Singaporean Journal Scientific Research (SJSR)
ISSN: 2231 - 0061 Vol.4, No.3 pp. 24-35

available at: : http://www.iaaet.org/sjsr

SOFTWARE QUALITY ENHANCEMENT FOR VALUE BASED SYSTEMS THROUGH STAKEHOLDERS QUANTIFICATION

[1]MUHAMMAD IMRAN BABAR, [2]MASITAH GHAZALI, [3]DAYANG N.A. JAWAWI

[1]Research Scholar, Department of Software Engineering, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
[2]Senior Lecturer, Department of Software Engineering, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
[3]Associate Professor, Department of Software Engineering, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
Email: [1]ibmuhammad2@live.utm.my, [2]masitah@utm.my, [3]dayang@utm.my

ABSTRACT

Software quality assurance plays an important role to check the overall quality of the software product especially when a product is a value based system. The valuable software product or product line is tested under strict circumstances to meet the minimum constraints of software quality. This paper focuses on stakeholders, requirements engineering, different testing techniques being applied in software professional environment, issues and current trends to resolve the requirement problems for continuous software quality improvement. This paper presents the criticality of stakeholders, requirements and software testing techniques for software professionals in terms of quality assurance. A model is proposed in order to achieve a high quality value based software application. There is the dire need to integrate stakeholders, requirements and testing in order to evaluate the performance and quality of a value based system. A systematic stakeholder analysis framework does not exist, and there is the need of a systematic framework that may be adopted as a standard. This research also focuses on a systematic stakeholder's identification and quantification framework.

Keywords: *Software Requirements, Stakeholders, Software Quality, Software Testing.*

1. INTRODUCTION

Value Based Software Engineering (VBSE) mainly deals with economic driven systems. The economic driven systems are based on innovation in which a new innovative business idea is proposed for economic leverage. The software systems that are designed for different life domains or industries, like finance, electronics, aviation, medical, mechatronics and other spheres of life, can be termed as valuable systems. These systems are not valuable in terms of finance, but these are also valuable in terms of human service. So the value of a system is counted in terms of its financial outcome and its services to mankind. The existing value based practices in research and in development of value based systems are "done in a value neutral setting in which every <stakeholder>, requirement, use case, object, test case, and defect is equally important" [1]. Software engineers assign an equal priority to all the aspects either human or technical, and this thing leads to low quality applications. Another important problem in the value based practices is "Earned value" systems track project cost and schedule, not stakeholder or business value" [1].

The existing models and approaches consider the value of all at par. They do not distinguish the value of each involved entity in a well-defined way. In value based systems' development an innovative idea is proposed and "yet unknown to the market" [2]. It is very hard to develop an application which is based on an innovative idea because it becomes difficult to understand the business value of the idea [3]. It is very difficult to predict that either the idea should be adopted or not in order to realize it [3]. The realization of an innovative idea must be in time because if the idea is launched by someone earlier then its realization will not be beneficial. So time is also a critical success factor in making fast decisions [2]. The quality of a value-based product is based on return on investment or how much it will pay in terms of its acceptance by a large community. Quality is one of the core issues in value based software engineering applications. The basic purpose of quality is to check the performance of an application or a system in view of the fulfillment of functional and non-functional requirements. On the other hand, the existing planning models are not sufficient for value based system development [4].

There is no decision support for industry professionals that how to use the term value in an effective way [5]. This paper focuses on the value of stakeholders, requirements engineering (RE) and software testing for value based systems. Any software project is initiated by a stakeholder or a set of stakeholders. There are different definitions of stakeholder, but in a simple way the term may be defined as "the people and organizations affected by the application" [6]. Stakeholders play an important role in requirements gathering for value based systems. Different models are presented in order to analyze the stakeholders. The purpose of all these models is to get a set of highly critical stakeholders. Requirements are gathered from stakeholders for realization, and the realization of requirements is verified through software testing techniques. In software, the functional requirements are treated as goals and non-functional requirements as softgoals [7]. Testing is the verification and validation of functional (goals) and non-functional (softgoals) requirements. A strong link or tight bond exists among stakeholders, requirements, testing, verification and validation [8]. These human and technical aspects, like stakeholders, requirements, testing, verification and validation, are taken into consideration as the quality aspects of a value based system.

The link between them is avoided or deliberately missed [9], and it results in flawed applications in terms of delays, cost, functionality, performance and reliability. A strong link among all five may help in developing highly valuable systems.
The major considerable aspects are stakeholders, the implementation of stakeholder's needs and the achievement of optimum quality level using effective testing. In all these quality aspects, the stakeholder is considered as the most important and a key aspect because a stakeholder is an initiator of all activities of a value based software system development. So for value based software system development there is the need to select highly valuable stakeholders for the success of the system. The success of a valuable system is defined in terms of its quality and acceptance by a wider community. Most of the work in value based system development focuses the issues of requirements engineering and system design, and there is less focus on the value of stakeholders. The existing techniques of stakeholders' analysis present a very high level process of identification and quantification. Stakeholders are the nucleus in the process of value based software system development. This is a glaring problem in development of value based systems that due consideration is not given to the value of stakeholders. Mostly the stakeholders are treated in a value neutral fashion. However, this research paper does not cover the problems induced by the technology adoption because each technology has its own issues.

Rest of the paper is divided into 7 sections. In section 2, the importance of stakeholders is described, and drawbacks of different stakeholders' analysis techniques are discussed. Section 3 describes the role of software requirements in software requirements engineering process of value based software systems. Section 4 is comprised of the main software testing techniques and section 5 describes the issues of software testing techniques. Section 6 describes the proposed model. Section 7 describes the proposed processes in order to initiate the model. Section 8 elaborates the stakeholders' identification and quantification (SIQ) method. Section 9 is based on selected case studies for the proposed SIQ method. Section 10 describes the discussion and future work while section 11 is about conclusion.

2. STAKEHOLDERS

The main purpose, of stakeholders' analysis, requirements engineering, testing, verification and validation approaches, is to check the overall quality of the software system or a component. In order to obtain a complete, consistent, conflict free and important set of requirements there is the need to select a right set of stakeholders. Stakeholders have a direct impact on the overall quality, and it means that stakeholders and software system quality are directly proportional to each other. Stakeholders are considered as the primary source of information or needs which must be included in the functional aspects of a software project which is under process of development [10]. Currently different techniques are presented for identification of stakeholders. Most of these techniques are used to classify the stakeholders on the basis of domain knowledge, professional skills, experience, personal approach, communication, and control in the organization.

In this paper, the valuable systems are focused with respect to the implementation of valuable requirements. For the value oriented models, there are no specified procedures in UML (Unified Modeling Language) which provides a robust way to express the requirements in different angles [11]. The valuable software systems are directly associated with financial matters. The early implementation of the innovative idea in the market helps in gaining market leverage so the time to market is also very essential. Therefore, in order to avoid financial risks it is necessary to explore the stakeholders and their interests in a professional way because stakeholders and their interests are directly proportional to the success rate i.e.

System Success = Stakeholders + Requirements + Testing

The success rate of software projects is mainly dependent upon critical stakeholders, critical requirements and testing. Too much work is performed on requirements elicitation and analysis, but the stakeholder analysis process is still immature. Instead of all the efforts in RE the reports show a strange figure of success rate of software projects i.e. the success rate of projects in 2004 was 34%, in 2006 35% and in 2009 the success rate of software projects was 32%, again 32% in 2010, and the success rate in 2011 was 34% [12]. There are many modeling approaches for value based systems but all these approaches focus on functional and non-functional requirements of value based systems [13]. These approaches do not focus the importance of the identification and quantification of the stakeholders for value based systems.

In the case of valuable systems, the stakeholders are dispersed across a particular geographical area, so it is difficult to capture the requirements of all the stakeholders. Currently the process of stakeholder quantification is also not mature though the researchers have provided different techniques for stakeholders' analysis. Pacheco and Garcia are of view "there is still no Stakeholder Identification Process (SIP) framework or uniform description" [14] and same is true for identification and quantification of valuable stakeholders for value based systems. There are few major techniques regarding stakeholders' identification, and currently no proper technique exists for stakeholders' quantification.

The existing techniques do not focus the stakeholders of value based systems specifically. The existing stakeholders' analysis techniques focus the stakeholders based on their key aspects like influence, relationships and roles [15-18]. There are some techniques that do not consider these aspects of influence, relationships and roles for stakeholders' analysis [19, 20]. It is an evident that the stakeholders' analysis techniques lack in uniformity, and they are not cohesive. CMM, CMMi and ISO also do not provide any guideline that how to quantify the value of success critical stakeholders. Ballejos and Montagna have presented a technique based on roles or types. The end results of the technique are value neutral because various stakeholders are found with the same value profile [21-23]. The technique consumes too much time and not cost effective. PisoSIA®. is a technique for stakeholders' identification process in which Mitchells' model is integrated with PISO. The initial results obtained from the technique are not correct, so it is difficult to adopt the technique.

The findings about existing stakeholders' identification and quantification approaches state that the existing techniques are complex and provide a description of stakeholders at a higher level of abstraction. These techniques provide no process level detail in order to quantify stakeholders, are not uniform, cannot be adopted as a framework, are time consuming and costly. Some are only proposed frameworks and not applied in real scenarios [25]. In order to handle all these problems, some major changes are required in the stakeholder identification and quantification process. Currently in the software development life cycle of value based systems most of the technical aspects are handled in a value neutral way. There is not a single technique which may be considered as vital for identification and quantification of valuable stakeholders. Initially in this research some changes are proposed in SDLC and then in the proposed process some phases are described.

3. REQUIREMENTS ENGINEERING

With respect to quality, much of the focus is given on software requirements and design of value based systems. Requirements Engineering (RE) plays a vital role in the development of innovative or value based systems. The innovation brings certain complexities during design and development phases which are too difficult to handle and the development of such products is very hard to expose easily [14]. The fulfilment of requirements (functional and non-functional) is the essential or distinguishing characteristic of quality. In requirement engineering phase, the functions are analyzed and in the implementation phase the work is done on operations [26]. RE process is comprised of a set of stages called elicitation, analysis, specification, and validation [27, 28]. RE is a process which is difficult to handle, and it is not associated with the size of the industry directly or indirectly [29]. Requirements play a vital role in the success of a given software system [30].

RE is a complex decision making process so there is the need of the involvement of all the relevant stakeholders which are directly or indirectly affected by the software system. As RE is based on expert decisions, so there may arise the problems in terms of time or schedule, the cost incurred the functionality and overall performance of the system [31] and also due to the wrong selection of stakeholders. Different RE models are presented by the engineers that describe how the RE process should perform instead of describing its general working procedure [32]. Stakeholders have a direct link with the RE process so the involvement of success critical valuable stakeholders is very essential in the development of value based software systems. Wrong selection or involvement of stakeholders leads toward quality compromises or even failure. The issues of requirements are prioritization, implementation, testability, verifiability and acceptability that are associated directly or indirectly with the stakeholders.

4. SOFTWARE TESTING

The optimum quality results can only be achieved by applying testing techniques in a practical way. During testing there is the need to keep in view the stakeholders' profiles and their demands. To reveal the defects in the software applications, the industry professionals actually fight with the quality issues of software performance, effectiveness, robustness or reliability, security, usability and correctness in functionality [33-37]. In industry test cases are applied to find out erroneous functions of a software system. The purpose of testing is [38]:

- To improve quality,
- For Verification and Validation (V & V)

Testing "may be put into effect with the aim of improving quality, assessing reliability, checking and conforming correctness" [33]. Software Testing is also called a simulation based verification in which a software program is tested with the help of certain input vectors [39]. Even after extensive testing it is not sure that all the defects have been removed during QA phase. Different testing techniques are used to test the overall quality of the software, and different input combinations are applied to check requirements verification and validation. The two major testing methods are whitebox and blackbox, and other testing techniques are the sub-categories of these two major categories.

4.1 Black box Testing

It is also called as functional testing. In black box testing, the input data is given using the interface and results in the form of output are analyzed. The output data is compared with the critical requirements. Using this technique the sources to develop the test cases are software specifications [32, 40, 41]

4.2 White box Testing

It is a testing technique in which the testers test the complete code in detail, and they have to make sure that each and every statement is executed once [42].

4.3 Gray box Testing

It is a combination of black box, white box, assertion testing and mutation testing. "The Gray box Methodology combines Black box testing & White box testing methods with Proof of Program Correctness, Assertion and Mutation testing into an integrated testing methodology to verify and validate that developed software has properly implemented its requirements" [43]. The formula for Gray box Testing is shown as [43]:

Gray box Testing = (Blackbox Testing + Whitebox Testing) + (Assertion Testing + Mutation Testing)

4.4 Mutation Testing

It is a destructive sort of testing and "is used to test the adequacy of the software test cases and not the software itself" [43].

4.5 Assertion Testing

It is a technique which serves as a "proof of program correctness" [44]. In assertion testing software is validated using "predicates and verifications" [43]. Predicates serves as the preconditions for validation of a program while verification is performed on the basis of predicates in order to check the correct output [43].

5. ISSUES OF SOFTWARE TESTING TECHNIQUES

In the presence of different software testing techniques still the testing process is not effective. Different problems are faced during software testing. In testing phase, the testing problems are also mainly associated with stakeholders and vague requirements. Stakeholders have a direct effect on all stages of software development life cycle. Some of the issues of software testing are stated here.

5.1 Effectiveness of Testing

In software testing, the core challenge is "how to uncover the difficult-to-find software problem" [36]. Numbers of different testing issues force one to think how these testing techniques can impart the impact of effectiveness at optimum level. It is hard to quantify the fault detection effectiveness [36, 45] of these techniques. Sufficient data (defects detected by the testing techniques) is not available on the basis of which we can compare the effectiveness of these testing techniques.

For rigorous application of software testing techniques, there is the need of experienced and trained engineers. However, the involvement of human subjects [31] introduces the factors of high cost and time. The simulation of test sets is another way to apply these testing techniques effectively [46]. Simulation helps to find out the results that how many faults are detected by a given testing technique. In simulation, there are the chances of human error, which may be the cause of faulty statistical results.

5.2 Reliability and Robustness

"Software reliability is concerned with how well the software functions to meet customer requirements. It is defined as the probability that, the software works without failure for a specified period of time" [47]. In safety or mission critical systems the reliability or robustness factor is dependent upon sub factors like operating system, system configuration, controller structures and communication links [31, 36]. The malfunctioning of any of these components may result in erroneous software system and the ultimate shattering of the reliability. The degree of tolerance under stressful environmental conditions is referred to as a component's robustness or reliability [31]. For quality goals, it is essential that the test cases, used to reveal the defects, must be effective and assure the high system reliability quality factor.

5.3 Time/Cost

One reason that the defects cannot be removed thoroughly from the software application is the time complexity or the execution time of test cases [48, 49] and the budgetary constraints. The time to market has a severe impact on the overall quality testing of the system. For the solution of this issue, different algorithms are developed with different claims like Grover's Algorithm, Quantum Algorithm etc.

5.4 Critical Analysis

The purpose of critical analysis is to analyze the impact of software testing techniques on the overall quality of the software system. It is evident that all the defects cannot be removed using a single technique or even applying all the techniques. The application of software testing techniques varies from application to application so the right selection by the tester is very essential. Gardikiotis and Malevris stated "there currently exists a dearth of software assurance techniques to assess the robustness of both the application and the operating system under strenuous conditions" [31]. However, the application of software testing techniques demands exhaustive efforts for reliable results.

5.5 Domain Knowledge

Another major problem in testing is the domain knowledge. It is essential for the testing team and all stakeholders that they must possess the domain knowledge of application which is under test. The domain knowledge can be defined in terms of software requirements or the use cases. The requirements are tested to verify and validate the developed software product. Simply the clear understanding of right requirements is vital for a fully functional application and "getting the requirements right is the key to building successful and reliable software products" [47].

6. PROPOSED MODEL

The software system success rate, as stated above, is directly proportional to stakeholders, their requirements or interests and testing. The right set of help the testers to test the system as per software requirements specifications. Stakeholders are the initiators of a project, so the whole success revolves around them. In proposed model, the stakeholders are taken as the key to success and the major focus is on stakeholders. Stakeholders, requirements and testing make a triangle and in the middle of this triangle there lies the essential part called software quality in terms of user acceptance and can be termed as the NUCLEUS. The triangle is named as STAR (Stakeholder Testing Acceptance Requirements) Triangle. It is clear from the STAR triangle that failure of stakeholder identification process, failure of the requirements process and the failure of the testing process will result in failure of the system. So the success in terms of acceptance is directly proportional to the three main factors i.e. stakeholders, requirements and testing. The trade off in case of stakeholders can be very dangerous, so the stakeholders' value priority is very high. Figure 1 shows the interdependencies between the three.

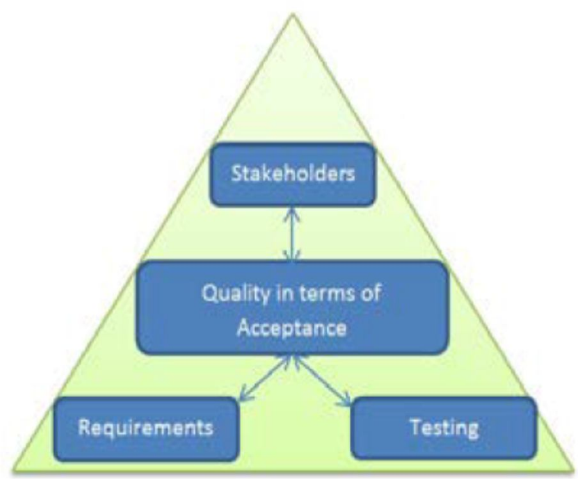

Figure 1. STAR Triangle

In the proposed model, the design and development phases of the software development life cycle are not discussed, and they are not part of this research. If the software development life cycle is taken into account, then the key phases of the SDLC are requirements analysis, system design, implementation, testing, release and maintenance. The proposed model, for value based systems, is also based on SDLC phases but with little modifications.

1. Business domain knowledge of requirements engineer
2. Value based stakeholders' identification process
   a) Domain based stakeholders
   b) Stakeholders domain or business understanding
   c) Stakeholders' experience
3. Value based stakeholders' quantification process based on metrics
4. Requirements Elicitation Process Standardization
5. Requirements Standardization
6. Requirements Prioritization process. Priority of the software requirements is defined based on financial benefit, requirements' implementation, requirements testability, cost incurred, market leverage and keeping in view that either the requirement is attracted or not. One may choose some existing techniques in order to prioritize SRS.
7. System Development Process Standardization
8. System Testing Standardization

7. PROPOSED PROCESS

The proposed process model requires a full elaborated research in all the eight key areas which are directly linked with main domains of stakeholders' analysis, requirements and testing. The proposed process consists of following stages.

- **a.** Framework for Stakeholders' Analysis
- **b.** Standardization of the Requirements
- **c.** Application Development
- **d.** Testing based on Standardized Requirements
- **e.** Software Quality Measurement

In proposed process, the first stage is the analysis of stakeholders. Different approaches are presented so far, and these approaches present a very high level picture of stakeholders' analysis. It is very difficult to adopt them as a framework. So in this phase of the proposed process a new framework will be proposed in order to analyze the stakeholders of a value based system. The proposed solution or framework for stakeholders' identification and quantification is based on following steps.

Perform rapid discussions and interviews with stakeholders

1. Write down the key responsibilities of the stakeholders.
2. Divide stakeholders into different categories based on the key responsibilities of the stakeholders.
3. Divide main categories of the stakeholders into sub-groups or categories.
4. Prioritize the stakeholders based on three main concepts namely stakeholders' key attributes, core functional needs and core non-functional needs.

Collect requirements from prioritized or valuable stakeholders in order to start the realization process.

The current framework for stakeholders' identification and quantification may be adopted as a standard. However, if it is not possible to adopt as a standard then at least it provides a way to analyze stakeholders thoroughly. This research paper focuses the first step of the stakeholders' analysis framework which is associated with the identification and categorization of the stakeholders into different groups. The remaining steps are out of the scope of this research and are not focused here.

8. STAKEHOLDERS' IDENTIFICATION AND QUANTIFICATION

This research paper is in continuation of the previous research by Baber et al., 2012 [50]. Stakeholders are directly associated with the software requirements analysis phase. The quality of the software application will be high if the requirements are correct and vice versa. The approach that is adopted here is a domain based approach. A variety of domains exists so the stakeholders and their needs vary in each domain. The most famous domains in the real life are education, medical, tourism, transportation, manufacturing, chemical, civil, electrical, mechatronics, mechanical, software, computer hardware, business, finance, management, hoteling, physics, mathematics and other applied sciences. Most of the software analysis techniques that are presented so far ignore the initial thorough analysis of the stakeholders. The existing techniques do not provide the way that how to initiate the research. In different approaches, different methods are adopted, and there is no uniform approach. Figure 2 describes the approach that is used to analyse stakeholders for a given value based software application. To start the stakeholders' identification process, the first step is to start the rapid sessions of discussions and conduct interviews of the stakeholders.

The main purpose of the discussions and the interviews is to find out the key responsibilities of the stakeholders. Stakeholders are divided into two main groups based on their key responsibilities namely *functional stakeholders* and *non-functional stakeholders*. The functional stakeholders are directly linked with the key process areas while the non-functional stakeholders are not involved in key process areas.

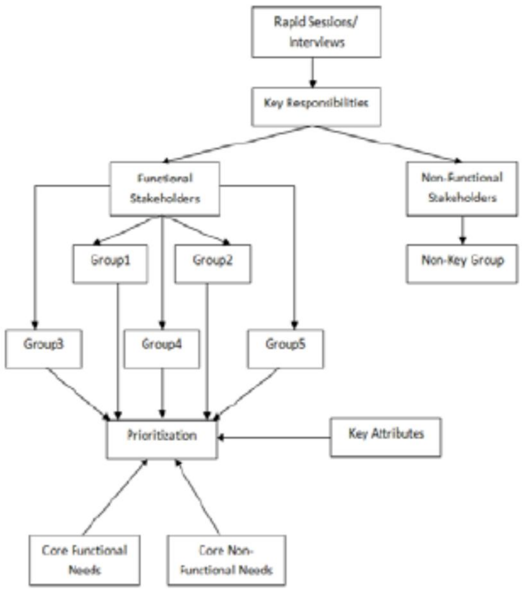

Figure 2. Stakeholders' Identification Model

The functional stakeholders' category is further divided into sub-groups. These sub groups are further categorized based on the similarities of core responsibilities between different stakeholders. The administrative team is placed in one group, the top management in another group and the store management group is a separate category. The functional stakeholders are prioritized based on their core functional needs, core non-functional needs and attributes. With the adoption of this approach, it becomes easy to evaluate the worth of a given stakeholder.

The core functional and non-functional needs are those requirements that add some value to the quality of the system. The key attributes are the attributes which depict the characteristics of a stakeholder in terms of technical and non-technical skills. These key attributes play a vital role in the whole stakeholders' identification and quantification process. The key attributes that are considered in this approach are *business process understanding* of the stakeholders, *experience* and *training*. The key stakeholder can be stated as:

Key Stakeholder=Core Functional Needs + Core Non-Functional Needs + Key Attributes

The next section describes the case studies in order to validate the said key stakeholders grouping approach.

9. CASE STUDIES

The two case studies are selected to validate the stakeholders' grouping approach for a given value based software application. The two case studies that are selected are Spare Parts Management System and College Management System. To initiate the research, 2 teams are selected. Each team is comprised of 4 software professionals, 1 is working as a requirement engineer, 2 as developers, and 1 as a tester and documentation writer. The main purpose of two teams is to generalize the results obtained from the case studies. The comparison of the results is performed to find out the worth of the proposed process.

9.1 Spare Parts Management System

The Spare Part Management System (SPMS) deals with the proper placement of a spare part in the store. However, the system plays a vital role in managing the inventory, financial records, and key vendor information. The current system that is adopted by the company is a manual system and is highly problematic in terms of paper based manual handling of the records. The SPMS will help to analyse the different economic aspects that how many parts are sold in a given time frame. Which parts are sold frequently and which parts are not sold in a bulk. Inventory checking helps to find out that which parts are not present in the store. It is easy to get the information about the parts that are prone to frequent failures and so on. The team visited the spare parts store of the Suzuki automotive company and listed the different stakeholders. Table 1 lists all possible functional stakeholders of SPMS that are reported by the requirements engineer. The non-functional stakeholders are not part of this list.

Table1: List of SPMS Stakeholders

Stakeholders	Number
Director Operations	1
Front Desk Manager	1
Front Desk Staff	3
Store Officer	1
Store Assistants	2
Admin Officer	1
Admin Assistant	2
Clerk	2
Accountant	1
Account Assistant	2
Total Stakeholders:	16

The team discussed different key processes of the business and noted the key responsibilities of the stakeholders based on key business processes. The total number of stakeholders in SPMS is 16. The reported stakeholders are director operations who handle all high level matters of the company including matters associated with financial streams. Front desk manager and staff care about customers dealing. Store officer and staff handle the store inventory. The administrative activities are performed by admin officer and admin staff. Accountant and account assistants handle all the accounts' activities of the SPMS.

9.1.1 Stakeholders' categorization

Based on the similar key responsibilities of the stakeholders the team has divided them into different categories namely Key Management, front desk management, administration, accounts and store. In figure 3, the Key Management category of the stakeholders represents the top management of SPMS and that category of the stakeholders mainly deals with the key business features. The front desk management deals with the front desk activities and provides services to customers. The administration group deals with the administrative affairs of SPMS. The accounts group manages all the activities related to the accounts of the organization. The stakeholders who are managing the store come in the domain of store group. The grouping makes easy to understand the business process, and this grouping is based on divide and conquer rule.

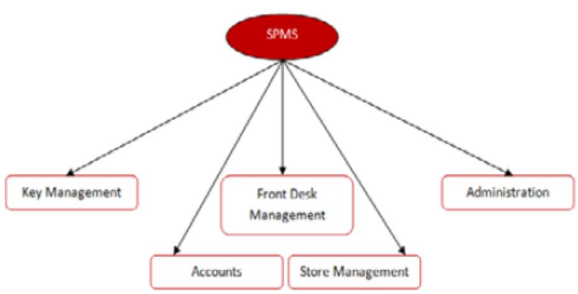

Figure 3. SPMS Stakeholder Categories

9.1.2 Stakeholders' prioritization

Stakeholders' prioritization is performed in order to select a right set of stakeholders for the given value based system. The prioritization of the stakeholders is performed based on expert judgment using the three key stakeholders' attributes namely *business process understanding, experience* and *training*. In this phase of the approach, the team has first chosen the experienced workers of the organization that have more experience in the similar capacity. The team then discussed the different business processes with each of the stakeholders and divided them into four classes based on their experience and skills. The four classes of the stakeholders are as under:

A. Extremely Important
B. Important
C. Considerable
D. Not Important

The category A of the stakeholders is extremely important category and all the stakeholders related to this class must be included in the requirements elicitation phase. Category B of the stakeholders is also important and must be included in requirements elicitation phase. The category B in this scenario is not an executive class however, an executive may come in other classes depending upon the prioritization criteria. Category C is the category that may be considered in requirements elicitation phase in case of any unclear or vague scenario. Table 2 describes the priority classes of the stakeholders of the SPMS as reported by the team. All executive members, of each group, are placed in category A.

Table 2: Prioritized List of SPMS Stakeholders

Stakeholder	Category
Director Operations	A
Front Desk Manager	A
Store Officer	A
Admin Officer	A
Accountant	A
Front Desk Staff (1)	B
Store Assistants (2)	B
Admin Assistant (1)	B
Account Assistant (1)	B

Out of 16 stakeholders the 10 are selected as success critical stakeholders for the SPMS. The gathered requirements from these stakeholders are given to the developers for realization.

9.2 College Management System

The second case study is about College Management System (CMS). CMS is a web based application and handles all academic and administrative activities in an educational institution. All day to day business activities of the college are managed using this system. The information related to students, faculty, staff, and all the related things can be saved and shared easily in CMS. The CMS provides the facility of online admission application to new students. The administrator can track the attendance of students and faculty and all other records like financial streams, total number of admission applications received and college inventory management facility. The team working on CMS reported a complete list of stakeholders and is given in Table 3.

Table 3: Prioritized List of CMS Stakeholders

Stakeholders	Number
Principal	1
Vice Principal	1
HODs	6
Assistant Professor	7
Lecturers	25
Admin Officer	1
Admin Assistants	3
Accounts Officers	1
Accounts Assistants	2
Admission Officer	1
Admission Office Assistants	2
Librarian	1
Library Assistants	2
Bursar	1
Bursar Assistants	3
Receptionist	2
Store Officer	1
Store Assistants	2
Clerks	5
Total Stakeholders:	68

The team has reported a total of 68 key stakeholders in CMS after understanding the key business activities of the college. The reported stakeholders are principal, vice principal, head of departments (HODs), assistant professors, lecturers, admin officer, admin assistants, accounts officers, accounts assistants, admission officer, admission office assistants, librarian, library assistants, bursar, bursar assistants, receptionist, store officer, store assistants and clerks.

9.2.1 Stakeholders' categorization

The team has discussed the responsibilities of the different stakeholders. To know the responsibilities of different stakeholders, team asked the heads to provide the list of responsibilities of each key stakeholder. This thing made easy to understand all the business processes of the college. Based on the key responsibilities and business processes the team has divided the stakeholders into 9 main categories. Figure 4 describes the stakeholders' categories of the CMS.

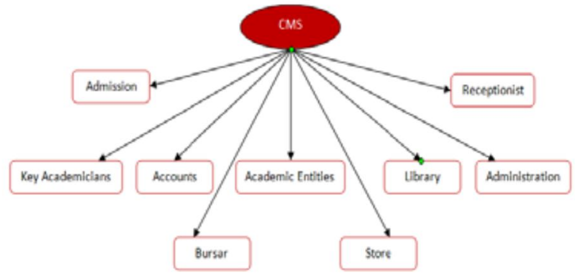

Figure 4. CMS Stakeholder Categories

The 9 key stakeholder categories of CMS are admission, receptionist, key academicians, accounts, academic entities, library, administration, bursar, and store. The key academician's category of the stakeholders deals with the overall performance and requirements of the college. The accounts category handles all activities related to accounts. The academic entities are the faculty members like assistant professors and lecturers, the library category, of stakeholders, deals with the library activities and administration category deals with all administrative affairs. The bursar category of stakeholders monitors all financial matters of the college while the store group deals with management of stationary and other items that are used in the college.

9.2.2 Stakeholders' prioritization

The different categories of the stakeholders are prioritized by the team using three key stakeholders' attributes namely *business process understanding, experience* and *training.* The brainstorming and expert judgment was applied by the team in order to resolve the plight of prioritization of such a large number of stakeholders. The team has divided the stakeholders into following priority levels.

A. Highly Expert
B. Experts
C. Required
D. Entry Level

Table 4: Prioritized List of SPMS Stakeholders

Stakeholders	Category
22	Highly Expert
8	Experts
4	Required
Total:	34

Out of the 68 stakeholders only 22 stakeholders are selected in the Highly Expert level of prioritization, 8 stakeholders in the Experts level, and 4 in the required level while remaining are not considered important for the CMS. A total, of 34 stakeholders, is given due importance by the team based on different levels of priorities defined by the team members.

10. DISCUSSION AND FUTURE WORK

The requirements gathered from prioritized stakeholders are realized into the systems by the programmers. It is found that the selected stakeholders have played a vital role in providing exact requirements for the systems. The developed systems are implemented and tested in the real environment and are appreciated by the larger community of the organizations.

In CMS, it was found that 2 of the functional aspects were found as un-compatible with the needs of the college. However, the stakeholders have stated them properly, but the team members were unable to perceive them in the real sense. The results of CMS team report that the process takes a little bit more time to evaluate the stakeholders based on the proposed key attributes. It is reported and suggested by both the teams that there is the need of some key metrics that should cover a wider range of the key attributes of the stakeholders. Both teams have prioritized the stakeholders based on different scales and this shows independence in process shaping. However, there is the need to provide a unique priority framework that may help to industry professionals in finding a unique prioritized set of stakeholders. The approach is good for the organizations having small and a large number of stakeholders. The future work focuses on to find out the key attributes of stakeholders. Based on these key attributes a systematic framework of stakeholders' identification and quantification for value based software systems will be proposed. Based on key attributes the key metrics will be derived that will help to find out economic worth of a stakeholder. The requirements obtained from these stakeholders will be highly beneficial for economic driven systems and will also help in effective and reliable testing.

11. CONCLUSION

This paper presents the three main aspects which are considered as vital to achieve high software quality i.e. critical stakeholders, requirements and software testing techniques. The quality of value based systems can only be appreciated if accepted by the wider community and stakeholders. The acceptance by a wider community means a higher economic output in terms of financial benefits. So far for value based systems the stakeholder analysis is not performed on an economic basis keeping in view the economic value of the stakeholder. The economic value of stakeholders is taken as the role played by the stakeholders in increasing the level of benefit for an organization. The success rate of the project is mainly associated with the stakeholders.

The proposed STAR Triangle has given high priority to stakeholders because stakeholders are the main vein of the system. Many of the requirements engineering problems and testing problems can be avoided by a careful selection of stakeholders. The existing models and approaches that focus stakeholders' analysis are not suitable and state of the art for value based systems. They lack in providing details at lower levels. However, the current proposed model focuses on to define the individual value of each and every aspect that can add economic value to a value based system. The existing models are taking all the quality aspects at par without any due discrimination of their values.

ACKNOWLEDGEMENTS

Special thanks to Universiti Teknologi Malaysia (UTM). Also, to our Embedded and Real-Time Software Engineering Laboratory (EReTSEL) members for their continuous support. Moreover, our special thanks to Tom Gilb for his kind support in this research.

REFERENCES

[1]. B. Boehm, "Value-based software engineering: Overview and agenda," *Value-Based Software Engineering*, pp. 3-14, 2005.

[2]. Bryant, "It's engineering Jim… but not as we know it: software engineering—solution to the software crisis, or part of the problem?," in *Proceedings of the 22nd international conference on Software engineering*, 2000, pp. 78-87.

[3]. Gordijn J., "Value-based Requirements Engineering, Exploring Innovative e-Commerce Ideas. ," SIKS Dissertation Series., 2002.

[4]. N. Zarvic , M. Daneva, and R. Wieringa, "Value-based requirements engineering for value webs," *Requirements Engineering: Foundation for Software Quality*, pp. 116-128, 2007.

[5]. Aurum and C. Wohlin, "A value-based approach in requirements engineering: Explaining some of the fundamental concepts," *Requirements Engineering: Foundation for Software Quality*, pp. 109-115, 2007.

[6]. S. A. Conger, *The New software engineering*: Course Technology Press, 1993.

[7]. H. Xie, L. Liu, and J. Yang, "i*-prefer: optimizing requirements elicitation process based on actor preferences," in *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009, pp. 347-354.

[8]. J. Offutt, "Software testing: from theory to practice," in *Computer Assurance, 1997. COMPASS'97.'Are We Making Progress Towards Computer Assurance?'. Proceedings of the 12th Annual Conference on*, 1997, pp. 48-51.

[9]. L. Briand, Y. Labiche, and Y. Wang, "Using simulation to empirically investigate test coverage criteria based on statechart," in *Proceedings of the 26th International Conference on Software Engineering*, 2004, pp. 86-95.

[10]. K. Bittner and I. Spence, "Establishing the Vision for Use Case Modeling, Use Case Modeling," ed: Addison Wesley Professional, Reading, 2003.

[11]. D. Graham, "Requirements and testing: Seven missing-link myths," *Software, IEEE*, vol. 19, pp. 15-17, 2002.

[12]. S. Group. (2012, July 2012). *The CHAOS Report*. Available: http://www.standishgroup.com/

[13]. M. J. Escalona and N. Koch, "Requirements engineering for web applications-a comparative study," *Journal of Web Engineering*, vol. 2, pp. 193-212, 2004.

[14]. C. Pacheco and I. Garcia, "Effectiveness of Stakeholder Identification Methods in Requirements Elicitation: Experimental Results Derived from a Methodical Review," 2009, pp. 939-942.

[15]. R. K. Mitchell, B. R. Agle, and D. J. Wood, "Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts," *Academy of management review*, pp. 853-886, 1997.

[16]. J. McManus, "A stakeholder perspective within software engineering projects," 2004, pp. 880-884 Vol. 2.

[17]. O. Preiss and A. Wegmann, "Stakeholder discovery and classification based on systems science principles," 2001, pp. 194-198.

[18]. S. M. Young, S. McDonald, H. M. Edwards, and J. B. Thompson, "Quality and people in the development of situationally specific methods," 2001, pp. 199-203.

[19]. C. Elayne and E. Tony, "The role of the stakeholder in managing change," *Communications of AIS*, vol. 2, 1999.

[20]. H. Sharp, A. Finkelstein, and G. Galal, "Stakeholder identification in the requirements engineering process," 1999, pp. 387-391.

[21]. L. C. Ballejos and J. M. Montagna, "Method for stakeholder identification in interorganizational environments," *Requirements engineering*, vol. 13, pp. 281-297, 2008a.

[22]. L. C. Ballejos, S. M. Gonnet, and J. M. Montagna, "A Stakeholder Model for Interorganizational Information Systems," in *Requirements Engineering: Foundation for Software Quality*. vol. 5025/2008, ed Berlin / Heidelberg: Springer, 2008b, pp. 73-87.

[23]. L. C. Ballejos and J. M. Montagna, "Stakeholders Selection for Interorganizational Systems: A Systematic Approach," *The Past and Future of Information Systems: 1976–2006 and Beyond*, pp. 39-50, 2006.

[24]. J. Davison, J. B. Thompson, D. A. Deeks, and M. Lejk, "PisoSIA® a stakeholder approach to assist change in information systems development projects and aid process improvement," *Software Quality Journal*, vol. 14, pp. 25-36, 2006.

[25]. R. Razali and F. Anwar, "Selecting the Right Stakeholders for Requirements Elicitation: A Systematic Approach," *Journal of Theoretical and Applied Information Technology*, vol. 33, 2011.

[26]. M. R. Lyu, *Handbook of software reliability engineering* vol. 3: IEEE Computer Society Press CA, 1996.

[27]. P. Roger, "Software Engineering a Practitioner's Approach," *McGrow-Hill International Edition*, 2005.

[28]. H. Kaindl, S. Brinkkemper, J. A. Bubenko Jr, B. Farbey, S. J. Greenspan, C. L. Heitmeyer*, *et al.*, "Requirements engineering and technology transfer: obstacles, incentives and improvement agenda," *Requirements engineering*, vol. 7, pp. 113-123, 2002.

[29]. M. I. Babar and S. A. K. Ghayyur, "A Review of Value Based Requirements Engineering for Value Webs," *International Journal of Reviews in Computing*, vol. 1, 2009.

[30]. F. P. Brooks, *The mythical man-month: essays on software engineering*: Addison-Wesley, 1995.

[31]. S. Gardikiotis and N. Malevris, "Program Analysis and Testing of Database Applications," in *Computer and Information*

Science, 2006 and 2006 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. ICIS-COMSAR 2006. 5th IEEE/ACIS International Conference on, 2006, pp. 368-373.

[32]. L. A. Macaulay, *Requirements engineering*: Springer-Verlag, 1996.

[33]. O. Iskrenovic-Momcilovic and A. Micic, "Mechatronic Software Testing," in *Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2007. TELSIKS 2007. 8th International Conference on*, 2007, pp. 486-489.

[34]. L. Bening and H. D. Foster, *Principles of verifiable RTL design: a functional coding style supporting verification processes in Verilog*: Springer, 2001.

[35]. P. Mathur, "Performance, effectiveness, and reliability issues in software testing," in *Computer Software and Applications Conference, 1991. COMPSAC'91., Proceedings of the Fifteenth Annual International*, 1991, pp. 604-605.

[36]. G. Antoniol, L. C. Briand, M. Di Penta, and Y. Labiche, "A case study using the round-trip strategy for state-based class testing," in *Software Reliability Engineering, 2002. ISSRE 2003. Proceedings. 13th International Symposium on*, 2002, pp. 269-279.

[37]. S. Shelar and S. Sawarkar, "Database Instances Generation Tool for White-Box Testing," in *Signal Acquisition and Processing, 2009. ICSAP 2009. International Conference on*, 2009, pp. 112-116.

[38]. J. Bach, C. Kaner, and B. Pettichord, "Lessons learned in software testing: a context-driven approach," ed: John Wiley & Sons. Yhdysvallat, 2002.

[39]. R. Zhao, M. R. Lyu, and Y. Min, "A new software testing approach based on domain analysis of specifications and programs," in *Software Reliability Engineering, 2003. ISSRE 2003. 14th International Symposium on*, 2003, pp. 60-70.

[40]. B. Beizer, "Software testing techniques," *New York, ISBN: 0-442-20672-0*, 1990.

[41]. T. Tsuchiya and T. Kikuno, "On fault classes and error detection capability of specification-based testing," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, pp. 58-62, 2002.

[42]. [42] L. Wang, "Issues on software testing for safety-critical real-time automation systems," in *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, 2004.

[43]. G. M. Kapfhammer and M. L. Soffa, "A family of test adequacy criteria for database-driven applications," in *ACM SIGSOFT Software Engineering Notes*, 2003, pp. 98-107.

[44]. M. M. Baig and A. A. Khan, "Plummeting the Software Testing Time Complexity," in *Software Engineering, 2009. WCSE'09. WRI World Congress on*, 2009, pp. 166-169.

[45]. M. Schmid, A. Ghosh, and F. Hill, "Techniques for evaluating the robustness of Windows NT software," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, 2000, pp. 347-360.

[46]. R. Bache and M. Mullerburg, "Measures of testability as a basis for quality assurance," *Software Engineering Journal*, vol. 5, pp. 86-92, 1990.

[47]. R. Zhao, M. R. Lyu, and Y. Min, "Domain testing based on character string predicate [software testing]," in *Test Symposium, 2003. ATS 2003. 12th Asian*, 2003, pp. 96-101.

[48]. L. Briand and Y. Labiche, "Empirical studies of software testing techniques: Challenges, practical strategies, and future research," *ACM SIGSOFT Software Engineering Notes*, vol. 29, pp. 1-3, 2004.

[49]. S. S and S. SD, "Database Instances Generation Tool for White-Box Testing," in *Signal Acquisition and Processing, 2009. ICSAP 2009. International Conference on*, 2009, pp. 112-116.

[50]. M. I. Babar, M. Ghazali, and D. N. A. Jawawi, "Considerations in Software Quality Enhancement for Value Based Systems," presented at the Malaysian Software Engineering Conference (MySec), 2012