Users' Perspective of Software Quality

Anas Bassam AL-Badareen,
Mohd Hasan Selamat, Marzanah A. Jabar, Jamilah Din, Sherzod Turaev
Anas_badareen@hotmail.com ,
{hasan, marzanah, jamilah, sherzod}@fsktm.upm.edu.my

Faculty of Computer Science and Information Technology
University Putra Malaysia
43400 UPM Serdang
Selangor Malaysia

Abstract: - In last decade, software engineering research increasingly focused on software quality enhancement and evaluation, whereas most of these researches concentrate on the internal/ development perspective. However, users mainly care in the quality of performs intended functions efficiently without knowing how the software product was developed, how it is work from inside, or it is internal quality. The success of software companies is completely depends on the users' satisfaction, which they decide to use a software product or not. Therefore, in software development, strong attention must be given to the user's satisfaction. This study aims to present the quality of the software products from user's perspective. From user's viewpoint, the characteristics of the software product are discussed. Finally, a user's perspective quality model is proposed.

Key-Words: - Software Quality, Quality Model, Software Evaluation, User Perspective, Functionality, Reliability, Performance, Usability, Portability.

1 Introduction

In last decade, the cost of software products is lower, that caused grow in the software market contention and software products are being used by individuals in addition to the corporations. Therefore, research in software engineering increasingly grew and focused on software quality evaluation and enhancement, whereas most of these researches concentrate on the internal/ development perspective [1].

Since, the software market interest on the user's satisfaction, more attention to perspective of users in software quality is required. Software users from different education background and culture are considered in developing software products. Hence, without considering these factors, the software will be less used [2], which means the software product failed in the market.

According to ISO9126, the main consideration of the users is the software usability, performance, and its effects without knowing what inside it, how it is work, or how it was developed.

Since, the software users does not care about all of software characteristics that are required to identify the quality of the software product, it is seems to be inaccurate to show them the quality that they are looking for. Therefore, the quality of the software as users need is very important in the market.

2 Software Quality Models

Since 1978, when McCall proposed first software quality model, several models were proposed to evaluate the characteristics of the software products. These models combined the different points of views: Manager, Developer, and user. Therefore, there is no a clear picture of the software quality shows to the intended users. For example, if such software product has a high maintainability and low usability might be equal to software has a high usability low maintainability.

3 User's Emotion Quality Models

In the other side, the affect of the software products on the user were considered and several emotions models were proposed. The aim of these models is to evaluate the software product from what the users feels when they use it. Éthier [3] proposed a B2C model calculates the emotions of the end users instead of software characteristics. The cognitive emotions were adopted in the model. Chen-Ya [4] examine the influence of user motivations and emotion factors on user behavior in the Web 2.0 environment.

4 User's Perspective Quality Factors

Whereas, different characteristics of software product were considered and measured, a number of these characteristics are considered by the end user's. In the following, the list of software characteristics, that considered by the end users and affect their emotions.

4.1 Functionality

The main idea of any software product is to perform specific business function. Therefore, the functionality of the software product considered as crucial factor in the software quality, which identify whether the software is usable or useless, regardless the values of other software quality factors. The functionality of the software presents whether the software product is suitable, the result is accurate, and whether certain standard is followed in order to perform intended functions. Figure 1, shows the characteristics of the software functionality.

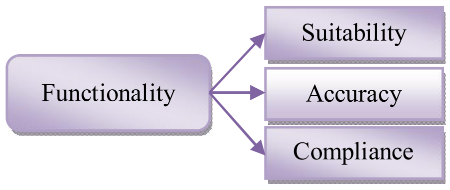

Figure 1: Software functionality

The **suitability** of the software presents how the system fit the developer's requirements [5]. Therefore, the suitability evaluates the ability of the software product to produce desired result and appropriate for a specified environment [6].

Software **accuracy** is defined as the ability of the software products to achieve its requirements [7], by producing accurate result as required by the system developer [5, 8]. Software accuracy affects the system safety, process continuity, maintainability, and totally the cost of the system [9].

The **compliance** presents whether the system has followed any standard or certificates to achieve the user requirements.

4.2 Reliability

The reliability of the software represents the ability to perform the intended function properly without any failures [7]. That is maintaining a level of services under specific condition within specific period of time during system operation [8, 10-11]. Therefore, the reliability measures the failures occurred in the software product within defined period of time [5, 8].

Moreover, the reliability considered the information and the system function safety harms, that may be caused by unauthorized people. Hence, the reliability of the software product consists of several characteristics: integrity, fault recovery, and maturity. Figure 2, shows the characteristics of the software reliability.

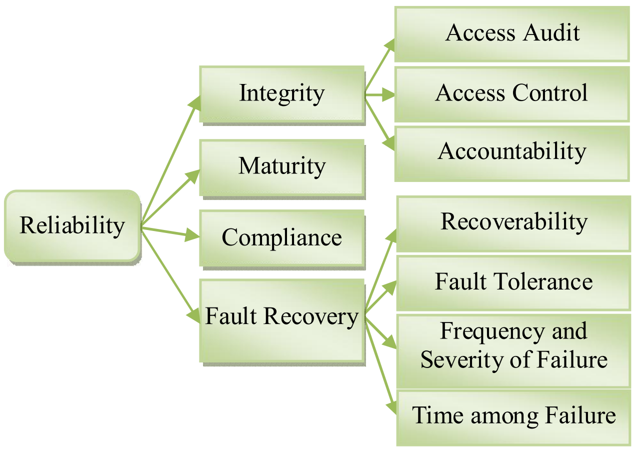

Figure 2: Software Reliability Characteristics

4.2.1 Software Integrity

High attention must be given to the system security [12], whereas, data and information are key factors for any establishment. Software products provide a support to recognize a people who might use a system, which identify the system users into authorized and unauthorized users [10, 13].

The software security is deal with the privileges that are given to the system users, in order to access specific functions such as (view, add, update, or delete data) [11]. Generally, it can be defined as the ability of the system to defend itself against unauthorized use [12] and to protect it components (data, information and functions) from [8].

Whereas software integrity intends to protect the system from any harm, the errors and faults need to be considered. Hart [14] defined the software integrity as a probability that a given system will operate within its specified limits without the occurrence of a software incident, which is directly related to the number of errors remaining in the software.

Moreover, the quantitative and qualitative risk analysis are used to reduce the level of threats risk to acceptable level. They also mention about the security preserving that can apply during software development and after produce the software.

In terms of user's access, two layers of access are defined, access audit and access control. **The access audit** allows the authorized users to enter and use a system. This level of access defined the user to authorized and unauthorized. **Access control** allows specific authorized users to access specific function within the system. At this level, the authorized users classified into several groups according to the privileges that are given to them.

Hence, the levels of security doesn't covered the responsibility of each action occurred within the system, which represents the level of the software security. The software accountability aimed to ensure that every action occurred in the system has been traced back to some entity [15].

4.2.2 Fault Recovery

Indeed, it is very difficult to expect all of failure that may occur during system operation. Hence, the ability of the system to perform it is functions during failure is a crucial during software using. The failure characteristics and types are calculated, the frequency and severity of failures and the time among failures. Fenton [16] presents the failure types and the reports of the failures collection. Moreover, the behavior of the system and the affection of the failures on the system are considered.

The **fault tolerance** is used to understand the context of the system which describes whether the system is able to maintain specific level of performance during faults that may occur during system using [5, 8]. At the same time, the system elegance during failure recovery is considered. The software **recoverability** represents the ability of the system to reestablish its level of performance and restore the data, which are directly affected from an unexpected failure [5, 8].

4.2.3 Maturity

Practically, the history of the software and the certificates are like stamp that verify the software features and characteristics. Software history describes the maturity story of the product [10], how much work has been done using this technology or number of versions released of the technology [8].

4.2.4 Compliance

The compliance presents whether the software has followed any standard or certificate in order to achieve certain level of reliability.

4.3 Performance

Software performance is a most affected software characteristic, which is affected by everything in the system product, from a software characteristics to the system environment such as operating system, middleware, hardware, and communication networks [17]. System performance is a make-or-break quality for software [18], which is an important nonfunctional attribute of software systems for producing quality software [19-21], that consider the run time property [22].

System performance is characterized by the amount of useful work accomplished by a system compared to the time and resources used. The performance factor is destined to evaluate whether the software application running efficiently on the computing resources available.

The performance factor represents the degree of the system efficiency to produce desired result during system operation. This degree is represented by combination of software and hardware attributes which influence on the time of answer and the range of the software services coverage. Figure 3, shows the characteristics of the performance factor.

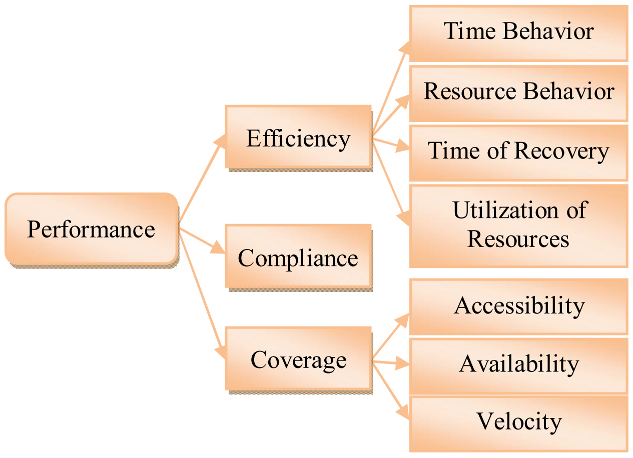

Figure 3: Software Performance Characteristics

4.3.1 Software coverage

In terms of ***coverage***, software performance concerned about the availability, accessibility, and the velocity. The **availability** of the software is the proportion of time a system is in a functioning condition [23]. This time up of the system represents the duration time of the system responding. The **velocity** of the system is the average rate of successful massages that are delivered through communication channels.

Software **accessibility** presents the degree to which a product, device, service, or environment is accessible by as many people as possible. This factor concerned about the area covered by the system. Beside the coverage area, the average of the success services that offered on this area is covered.

4.3.2 Software efficiency

In terms of **speed**, the performance factor represents the ***efficiency*** to perform software functions and the **time to recover** a system from failures. The **efficiency** is the ability of the system to use its hardware and software resources to perform its functions in order to achieve required requirements [7, 13].

Thus, this factor concerned about the amount of resources used [5] under specific conditions [10] during required functions performing [8]. Therefore, it is dealing with processor speed and storages capacity.

The **time behavior** represents the ability of the system to perform specific function under stated condition within appropriate response time and throughput rate.

Resource behavior is the capacity of the storages that used under specific condition in order to perform specific task. Besides, the efficient use of the resources is considered. The **utilization of resources** represents the ratio of the available resources used by the software application.

The **time of recovery** represents the time required by a system to reestablish it is level of performance and recover it is data that affected from unexpected failures [23].

4.3.3 Compliance

The **compliance** presents whether the system has followed any international standard or certificate in order to achieve a level of performance.

4.4 Usability

According to ACM the usability engineering (called human-computer interaction engineering) is defined as “a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and the study of major phenomena surrounding them”.

The characteristics of usable software as shows in figure 4, were discussed in [24]. According to the software life cycle phases, the usability characteristics were classified into three main categories: Interface characteristics, training, and operation supportability.

4.4.1 Interface Characteristics

The interface factor considered the user interface characteristics, which are aesthetic, consistency of the user interface, and communicativeness. The interface aesthetic presents the beautifulness of the interface and how much it is liked by the end users. The consistency of the user interface presents whether the user interface has any contradictions in term of words, situation, or actions. The interface communicativeness presents how software well communicates with the end user.

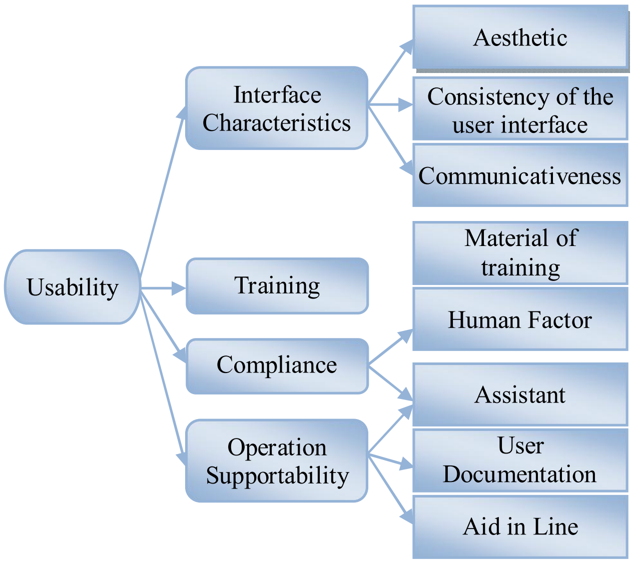

Figure 4: Usability Factor

4.4.2 Training

The training factor considered the material and the process of produce a motivated user who has basic skills to operate the system. This factor consists of material of training and human factor. The material of training is the documents are used to train the end users.

The material of training factor presents the quality of the materials that used to teach the end users the basic skills of how to use a system. This factor consists of completeness, clarity, consistency, and suitability.

4.4.3 Operation Supportability

Operation supportability is the facilities that are used to support the end users during using the system.

The user assistance is a general term for guided assistance to software product users. Assistance can automatically perform procedures or step users through the procedure, depending on the question that the user asked.

Online help (aid in line) is a form of users' assistance. That is designed to give assistance in the use of a software application or operating system by present information on a broad range of subjects through computer software. The online help version of the installation instructions meets the users' usability requirements by allowing users to access information directly from the interface itself [25].

User document is an electronic or printed body of material that provides information to users of software. The user documentation and the help system should be complete; the help should be context sensitive and explain how to achieve intended tasks [5]. Therefore, the document should be clear, complete, consistent, and suitable.

4.4.4 Compliance

The usability compliance shows whether the software product has followed any international standard or certificate in order to achieve the level of usability.

4.5 Transferability/Portability

Software **transferability** expresses the ability of the software to work properly in different type of platforms [7]. It is deal with effort required to transfer a program from one hardware configuration and/or software system environment to another [11, 13] with little modification [5]. This characteristic refers to how the software can be adopted to changes its environment or with its requirements [10]. Figure 5, shows the characteristics of the portable software.

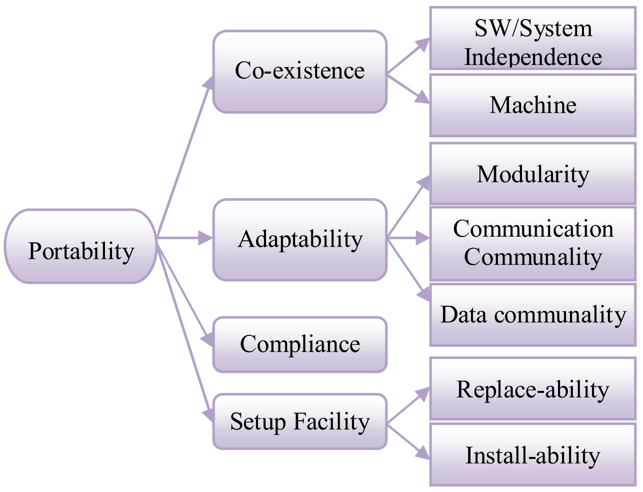

Figure 5: Portability Factor

4.5.1 Coexistence

Coexistence (integrated) is a state in which two or more systems are working together while respecting their differences and resolving their conflicts nonviolently. **System integration** is combination of several functions of several productivity software programs into one application. In order to evaluate whether the system able to integrate with others, two characteristics have to be considered, software system independence and machine independence. **Software system independence** is represent the degree to which program is independent of nonstandard programming language features, operating system characteristics and other environment constraints. **Machine independence** (hardware independence) is the degree to which the software is de-coupled from its operating hardware.

4.5.2 Adaptability

Software **adaptability/interoperability/interface facility** is the ability of the system to provide the users with tools to make them able to change the system characteristics [26]. In order to be able to adapt the system to different specified platforms [5]. It is evaluated by the degree of ease with which the system is adapted to new environments. Hence, adaptability factor concerned about software modularity, communication communality, and data communality.

Decomposable (**modular**) system which is combines several independent manageable parts. These parts are developed to be communicative with other parts and to be independent from any out affect. **Communication commonality** is represents the degree to which standard interfaces, protocols and bandwidth are used. **Data commonality** is explicit the use of standard data structures and types throughout the program.

4.5.3 Setup Facility

The setup ability of the software is concerned, which is the ability to install the software and to replace the previous versions. Software **install-ability** is the ability to install the software product easily in different platform [5, 8, 26]. **Replace-ability** represents the ability of the software to replace others specified in the environment of that software [8]. It evaluate whether the software compatible with its previous versions, which means that the software product can easily replace the previous version without any major efforts [5].

4.5.4 Compliance

The **compliance** of the portability factor is also considered. It represent whether the system has followed any standard or international certificates, in order to verify the portability characteristic.

5. Conclusion

User's satisfaction is the essential key of success any product in the market. Recently, software quality research concentrates on user satisfaction and emotion during using software products. This helps software developers to produce software products likely and accepted by end users. In this study, the characteristics of software products that affect the user satisfaction and emotions were discussed. Furthermore, the model of software quality from user's perspective is proposed. For future, the relationships between software product characteristics and user's emotions have to be defined.

References:

[1] B. Boehm*, et al.*, "Fifth Workshop on Software Quality," in *Software Engineering - Companion, 2007. ICSE 2007 Companion. 29th International Conference on*, 2007, pp. 131-132.

[2] R. Holcomb and A. L. Tharp, "An amalgamated model of software usability," in *Computer Software and Applications Conference, 1989. COMPSAC 89., Proceedings of the 13th Annual International*, 1989, pp. 559-566.

[3] J. Éthier*, et al.*, "B2C web site quality and emotions during online shopping episodes: An empirical study," *Information & Management,* vol. 43, pp. 627-639, 2006.

[4] W. Chen-Ya*, et al.*, "Emotion and Motivation: Understanding User Behavior of Web 2.0 Application," in *Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on*, 2009, pp. 1341-1346.

[5] A. Sharma*, et al.*, "Estimation of quality for software components: an empirical approach," *SIGSOFT Softw. Eng. Notes,* vol. 33, pp. 1-10, 2008.

[6] T. Kroeger and N. Davidson, "A Perspective-Based Model of Quality for Software Engineering Processes," in *Software Engineering Conference, 2009. ASWEC '09. Australian*, 2009, pp. 152-161.

[7] J. A. McCall*, et al.*, "Factors in Software Quality," *Griffiths Air Force Base, N.Y. Rome Air Development Center Air Force Systems Command,* 1977.

[8] A. Kumar*, et al.*, "A quantitative evaluation of aspect-oriented software quality model (AOSQUAMO)," *SIGSOFT Softw. Eng. Notes,* vol. 34, pp. 1-9, 2009.

[9] C. E. Davis, "Evaluating computer programs for analyzing industrial power systems: what users need to know," in *Petroleum and Chemical Industry Conference, 1990. Record of Conference Papers., Industry Applications Society 37th Annual*, 1990, pp. 77-85.

[10] M. Torchiano*, et al.*, "COTS products characterization," presented at the Proceedings of the 14th international conference on Software engineering and knowledge engineering, Ischia, Italy, 2002.

[11] F. Haiguang, "Modeling and Analysis for Educational Software Quality Hierarchy Triangle," in *Web-based Learning, 2008. ICWL 2008. Seventh International Conference on*, 2008, pp. 14-18.

[12] S. Khaddaj and G. Horgan, "A Proposed Adaptable Quality Model for Software Quality Assurance," *Journal of Computer Sciences,* vol. 1, pp. 482-487, 2005.

[13] J. J. E. Gaffney, "Metrics in software quality assurance," presented at the Proceedings of the ACM '81 conference, 1981.

[14] G. Hart, "The software integrity of a computer system installed in a royal naval frigate," *Microelectronics Reliability,* vol. 22, pp. 1061-1066, 1982.

[15] E. Bertino*, et al.*, "End-to-end accountability in grid computing systems for coalition information sharing," presented at the Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead, Oak Ridge, Tennessee, 2008.

[16] N. E. Fenton and S. L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*: PWS Publishing Co., 1998.

[17] M. Woodside*, et al.*, "The Future of Software Performance Engineering," in *Future of Software Engineering, 2007. FOSE '07*, 2007, pp. 171-187.

[18] K. S. Jasmine and R. Vasantha, "Identification Of Software Performance Bottleneck Components In Reuse based Software Products With The Application Of Acquaintanceship Graphs," in *Software Engineering Advances, 2007. ICSEA 2007. International Conference on*, 2007, pp. 34-34.

[19] D. E. Geetha*, et al.*, "Predicting performance of software systems during feasibility study of software project management," in *Information, Communications & Signal Processing, 2007 6th International Conference on*, 2007, pp. 1-5.

[20] C. E. de Barros Paes and C. M. Hirata, "RUP Extension For the Software Performance," in *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, 2008, pp. 732-738.

[21] C. Del Rosso, "The process of and the lessons learned from performance tuning of a product family software architecture for mobile phones," in *Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings. Eighth European Conference on*, 2004, pp. 270-275.

[22] K. Tokuno and S. Yamada, "Dynamic Performance Analysis for Software System Considering Real-Time Property in Case of NHPP Task Arrival," in *Secure System Integration and Reliability Improvement, 2008. SSIRI '08. Second International Conference on*, 2008, pp. 73-80.

[23] P. Eeles. (2005, 3/13/2010). *Capturing Architectural Requirements*. Available: http://www.ibm.com/developerworks/rational/library/4706.html

[24] A. AL-Badareen, B.*, et al.*, "Software Usability Factor within Software Life Cycle," in *The 2010 International Conference on Intelligent Network and Computing*, Kuala Lumpur, Malaysia, 2010.

[25] D. Yeats, "Revising documentation deliverables based on usability evaluation findings: a case study," presented at the Proceedings of the 22nd annual international conference on Design of communication: The engineering of quality documentation, Memphis, Tennessee, USA, 2004.

[26] N. J. C. Primus, "A generic framework for evaluating Adaptive Educational Hypermedia authoring systems," MSc, Business Information Systems University of Twente, Enschede, 2005.