2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications

Combining IID with BDD to Enhance the Critical Quality of Security Functional Requirements

Sen-Tarng Lai
Dept. of Information Technology and Management,
Shih Chien University,
Taipei, 10462, Taiwan
e-mail: stlai@mail.usc.edu.tw

Fang-Yie Leu, William Cheng-Chung Chu
Dept. of Computer Science,
Tunghai University
Taichung, 40704, Taiwan
e-mail: leufy@thu.edu.tw

***Abstract*—In software system, functional requirements are primary system requirements. Client cannot explicitly depicted security requirements and the development team is hard to understand security requirements, makes security requirements difficult to specific implant software system. In software system development process, security requirements often neglected and ignored. However, the cost of correcting security flaws in maintenance phase is over 100 times in requirements phase. Can't effectively improve the system security, enterprises loss is bound to continue to expand. In order to enhance the security of software system, this paper combines the popular software development methodology IID (Interactive and Incremental Development) with BDD (Behavior Driven Development), institutionalized requires functional requirements must be integrated into security requirements. And, using BDD process features to evaluate the major quality of security functional requirements. Timely identifies and modifies the quality defects of security functional requirements item, effectively enhance the security of software systems.**

Keywords-BDD, IID, security requirements, functional requirements, security functional requirements

I. INTRODUCTION

In software system, functional requirements are primary system requirements items but security requirements often are omitted or ignored. Data presented by Fortify in 2008 indicate that the cost of correcting software security flaws in maintenance phase is over 100 times than the cost in requirements phase [1]. Can't effectively reduce software system security vulnerabilities, enterprise or organization loss bound to continue to expand. Because the client can't explicitly depicted security requirements and the development team hardly understood security requirements. In software system development process, security requirements often are neglected or ignored. The experienced system analysts can be handily to complete the analysis and description of functional requirements. But, about the security requirements analyzing, most of system analysts can't be handily to complete. Client doesn't care the security requirements and system analysts lack of practical security requirements analysis experience [2]. In addition, in system analysis phase, security requirements still have not a set of standard and suitable documents representation to cause the security requirements difficult to specific implant software system. Continuous learning and participation practical operations can effectively increase analysis experience of security requirements. However, in order to enhance system security, planning the suitable security requirements representation and analyzing methodology are the worthy of deep exploration topic.

In 1970s, waterfall development model have been proposed and applied a long period time, it always is a favorite model in large organizations and enterprises. Waterfall model requests exact for each phase development documents. The documents except have high completeness and correctness, and must be passed formal review activity before entering next development phase [3]. In addition, for the discovered defects in review activity, waterfall model has flexibility and features to feedback to the related phases. Timely modification and revision can avoid the problems and defects extension. However, waterfall model does not allow iterative and incremental requirements. It's meaning that overall system requirements must be accomplished, then development procedure can enter follow-up phase. In mobile network age, the environment changes quickly and the information product upgrades frequently. High flexibility software development model IID (Interactive and Incremental Development) becomes a new trend [4]. Incremental functional or security requirement items can reduce unclear scenario description, ambiguous state definition to enhance requirements changeability and high accomplishment ratio.

IID became the mainstream of software development in recent decade [4]. Based on IID, many popular software methodologies and processes have been proposed for example Unified Process and Agile Process. BDD (Behavior Driven Development) is a suitable process for agile software development [5, 6]. One of major advantages of BDD is test cases must to be preplanned and designed before implementation [7]. Therefore, based on BDD process, the behavior description should have high quality to continue the follow-up development operations. Quality defects of behavior description can be identified timely. For ensuring system security, behavior constraint, data and asset protection should be considered by security requirements.

978-1-4799-4173-5/14 $31.00 © 2014 IEEE
DOI 10.1109/BWCCA.2014.78

This paper institutionalized requires functional requirements must be integrated into security requirements and be called the Security Functional Requirements (SFR). SFR item uses BDD processes can collect key quality factors to timely identify SFR item quality defects. In this paper, based on BDD feature, collects SFR item quality factors, proposes the SFR item quality measurement model and plans a BDD-based Quality Enhancement Procedure (BQEP) for improving SFR item quality. Applied the quality measurement model, SFR item quality defects can be identified. Utilize BQEP with quality improvement operation, SFR item quality can be enhanced continuously and software security can be enhanced. In Section II, studies importance of security requirements and major features of BDD. In Section III, discusses the difference between functional and security requirements, and quality factors of SFR item. In Section IV, based on quality measurement model, plans a SFR item quality improvement mechanism. In Section V, evaluates the efficiency of BQEP. Finally, in Section VI, describes advantages of the BQEP and the future work.

II. SECURITY REQUIREMENTS, IID AND BDD

A. Security requirements issues

Networks and information systems have become the focus of people life. However, they also brings unpredictable security crisis. According to CERT/CC latest statistics data show that from 2005 to 2008 third quarter, analysis of software vulnerabilities total cases are 27,346 [8] (shown in TABLE I). More security vulnerabilities of information system cause the more serious security crisis. There are many international groups and organizations (SANS (security training, certification and research institutions), OWASP (Open Web Application software security program, Open Web Application Security Project) very concern the software and Web App security. They continuously published the key of Web App security vulnerabilities and defects: SANS Top-20 Security Risks [9] with the OWASP Top 10 security vulnerabilities [10], to help reduce software and Web App security risks.

Table 1. 2005~2008 Q3 software vulnerabilities statistics data

Years	Volumes
2005	5,990
2006	8,064
2007	7,234
Q1-Q3, 2008	6,058
Summary	27,346

*Data source: CERT/CC

Security vulnerability of software system cause the loss and crisis which is hard expectations and evaluations. In order to effectively reduce loss of software security vulnerability, it is necessary to overall consider the security requirements of software development process [11, 12, 13]. Security requirements are the basis of secure software development operations and the phase review and system acceptance activities. The experienced system analysts can be handily to complete the analysis and description of functional requirements. However, about the security requirements analyzing, most of system analysts can't be handily to complete. It is because that system analysts lack of practical security requirements analysis experience. Besides, in system analysis phase, security requirements still have not a set of standard and suitable documents format. Continuous learning and participation practical operations can effectively increase experience of security requirements. However, a suitable security requirements representation and analysis methodology is a critical issue for improving system security. Summarize security requirements defects, four major deficiencies should be modified by suitable process:

- Can't control the item coverage and size.
- Lack suitable representation format.
- Lack verification and validation capability.
- Lack communication capability.

B. IID and BDD

The quality of system requirement specification must keep the characteristics of correctness, completeness, and consistency, otherwise the requirement phase documents will be requested to revise or redo. Existing development models have flexible modified the requirement specification style. For instance, in IID method, users can incrementally provide the requirements, i.e., it is unnecessary to deploy entire requirements in a certain period of time [4]. Therefore, software development risk can be greatly reduced. In the rapid prototyping model, prototype products are quickly developed; this helps in quickly and accurately determining user requirements. Prototypes help in identifying incomplete, inconsistent, or incorrect requirement specification. Spiral development model primarily focusses on development risks and has high flexibility. In the development process, the spiral method offers adaptability for encountering different risks. Further, if the development risks can't be effectively reduced then the spiral model recommends terminating or aborting the software project. In February 2001, seventeen software developers met at a ski resort in Utah, USA for two days and drafted the Manifesto for the agile software development process [14]. Many of participants had previously authored their own software development methodologies, including Extreme programming, Crystal, and Scrum [15, 16]. Agile software development proposes several critical viewpoints:

(1) The development process should not involve more of the analysis and design phase operations and documents.
(2) The programming phase should be started as soon as possible because a workable software is more practical than a development document.
(3) The software being developed should have and support high modifiability.

(4) The cooperative relationship between the system developers and system users should be enhanced.

BDD (Behavior Driven Development) is a suitable process for agile software development method. Advantages of BDD (shown as Figure 1) [7, 17] include:

- In software development phases, test cases can be preplanned and designed.
- Assist to build a consensus among the stakeholder, developer and user.
- Very concern the communication between clients and development teams.
- Emphasize fixed time features or simple requirements that can be accomplished in three weeks.
- Provide automatic regression testing.
- Assist behavior description revision and design refactor.

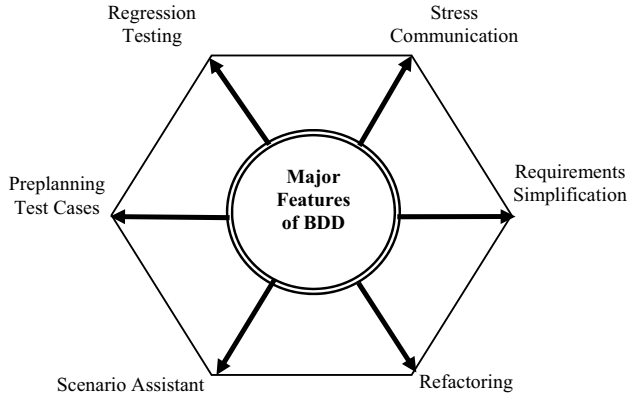

Figure 1. BDD major features

BDD process and IID methods have several well features can help modify security requirements deficiencies. In BDD process, user stories are broken down for the fixed time and simple requirements that should be accomplished in 3 or 4 weeks. The fixed time features and simple requirements can effectively modify the second deficiency of security requirements. BDD borrowed the concept of the ubiquitous language from domain driven design. Ubiquitous language is a semi-formal language that is shared by all members of a software development team [17]. Apply the ubiquitous language of domain driven design, can modify the third deficiency of security requirements. In BDD process, user stories (behaviors) need plan test cases before design and implementation. Therefore, user stories have to build a consensus of among the user, the developer and the stakeholder, and client representative works with the development team at all time. Building a consensus and all time participation can effectively improve communication capability and modify the fourth deficiency of security requirements.

III. IMPORTANCE OF SECURITY FUNCTIONAL REQUIREMENTS

A. Functional requirements and security requirements

System requirements can commonly separated into Functional Requirements and Non-Functional Requirements (NFRs). Security requirements belong to NFRs. Generally, system requirements just focus on the functional requirements deployed, but security requirements often are omitted or ignored. However, in many cases, it is necessary to combine functional and security requirements. Related definitions of the function requirements and the security requirements are described as follows:

(1) The official definition for a functional requirement specifies what the system should do: "A requirement specifies a function that a system or component must be able to perform.". Functional requirements specify specific behavior or functions, for example: "Display the heart rate, blood pressure and temperature of a patient connected to the patient monitor." Functional requirements describe the behaviors (functions or services) of the system that support user goals, tasks or activities. Non-functional requirements include constraints and qualities.
   - System qualities are properties or characteristics of the system that its stakeholders care about and hence will affect the satisfaction degree with the system.
   - A constraint is a restriction on the degree of freedom we have in providing a solution.

(2) Firesmith identifies 10 types of security requirements [18]. Major purpose of Security requirements is specify the capability in protecting system resources such as data, and program from malicious attack.

There are many difference attributes between functional requirements and security requirements (shown as Table II). Data presented by Fortify in 2008 indicate that the cost of correcting security flaws at the requirements phase is up to 100 times less than the cost of correcting security flaws in maintenance phase [1]. It's mean that identifies and corrects the security flaws in early phase can save great amount cost. Client and analyst have higher understanding of functional requirements than security requirements. Methodology maturity level of functional requirements is higher than the security requirements. Security requirements should suitably combine with functional requirements to show the security requirements importance and necessity. For improving security requirements representation, analysis technology, development operations and verification capability, functional requirements should combine with security requirements to be the security functional requirements.

TABLE II. Difference analysis table between Functional Requirements and Security Requirements attributes

Requirements / Attributes	Functional Requirements	Security Requirements
System impact	High	Very high
Understanding of client	High	Low
Understanding of analyst	High	Low
Methodology maturity level	High	Low

B. Representation of SFR

SFR are the basis of secure software development process, and the major documents for phase reviews and acceptance testing. The experienced system analysts can be handily to complete the analysis and description of functional requirements. However, about the security requirements analyzing, most of system analysts can't be handily to complete. It is because that system analysts lack the experience on learning, using and practical security requirements analysis. Besides, a use case is a popular methodology used in system analysis to identify, clarify, and organize system requirements. Use case model is accepted by most of system clients, developers and stakeholders. Many scholars and experts think that use cases are very suitable for functional requirements analysis, but are not suitable for NFR analysis. Although use case is not suited to identify, clarify, and organize security requirements. Use case model still is recognized by most of system clients, developers and stakeholders. Therefore, several new analysis models for analyzing security requirements almost are also based on the use case model. The analysis models of security requirements include misuse cases [19], abuse cases [20] and security use cases [21]. Misuse case model is one of the most frequently be discussed and used security requirements model.

Use case diagram and scenario description usually specify specific behavior and function. Misuse case diagram and scenario description specify the capability in protecting system resources such as data, and asset from malicious attack (shown as TABLE III). Representation of SFR items can integrate with use case, misuse case diagrams and behavior scenario, protection scenario descriptions (shown as Figure 2).

TABLE III. Difference table between Functional Requirements and Security Requirements representation

Requirements / Representations	Functional Requirements	Security Requirements
Diagram	Use case	Misuse case
Scenario description 1	System behavior	Constraint
Scenario description 2	System function	Data protection
Interface definition	High impact	Low impact
semi-formal language	Ubiquitous language	Constraint language

C. Critical quality factor of SFR

Combining IID with BDD for secure software system development, SFR iteem can be drapped up and should keep five major features to increase secure software quality. Five major features of SFR are described as follows (shown as Figure 3):

(1) Common recognition: Major purpose of SFR to is improve system security. Therefore, SFR items must become the common recognition and acceptable uniform documents among clients, system developers and stakeholders. SFR should have the quality characteristic of communication capability.

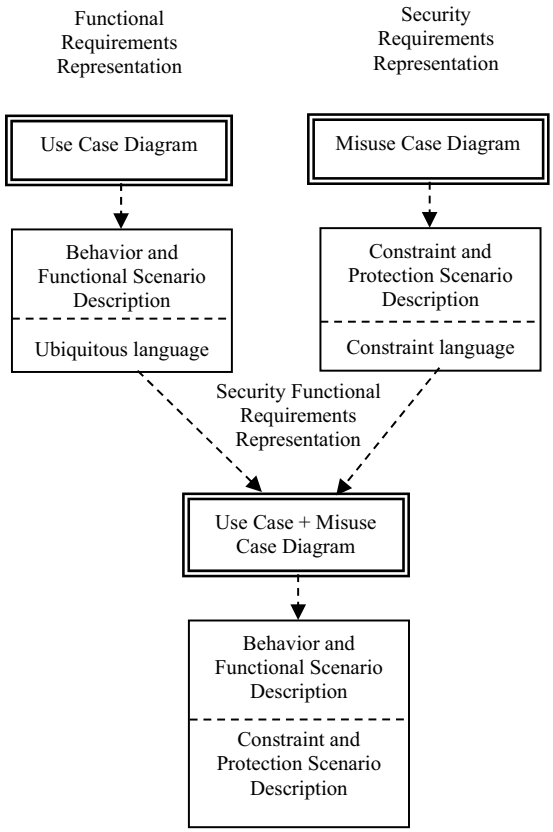

Figure 2. Representation of Functional Requirements, Security Requirements and SFR

(2) Development basis: SFR is basis of secure software system development. Therefore, SFR items must have an uniform and standard documents format. SFR is an important communication media among client, developers and stakeholders. And, in system development process, ensuring the SFR item can be embedded into the system.

(3) Review and validation capability: In software development process, each iteration has to pass review activity to ensure the process quality. In the transformation phase, the system must pass the acceptance test to ensure the system quality. SRF item is critical basis for review activity and acceptance test.

(4) Low complexity: In agile process, each user story must be implemented and accomplished in 2 to 4 weeks. Therefore, SFR items should consider the security and functional complexity. Low complexity SFR item can suit for the feature of agile process and BDD.

(5) Low coupling: Refactoring is a major feature of agile process. For keeping the refactoring feature, SFR item should have low coupling. Low coupling SFR can increase the modifiability to adapt to requirement revisions or environment changes.

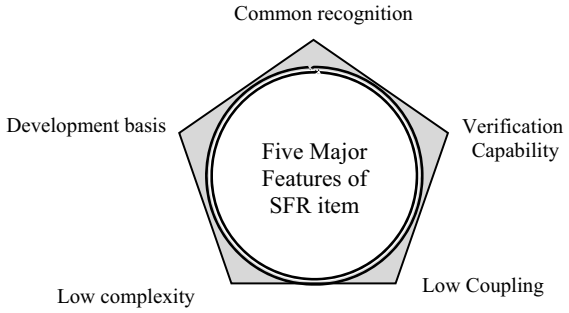

Figure 3. Five major features of SFR items

High quality SFR items should have three critical quality characteristics which includes communication, assurance and change. Discuss three critical quality characteristics and describe related quality factors as follows:

(1) Communication quality: SFR item communication quality should consider scenario description and interface definition two factors:
- Scenario description quality: SFR are the basis of follow-up development. Therefore, SFR item scenario description must have correctness, completeness and consistency and readability characteristics to create a common consensus among the user, the developer and the stakeholder. SFR item scenario description should consider correctness, completeness and consistency and readability factors.
- Interface definition quality: Interface definition is a basis of the SFR item communication and integration. Well defined interface can build the interactive relationship between SFR items and others items. SFR item interface definition should consider explicitness and consistency factors.

(2) Assurance quality: SFR item assurance quality should consider phase review capability and system acceptance capability two factors:
- Phases review capability: Requirement documents are the base of phases review activity. SFR item must have the capability to plan the review checklists and test cases for testing. Phase assurance activity can identify the quality problems and defects in early phase to reduce extra development resource expense.
- System acceptance capability: Requirement documents are major basic of system validation activity. SFR item must have the capability to plan acceptance test cases for acceptance testing. System acceptance activity can identify the quality problems and defects before system delivery to reduce extra disputes. System acceptance capability should consider the acceptance checklists planning and acceptance test case design factors.

(3) Change quality: SFR item change quality should consider low complexity and high modularity two factors:
- Low complexity: SFR item should have simplify and low complexity to reduce change impacts. Environment changes or requirement revisions, SFR item can adapt the environment or requirement to modify or adjust quickly. Low complexity should consider requirement item size, tables, files, data items and operation behaviors quantified factors.
- High modularity: SFR item should have low coupling and high cohesion to overcome change affects. SFR item can be easy isolated for adapting the environment changes or requirement revisions and to modify or adjust quickly. High modularity should consider requirement items coupling and cohesion affect factors.
- Traceability: SFR item documents for each development phase must have cross-reference relationship to handle SFR item change requests. Traceability is based on documents cross-reference relationship that can assist correct and complete revision operation, and revised quality assurance activities.

IV. BDD-BASED SFR ITEM QUALITY IMPROVEMENT PROCEDURE

In this section, we propose a quality measurement model for SFR item and based on the measurement model establish a SFR item quality improvement procedure.

A. SFR item quality measurement model

Single factor or measurement can only measure or evaluate the specific attribute item. In order to effectively monitor and assess the quality characteristic problems and defects, individual factor or measurement should to make the appropriate combination [22, 23, 24]. Two kind of metric combination models are Linear Combination Model (LCM for short) [22, 23 25] and Non-Linear Combination Model (NLCM for short) [22, 23, 26]. NLCM has higher accuracy measurement than LCM. However, LCM has high flexibility, more extensible and easy formulation than NLCM. For this, in this paper, LCM is applied to SFR item quality measurement. The different SFR activities have different quality metrics be shown. Therefore, before using the linear combination model, the quality factors must be collected and normalized. Refer to predefined weight values and four combination formulas, basic layer quality factors can be combined into three quality measurements. Finally, the formula combines three critical quality measurements into an indicator of SFR item quality measurement. Four formulas described as follows:

(1) Communication Quality Measurement (CmQM) is combined scenario description factors and interface definition factors. CmQM generation steps describes as follows:

Step1. SFR item scenario description quality is combined items documents correctness, completeness, consistency, and readability etc. quality factors

Step2. SFR item interface definition quality is combined requirement items compliance, completeness and

consistency factors.

Step3. Combining scenario description and Interface definition quality into a CmQM, shown as Formula (1):

CmQM: Communication Quality Measurement

SDQM: Scenario Description Quality Metric W_1*: Weight of SDQM*

IDQM: Interface Definition Quality Metric W_2*: Weight of IDQM*

$$CmQM = W_1 * SDF + W_2 * IDF \quad W_1 + W_2 = 1 \qquad (1)$$

(2) Assurance Quality Measurement (AQM) is combined phase review factors and system acceptance factors. AQM generation steps describes as follows:

Step1. SFR item phases review capability is combined review checklists, functional and security test case planning factors.

Step2. SFR item system acceptance capability is combined acceptance checklists and acceptance test cases factors.

Step3. Combining scenario description and Interface definition quantified values into an AQM, shown as Formula (2):

AQM: Assurance Quality Measurement

PRQM: Phases Review Quality Metric W_1*: Weight of PRQM*

SAQM: System Acceptance Quality Metric W_2*: Weight of SAQM*

$$AQM = W_1 * PRQM + W_2 * SAQM \quad W_1 + W_2 = 1 \qquad (2)$$

(3) Change Quality Measurement (ChQM) is combined complexity, modularity and traceability. ChQM generation steps describes as follows:

Step1. SFR item complexity is combined size and related data items amount factors.

Step2. SFR item modularity is combined cohesion and coupling factors.

Step3. SFR item traceability is combined items cross-reference table and phases cross-reference table two quality factors.

Step4. Combining complexity, modularity and traceability into a ChQM, shown as Formula (3):

ChQM: Change Quality Measurement

CQM: Complexity Quality Metric W_1*: Weight of CQM*

MQM: Modularity Quality Metric W_2*:Weight of MQM*

TQM: Traceability Quality Metric W_3*: Weight of TQM*

$$ChQM = W_1 * CQM + W_2 * MQM + W_3 * TQM \quad W_1 + W_2 + W_3 = 1 \qquad (3)$$

Finally, combining CmQM, AQM and ChQM into a SFR item quality indicator, shown as Formula (4).

ISFRI: Indicator of SFR Item Quality Measurement

CmQM: Communication Quality Measurement W_{cmq}*:Weight of CmQM*

AQM: Assurance Quality Measurement W_{aq}*:Weight of AQM*

ChQM: Change Quality Measurement W_{chq}*:Weight of ChQM*

$$ISFRI = W_{cmq} * CmQM + W_{aq} * AQM + W_{chq} * ChQM \quad W_{cmq} + W_{aq} + W_{chq} = 1 \qquad (4)$$

Using three combination formulas to generate three major SFR item quality measurements and combine three SFR item quality measurements into a SFR item quality indicator. Through seven groups of quality factor and four combination formulas to generate a SFR item quality indicator called as the SFR item quality measurement model. The architecture of SFR item quality measurement model is shown in Figure 4.

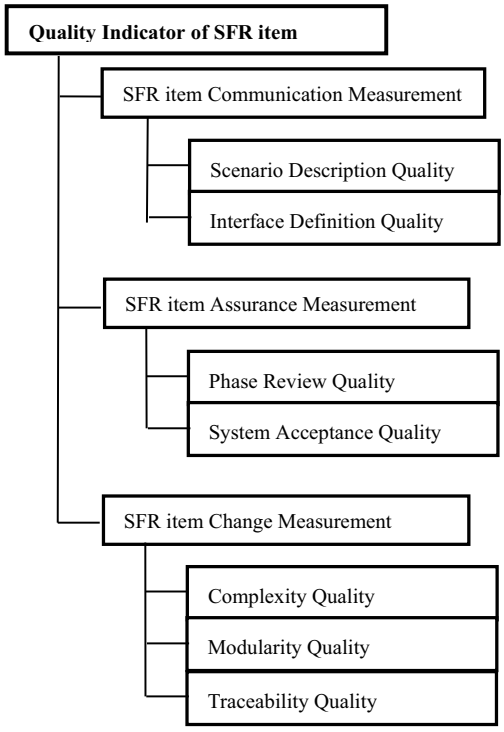

Figure 4. Architecture of SFR item quality measurement model

B. SFR items quality improvement procedure

PDCA model is an approach for the control and continuous improvement of processes and products. In this paper, based quality measurement model, plans a BDD-based Quality Enhancement Procedure (BQEP) for improving SFR item quality. BQEP is divided into four major phases that include SFR items drafting phase, measurement phase, identification phase and revision phase. The detailed operation of the SFR items BQEP (shown as Figure 5) describes as follows:

(1) SFR drafting phase: In drafting phase, based on IID method and BDD process steps, SFR items are drawn up.

(2) Quality measurement phase: Collecting, quantifying and combining the basic layer quality factors can generate high layer quality measurement. Combining high layer quality measurements can the indicator of

SFRI. The quantified quality data can help us identify difference layer defect SFR items.

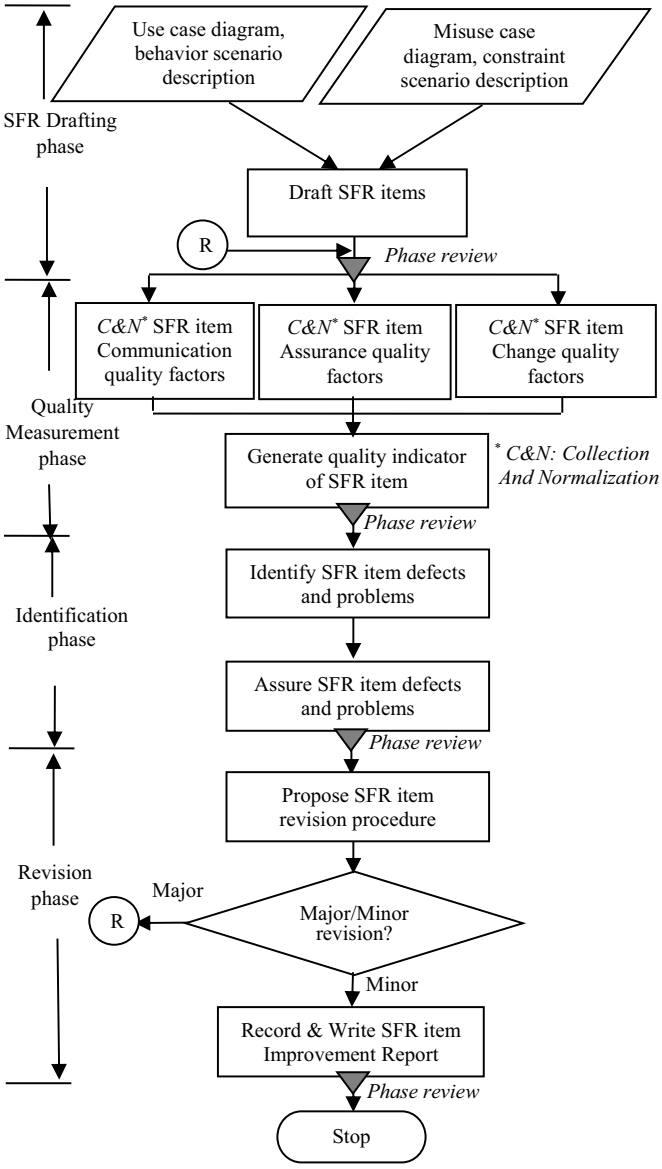

Figure 5. SFR item BQEP operation flowchart

(3) Identification phase: Based on SFR quality baseline, SFR defect and problem can be identified by rule-based approach.
- If defects belong to SFR item Communication Measurement, then the basic layer SFR item communication quality factors that include scenario description or interface definition should be inspected. According to inspection report, SFR item communication quality defects should be corrected.
- If defects belong to SFR item Assurance Measurement, then the basic layer SFR item assurance quality factors that include phase review quality or system acceptance quality should be detected. According to inspection report, SFR item assurance quality defects should be corrected.
- If defect belong to SFR item Change Measurement, then the basic layer SFR item change quality factors that include requirement item complexity, modularity and traceability should be detected. According to inspection report, SFR item change quality defects should be corrected.

(4) Revision phase: Any type revision need record revised items to check improvement result. However, accomplished major revision, SFR items must re-measure quality model to assure improvement result.

V. CONCLUSION

Information systems become the focus of people life also causing the system security becomes an important and interested issue. The network intrusion, malicious users, virus attack and system security vulnerabilities have continued to threaten system normal operation, making software system security encounter serious test. In this paper, we discuss functional and security requirements of software system, and survey security functional requirements presentation and quality factors. IID methodology and BDD process are effective software development models for increasing system successes ratio. In order to enhance the security of software system, this paper combines the popular software development methodology IID with BDD, institutionalized requires functional requirements must integrated into security requirements. Based on BDD process, the paper proposes a quality measurement model for SFR items to help identify SFR item quality defects. According to quality defects of SFR item, a BQEP is designed for continuous enhancing SFR item quality. The SFR item BQEP owns three advantages describe as follows:

- Based on quality factors and quality measurement model, SFR item defects can be identified.
- Model formula has clear, simple and high flexibility.
- Apply BDD and IID major features to enhance SFR item quality.

ACKNOWLEDGEMENTS

This research was supported by the Shih Chien University 2013 research project funds (Project No.: 102-05-04010)

REFERENCES

[1] Meftah, B., "Business Software Assurance: Identifying and Reducing Software Risk in the Enterprise," 9th Semi-Annual Software Assurance Forum, Gaithersburg, MD, October 2008. https://buildsecurityin.us-cert.gov/swa/downloads/Meftah.pdf.

[2] Tondel, I.A.;Jaatun, M.G.; Meland, P.H. "Security Requirements for the Rest of Us: A Survey", IEEE Software,vol. 25, no.1, 2008, pp.20-27.

[3] Pressman, R. S. "Software Engineering: A Practitioner's Approach," McGraw-Hill, New York, 2010.

[4] Larman C. and Basili, V.R."Iterative and incremental development: A brief history", IEEE Computer, June 2003, pp.47-56.

[5] Bellware, Scott, Behavior-Driven Development, Code Magazine, June 2008. (Retrieved 10 May 2012)

[6] Keogh, L., Translating TDD to BDD, 2009, by (http://lizkeogh.com/2009/11/06/translating-tdd-to-bdd/)

[7] Solis, C. and Xiaofeng Wang, A Study of the Characteristics of Behaviour Driven Development, Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference, pp. 383–387.

[8] CERT/CC（http://www.cert.org/stats/cert_stats.html）(2008/12)

[9] The Top Cyber Security Risks (http://www.sans.org/top-cyber-security-risks/) (2010/5)

[10] OWASP Top 10 (https://www.owasp.org/index.php/Top_10_2013-Top_10) (2013/7)

[11] McGraw, G. Software Security – Building Security In, Addison-Wesley, 2006.

[12] Viega, J., McGraw, G., Building Secure Software, Addison-Wesley 2004.

[13] N. Davis, W. Humphrey, Jr. S. T. Redwine, G. Zibulski, and G. McGraw, "Processes for Producing Secure Software," IEEE Security & Privacy, vol. 2, no. 3, 2004, pp. 18-25.

[14] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., …Thomas, D. (2001). Manifesto for agile software development Retrieved 17 February, 2003, from http://www.agilemanifesto.org.

[15] Szalvay, V. "An Introduction to Agile Software Development," Danube Technologies Inc., 2004.

[16] Schach, S. R., "Object-Oriented Software Engineering," McGraw-Hill Companies, 2010.

[17] North, Dan, Introducing BDD, Available at: http://dannorth.net/introducing-bdd, March 2006. (Accessed June 13, 2014])

[18] Firesmith, D, "Engineering Security Requirements," Journal of Object Technology, vol. 2, no.1, 2003, pp.53–68.

[19] Sindre G. and Opdahl, A.L. (2005) "Eliciting Security Requirements with Misuse Cases," Requirements Eng., vol. 10, no. 1, pp. 34–44.)

[20] McDermott J. and Fox, C. (1999) "Using Abuse Case Models for Security Requirements Analysis," Proc. Computer Security Applications Conf., IEEE CS Press, pp. 55–64.

[21] Firesmith, D., "Security Use Cases," Journal of Object Technology, vol. 2, no. 3, 2003, pp. 53–64.

[22] Fenton,. N. E. Software Metrics - A Rigorous Approach, Chapman & Hall1991.

[23] Conte, S. D. ,Dunsmore, H. E., and Shen,V. Y., Software Engineering Metrics and Models,Benjamin/Cummings, Menlo Park, 1986.

[24] Daniel Galin, Software Quality Assurance, Addison-Wesley, 2004.

[25] Boehm, B. W., Software Engineering Economics, Prentice-Hall, New Jersey, 1981.

[26] Lai, S. T. and Yang, C. C., "A Software Metric Combination Model for Software Reuse," Proc. of 1998Asia-Pacific Software Engineering Conference (APSEC'98), 1998, pp. 70-77.