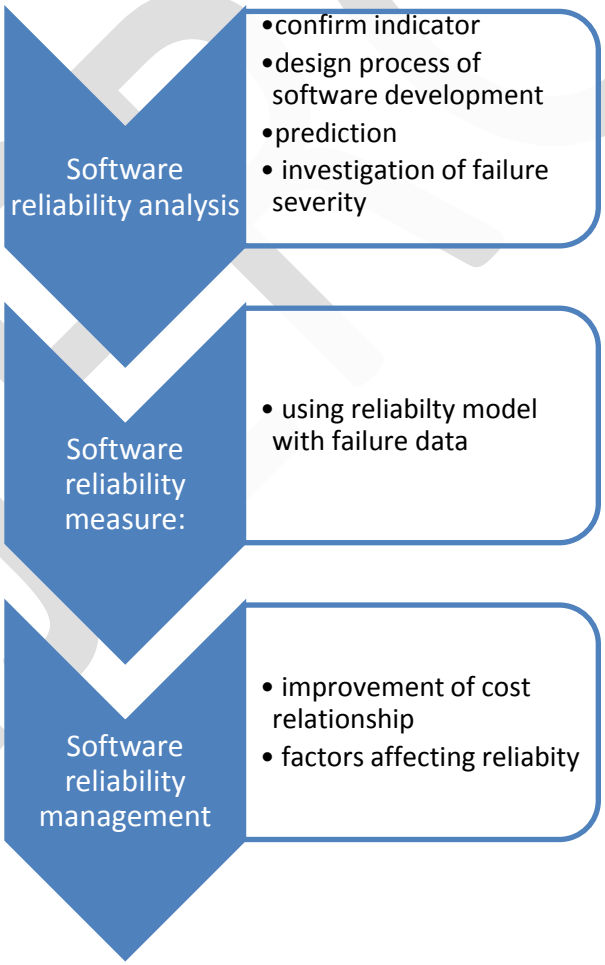

Figure 5: Software Reliability Content

Generally issues with reliability begins at the start of software development. So, software reliability can be divided into five phases:
Hence following problems can be resolved during implementation process of software reliability engineering.

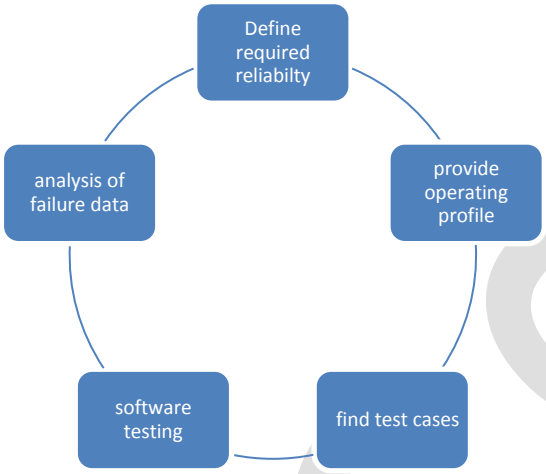

Figure 6: Software Reliability Phases

1) Software Reliability Metrics.
Gray box testing and accelerated test for software reliability assessment method.
2) Analysis and design of software reliability
The confirmation of forecasting and stopping failure comes from information response in structure plan phase which is the essential attribute of reliability engineering.
3) Software Testing and Verifiability.
Automatic generation of test cases, scripts and test data and profile operation .
4) Software Reliability Management.
The management controls development ,purchase and reuse of software, alongwith adapting to changes and confirmation of factors that affects software reliability.
On the other hand, the availability and confidentiality of the software product should be preserved is the key objective of security based specification. Research has reported that security requirement engineering is based on security based requirement engineering specifications and processes. This paper focus on analyzing those security properties that must be a part of software development and production. Security specifications are described by many languages in the literature. Security requirement is a term used to define those requirements which if not implemented causes vulnerability. In other words, specifications that identify attacks on software development and production leading to failure. Similarly, a process is desirable for derivation of these requirements. This process is security requirement engineering process. The most important idea of this process is to make out security requirements activities that can practically make certain security of the software produced. The security processes should have the following activities. The activities are analyzed from requirement engineering processes based on security [11, 6, 16, 17-19].Some activities should be performed iteratively to meet security satisfaction .
R1. High level and low level (e.g., password length) functional specifications and identification of environment for software implementation
R2. Finding out resources and their valuation.
R3. Identification of users, attackers and their interest in the software.
R4. Identification of capabilities and possible threats from attackers.
R5. Specification of misused scenarios and use cases.
R6. Identification of security goals , mechanisms, constraints and policies derived by negotiating with the stakeholders .
R7. Identification of security errors and characterization of exit criteria depending on state of the software(calculation of security state by using security index [7]).
R8. Risk and cost/benefit analysis.
R9. cataloging and prioritization of security requirements of low level.
R10. Inclusion of low level requirements based on security in software.

CONCLUSION

Software products are not competitive because reliability management can't be introduced in production and design. These flaws limit development of software based on reliability engineering. Enforcement of these measures can be helpful in development of reliability engineering. Identification of 10 vital activities are carried out in this research. As a future work, a software model can be generated that could figure out reliability issues mentioned above and implement these activities for security success in software development. .

REFERENCES:

[1] M R Lyu. Handbook of software reliability engineering, New York, McGraw-Hill and IIEE Computer Society Press, 1996
[2] G. McGraw, Software Security: Building Security In, Addison Wesley, 2006.
[3] J. Juerjens, Secure Systems Development with UML, Springer, 2005.
[4] John D M. Software reliability engineering. Hanke .Beijing: China Machine Press 2003, 1-2
[5] Musa J. D., Iannino A, Okumoto K. Software reliability: measurement, prediction, application [M]. McGraw-Hill, Columbus, 1987
[6] C.B. Haley, J.D. Moffett, R.Laney, and B. Nuseibeh, "A Framework for Security Requirements Engineering," In Proceedings of the International Workshop on Software Engineering for Secure Software (SESS'06), Shanghai, China, ACM Press, 2006, pp. 35-41.
[7] M. Howard and S. Lipner, The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software, Microsoft Press, 2006.
[8] I. Flechais, C. Mascolo, and M.A. Sasse, "Integrating Security and Usability into the Requirements and Design Process," International Journal of Electronic Security and Digital Forensics, Inderscience Publishers, Geneva, Switzerland, 2007, vol. 1, no. 1, pp. 12-26.
[9] A. Apvrille and M. Pourzandi, "Secure Software Development by Example," IEEE Security and Privacy, IEEE CS Press, 2005, vol. 3, no. 4, pp. 10-17.
[10] D. Gilliam, J. Powell, E. Haugh, and M. Bishop, "Addressing Software Security Risk and Mitigations in the Life Cycle," In Proceedings of the 28th Annual NASA Goddard Software Engineering Workshop (SEW'03), Greenbelt, MD, USA,2003, pp. 201-206.
[11] OWASP CLASP Project, http://www.owasp.org/index.php/Category:OWASP_CLASP_Project. Last Accessed March 2009.
[12] M. Essafi, L. Labed, and H.B. Ghezala, "S2D-ProM: A Strategy Oriented Process Model for Secure Software Development," In Proceedings of the 2nd International Conference on Software Engineering Advances (ICSEA'07), Cap Esterel, French Riviera, France, 2007, p. 24.
[13] N. Davis, "Secure Software Development Life Cycle Processes: A Technology Scouting Report", Technical Note CMU/SEI-2005-TN-024, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.
[14] Renzuo Xu, Software Reliability Engineering, Beijing, Tsinghua University Press, May, 2007
[15] Xizi Huang, Progress Review of Software Reliability in 1990s, Equipment Quality, Vol.9, September, 2000
[16] H. Mouratidis, P. Giorgini, and G. Manson, "When Security Meets Software Engineering: A Case of Modeling Secure Information Systems," Journal of Information Systems, Elsevier Science, 2005, vol. 30, no. 8, pp. 609-629.
[17] J Viega, "Building Security Requirements with CLASP," In Proceedings of the 2005 International Workshop on Software Engineering for Secure Systems (SESS'05), St. Louis, MO, USA, 2005, pp. 1-7.
[18] N.R. Mead, E. Hough, and T. Stehney, Security Quality Requirements Engineering (SQUARE) Methodology, Technical Report CMU/SEI-2005-TR-009, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.
[19] D. Mellado, E. Fernandez-Medina, and M. Piattni, "A Common Criteria-Based Security Requirements Engineering Process for the Development of Secure Information Systems," Computer Standards and Interfaces, Elsevier Science, 2007, vol. 29, pp. 244-253