Identifying Potential Usability Risk During Software Development Process

T. Jayaletchumi[a], Suhaimi Ibrahim[a,]*, Mohd Naz'ri Mahrin[a,]*

[a]*Advanced Informatics School (AIS) , Universiti Teknologi Malaysia (UTM) , Jalan Semarak, 54100 Kuala Lumpur.*

Abstract

Usability is an important factor in ensuring successful and usable software product development from a user's perspective. Ignorance and unawareness about the concept of usability and failure to address usability during software development process will affect user system acceptance which could lead to failure of software products. Therefore, usability can be considered as a risk factor in achieving usable software product. Unfortunately, there is still lack of proper definition, attributes and management of usability risk during software development process. This paper presents comprehensive study on the concept of usability risks, risk management and risk identification techniques. The objective of this paper is to provide potential usability risks that should be managed during software development process

Keywords: USABILITY, SOFTWARE PRODUCT, USABILITY RISKS, RISK MANAGEMENT, RISK IDENTIFICATION.

* Tel.: 603-26154452; fax: +603-26930933.
E-mail address: suhaimiibrahim@utm.my, mdnazrim@utm.my

1. Introduction

Software quality is an important aspect of any software development process. Quality of software depends on the process followed during its development [1] and any improvement of quality after the completion of software is unadvisable as it increases the cost and is almost remaking the product [2]. Usability has been recognized as an important quality factor of a software system and has always been present even in the very first models of software quality known as *FCM* proposed by McCall in 1977 (also known as *McCall's model*) [3]. Usability is defined by ISO as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use".

In Malaysia, the industry's usability awareness is relatively low with most are unaware of the existence of ISO 9241: Part 11 usability standard [4]. This low awareness is also reflected in the governmental organizations whereby even the term usability is not used among the in-house product designers and developers [5] but believed that usability aspects may be similar to what is known in MAMPU, as Proof of Concept (POC). POC consists of User Requirements Gathering, Functional Testing, User Acceptance Test (UAT) and Final Acceptance Test (FAT). However, the level of awareness in the industry is still higher compared to that of the government sectors and government linked companies. Low usability awareness will create more usability problems in a system or a website [6]. The existence of usability problem can be concluded as lack of quality in a software system or software system tends to be less usable and fails to fulfill the expectations of its users.

In the context of Malaysian government, Malaysian Administrative Modernization and Management Planning Unit (MAMPU) as the responsible government agency to oversee the functions of administrative modernization and human resource planning for the public sector, has carried out a Baseline Survey on E-Counter Services and Paperless Government on April 2011 [7]. The objectives of this survey are to gather basic information regarding total services and mode of the transaction (online or manual) offered by Malaysian government and to expand initiatives towards *Paperless Government.* Findings from this survey show that 35% of government services are provided through online systems. The most crucial finding shows that the level of usage of these online services is only 40% of the overall online services provided by government.

In 2011, a holistic evaluation model was presented which seamlessly integrates usage analysis and usability in the assessment of the communication quality of a web application using the User Experience Risk Assessment Model [8]. Based on this work,

usage analysis from baseline survey of E-Counter Services and Paperless Government indicated that the probabilities for users to be actually exposed to usability problems are higher. This is also supported by a study which has revealed several issues on usability and accessibility problems of Malaysia e-government website such as high number of broken links and slower accessibility speed [9]. Other studies also revealed that major usability problems includes poor impact of usability activities in product designs; limited skills and knowledge on usability among the designers and management; unawareness on various activities of usability engineering life-cycle; inappropriately used usability methods and even political games around usability [10].

With usability problems in most of the systems or website that we operate, it is crucial to reduce usability problems. Usability professionals have found that to successfully build usable software product, usability needs to be closely integrated with software development process [11]. There are many efforts to define and integrate a formal usability process into Software Development Lifecycle (SDLC) to improve the interaction and quality of the systems [12][13][14][15][16].

However, integration of usability practices into SDLC had faced many barriers in its implementation. It was reported that it is difficult to introduce a new practices into an SDLC [17]. It is also found that usability practices are not part of requirement engineering [16], so developers are often given an incomplete, confusing, and sometimes contradictory requirement.

Approach Centered on Usability and Driven by use cases (ACUDUC) process integrates usability in the use-case driven SDLC, only in requirement phase and not practiced in any organization [18] and research has also shown that Usability engineering is not being fully integrated in all phases of SDLC and is mostly limited to requirement and design phase. In fact their practical implementation is largely missing [19]. As a result, many development teams are facing an increasing challenge in avoiding and minimizing usability problems.

Usability problem can be considered as significant risk factor in producing usable software product. A common understanding in a very early phase about challenges and risks for product usability is required [20]. If usability risks are managed well, the overall chances of reducing risk of failure and producing usable software product could be increased. To our knowledge, study on usability risk and its impact to SDLC are still lacking. Not many approaches have been identified in the literature that defines potential usability risk of software product. Therefore, there are needs to identify

potential usability risk that development team should be aware and tackle during software development process.

This paper is structured as follows. Section II reviews existing studies on usability risk, risk management and risk identification approaches. A proposed methodology is illustrated in Section III. Section IV explains on findings and Section V includes conclusion and future work. This paper concludes with references.

2. Literature Review

Software products face many risks in their development lifecycles. Risks in software products can be defined in general as the probability of suffering loss while pursuing goals due to unpredictable factors [21]. Software risk management has become a crucial step to effectively manage software risks during its development lifecycle [22].

A. Usability

For the past years, usability has been defined with many meaning and definition. For example, ISO/IEC 9126-1 (2000) has defined usability as "The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions", ISO 9241-11 (1998) defined it as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" and IEEE Std. 610.12 (1990) defined usability as "The ease with which a user can learn to operate, prepares inputs for, and interpret outputs of a system or component" [45]. Generally, usability has been proposed as combination of different attributes and sub attributes [42].

B. Risk and Problem

Conceptually risk and problem are different. A risk is an uncertain future event that could have a negative effect (threat) or a positive effect (opportunity) on the project objectives. But a problem statement describes a 100% certain condition that exists now and threatens achieving the project objectives. Understanding the difference between a risk (threat) and a problem is important because they are treated differently in planning and execution stages.

The concept of risk has been widely used in many areas such as business and management, finance, insurance, security, economy, health, safety, environment and many more. In the context of software development, the term 'risk' has been used with various definitions. The earliest definition was given by McFarlan [51] and Boehm [24].

Many definitions of risk in the context of software development has been given in earlier research but many definitions looks similar but handles different aspects and effects of risk (including negative or positive effects). However, this study analyzes risk as **negative consequence and not as an opportunity.**

C. Usability Problem and Usability Risk

Usability problems are defined as a possible threat to an optimal user experience and vulnerabilities as exposure of users to the threats. Therefore, usability problems can considered as significant risk factors for detrimental user experience [8]. Studies also had shown that poor security usability actually represents a serious vulnerability in a system and vulnerability is an attribute of risk [53].

In term of usability, risk can be defined as "users do not make use of a product" [52]. The term usability risk was first introduced in the context of e-commerce and WWW services. Usability risk is the potential that a chosen action or activity lead to a loss or an undesirable outcome which could impact the usability of a software product. It is related to user acceptance and meeting user's requirement. Usability risk is also an important factor for software product failure because it is not related to technical product quality but it is a problem that occurs in product use which leads to negative user experiences [20]. It was found that usability problems impact the usability and overall quality of a software problem. Therefore, it can be concluded that usability problem is significant risk factor in producing quality and usable software products.

Usability risks need to be managed well to reduce software product failure and produce more usable software product. But, there are great ignorance on managing usability risk compared to managing other risks such as technology risk, market risk and money risk [44].

D. Software Risk Management

Risk management is a set of activities used to manage risks. Many literatures have presented several risk management frameworks with different activities to manage risk.

The holistic approach in risk management involved six activities: risk identification, risk strategy and planning, risk assessment, risk mitigation/avoidance, risk reporting and risk prediction [23]. Software risk management was presented as two primary steps. These two steps are risk assessment (which includes risk identification, risk analysis, and risk prioritization) and risk control [24].

Cooper and Chapman have approached risk management as a multiphase `risk analysis' which covers identification, evaluation, control and management of risks [25]. In [26], risk management approach has been formulated as steps consisting of risk identification, risk analysis and risk response.

On the other hand, Richard Fairley stated seven activities to manage software product risk [27]. These steps are including identifying risk factors, assessing risk probabilities and its effects on the project, developing strategies to mitigate the identified risks, monitoring risk factors, invoking a contingency plan, manage the crisis, and recovering from a crisis.

The European Community promoted a comprehensive risk management methodology (RISKMAN) consisting of several phases, including risk identification, risk assessment, risk evaluation, risk mitigation, contingency estimate, decision making and control and monitoring. The RISKMAN provides a more comprehensive framework to enumerate and assess potential risk factors associated with a project [28].

Ian Summerville [29] has defined risk management as an approach with four basic processes which is Risk Identification, Risk Analysis, Risk Planning/Mitigation, and finally Risk Monitoring and Controlling. Other than that, methodology governing five core elements namely risk identification, risk measurement, risk assessment, risk evaluation, risk control and monitoring was also developed and called as the Risk Management Process (RMP) [30].

It was noticed that some study on risk management did not attempt to develop the risk management approach in a systematic and structured fashion and some focused on the measurement stage in their discussion [31]. Most of the risk management frameworks focus on the software process risks and not software products risks [32].

Even though the approaches are different from each other, common core activities/phases can be identified among these approaches. These core activities include risk identification, risk analysis, risk prioritization, risk mitigation and risk monitoring.

E. *Risk Identification*

Risk identification is the first step in software risk management process. The objective of risk identification is to determine the risk factors that may affect the project or product and document their characteristics. Risk factor is event or a situation that

increases the occurrence possibility of a risk incident. Identifying the risks early is the key to minimizing the effects or avoiding the effects of the risks altogether.

There are many techniques for identifying risks including interviewing, brainstorming, voluntary reporting, decomposition, assumption analysis, critical path analysis, and utilization of risk taxonomies [22]. Risks can be identified by interviews with selected experienced project managers or with experts in the field. The interviewees identify risks on the project based on their experience, the project information, and any other sources that they find useful [33]. This technique involves a lot of time.

Brainstorming techniques also has been used to identify risk factors in software product. Brainstorming is a technique by which a group attempts to generate ideas or find a solution for a specific problem [29]. However, planning the brainstorm session may be hassle.

Assumptions analysis is a technique that explores the assumptions' accuracy used in the project development plan. It identifies risks to the project from inaccuracy, inconsistency or incompleteness of assumptions. However this technique requires good analytical skills.

To identify risk factors, others techniques that can be used are Delphi, Checklist, Time a fact-finding and diagram techniques [34][35]. Diagram techniques include cause and effect diagrams, influence diagram and process flow diagrams. The Delphi technique is a method by which a consensus of experts can be reached on a subject such as project/product risk. The experts are identified but participate anonymously [36]. The Delphi technique helps reduce bias and minimizes the influence of any one person on the outcome. This technique could be costly.

On the other hand, checklist is a quick way to identify risks in a new project or product development by referring checklists of risks prepared based on information collected from past projects [37]. A checklist should not be considered as complete and the possibility of other risks should be addressed. The results obtain from checklist technique may be too general.

Observation and documentation analysis also used to identify risks. An observation technique enables interaction between researchers and the subjects during the study execution. During these interactions, the data were collected in a systematic and unobtrusive way, enabling the capture and document of potential risks. Where else,

documentation analysis is another source of evidence of possible mistakes and risks. All of this documentation can be used to avoid the occurrence of known problems and risks. It is also important to highlight that the documentation must always be updated [38].

Strengths, weaknesses, opportunities and threats (SWOT) analysis is also used to increase the breadth of the identified risks from each of the SWOT perspectives [39]. Other listed common risk identification methods are Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Hazard and Operability study (HAZOP) and Failure Mode Effect Analysis (FMEA) [40].

In [41], other risks identification steps also has been recognized such as use of tools, providing risk scenario or risk description and applying continuous and early identification method.

3. Methodology

The primary aim of this paper is to identify potential usability risk during software development process. Reducing identified usability risk could improve quality and increase usability of a software product to fulfil the expectations of its users. This also could create awareness in development team on usability risk that need to be reduced or eliminated during software development process.

In 2012, Aman Kumar, Arvind and Hardeep suggested that factors affecting the quality of software can be identified from attributes defined in software quality models [1]. Taking this idea, this paper considers attributes of usability as factors in producing usable software products. These usability attributes are subjected to risks that a software product might have troubles in that area. Scenario which affects the ability to achieve these attributes is considered as potential usability risk during software development process.

To enable identification of usability risk from usability attributes, it is important to determine integrated usability attributes since there are different literatures describing usability in various definition, attributes and models. Integrated usability attributes could help to decide whether the particular software system that being developed is usable. Efforts to derive integrated usability attributes have been done by Sanjay, Anubha and Ajay. They have suggested an integrated usability model that describes overall concept of software usability and explains it by means of a detailed taxonomy [42].

For this reason, 19 software quality models and 4 usability model for e-government is integrated to derive integrated usability attributes. Usability models for e-government was included because the scope of this study also covers usability of e-government applications.

Table 1. The selected software quality models and usability model for e-government.

Quality model	Usability model for e-government
1) ISO 9241-11 [68]	1) Quality in Use Integrated Measurement (QUIM) [47]
2) McCall [3]	2) Usability Assessment Framework of Haptic System (Haptic) [48]
3) Boehm [54]	3) Usability Maturity of Open Source-Model (OS-UMM) [49]
4) Shackel [55]	4) The Quality of Sustainable e-Government Development (QSeD) [50]
5) Constantine & Lockwood [46]	
6) Preece *et al*. [60]	
7) FURPS [57]	
8) IEEE Std. 1061 [58]	
9) Nielsen [59]	
10) Preece *et al*. [61]	
11) ISO 9126-1 [69]	
12) Dix *et al*. [62]	
13) Donyaee *et al*. [63]	
14) Bevan *et al*. [56]	
15) Abran *et al*. [45]	
16) Bass *et al*. [64]	
17) Dubey *et al*. [67]	
18) Schneiderman *et al*. [65]	
19) Alonso-Rios *et al*. [66]	

Models above was analysed to determine usability attributes mentioned in each models and based on this integrated usability attributes are derived.

After a detailed analysis on integrated usability attributes, a list of potential usability risk during software development process are produced. This risk is listed based on own perception but it will be compared with risks identified from usability test from literature, followed with an empirical research which will be conducted in Malaysian Public Sector to validate the identified usability risks.

4. Findings

A. Integrated Usability Attributes

After analyzing usability attribute in each models stated in Table 1, integrated usability attributes are suggested as *Effectiveness, Efficiency, Satisfaction,*

Comprehensibility, and Safety. Many models have included these attributes in their quality models and the summary is shown in Table 2.

Table 2.Suggested integrated usability attributes and total models that had mentioned these integrated usability attributes in its model

	Integrated usability attributes				
	A	B	C	D	E
Quality model	17	12	14	15	5
Usability model for e-government	4	3	4	4	2

A= Effectiveness, B= Efficiency, C= Satisfaction, D= Comprehensibility, E= Safety

From Table 2, it shows that most quality and usability models have included effectiveness, efficiency, satisfaction and comprehensibility as one of the important attributes of usability. Even though safety was not included as usability attributes in most quality and usability model, this study will include this attribute because it critically important that a user are protected against unintended actions or mistakes.

B. Potential Usability Risk

As discussed earlier, there are five integrated usability attribute: effectiveness, efficiency, compreensibility, satisfaction and safety. Based on these attributes, potential usability risks have been proposed as shown in **Table 3**.

From Table 3, it has been discovered that there are three potential usability risks which could affect satisfaction of a user and efficiency of a software product respectively, five potential usability risks which affects safety of a software product, seven potential usability risks that affects effectiveness of a software product and six potential usability risks that affects comprehensibility aspect of a software product. This table also shows that most usability risks lies under the attribute of effectiveness.

Table 3: Potential Usability Risk

Attributes	Potential Risk
Effectiveness	Low percentage of task accomplishment
	Incorrect task execution
	Incomplete functionalities to perform a task
	Lack of cultural diversity in user interface
	Inability to adapt to changing user preferences and environment
	High ratio of failure/errors • Human error • Execution error
	Lack of user control
Efficiency	Incorrect or inaccurate result produced
	Lack of utilization of command
	Longer execution time of a task
Satisfaction	Lack of software stability
	Lack of trust on software
	Lack of aesthetic features and good UI design
Comprehensibility	Lack of clarity in system's properties and functionalities
	Lack of skills for user and developer
	Inadequate training for user
	Longer time to learn the software
	Incomplete and Inadequate documentation/ user manuals
	Insufficient support system (help)
Safety	Loss of information/data
	High vulnerability to threat
	High prone to system failure/corruption
	Environment prone to hazards
	Change in environment

5. Conclusion and future works

Software usability had become an essential and crucial factor in determining overall quality of a software product. Realizing this, many studies have suggested approaches in integrating usability activities and techniques into software development process but there are still many constraints in its implementation.

This paper proposes a concept in ensuring high usability in software products by using the concept of risk management. The idea of this concept is that if usability risks can be identified and managed well, the overall chances of reducing risk of failure and producing usable software product could be increased. Since studies on usability risk is still lacking, this will be a good approach for development team to be aware of potential usability risk that must be managed during software development process. As mentioned in the literature review, the four basic processes in risk management are risk identification, risk analysis, risk mitigation and risk monitoring. This paper focuses in the first process, risk identification.

To identify usability risks, crucial and decisive usability attributes (from integrated usability models) has been recognized as effectiveness efficiency, satisfaction, comprehensibility and safety. Based on this attributes, potential usability risks are deduced with the perception that usability risks are activities that leads to failure in achieving usable software product. Using attribute in determining usability risks benefits the software development team and academicians for accessing precise usability risk during software development process. A total of 24 risks had been identified.

In future, the identified usability risk will be compared with risks identified from usability test, followed with an empirical research which will be conducted in Malaysian Public Sector to validate the identified usability risks. Risk identification techniques such as interview, survey and brainstorming will be used. Once all usability risks have been identified and validated, each usability risk will be analyzed by determining likelihood and impact in producing a usable product. This will be later used to formulate a Usability Risk Model. Finally this model will be mapped against Software Development Lifecycle (SDLC) to show the impact of usability risks in software development process.

References

[1] Sharma, A.K., A. Kalia, and H. Singh, *An Analysis of Optimum Software Quality Factors*, IOSR Journal of Engineering Apr. 2012, Vol. 2(4) pp: 663-669.

[2] Henry, S. and S. Wake, *Predicting maintainability with software quality metrics.* Journal of Software Maintenance: Research and Practice, 1991. **3**(3): p. 129-143.

[3] McCall, J. et al.: *Factors in software quality*. Vol. 1,2,3. AD/A-049- 014/015/055. Nat. Tech. Inf. Service. Springfield. 1977.

[4] Idyawati, H., Mahmud, M., Yeo, A. W.: *HCI practices in Malaysia: a reflection of ICT professionals' perspective*. In: Fourth International Symposium on Information Technology, Kuala Lumpur,15–17 June 2010..

[5] Douglas, I. and Z. Liu, *Global usability*. 2011: Springer.

[6] Dubey, S.K. and A. Rana, *Analytical Roadmap to Usability Definitions and Decompositions.* International Journal of Engineering Science and Technology, 2010. **2**(9): p. 4723-4729.

[7] Modernization and Management Planning Unit (MAMPU), *Kajian Baseline Perkhidmatan E-Kaunter & Paperless Government,* 2011.

[8] Inversini, A., L. Cantoni, and D. Bolchini, *Connecting usages with usability analysis through the user experience risk assessment model: A case study in the tourism domain*. 2011. p. 283-293.

[9] Isa, W.A.R.W.M., et al., *Assessing the Usability and Accessibility of Malaysia E-Government Website.* American Journal of Economics and Business Administration, 2011. **3**(1): p. 40-46.

[10] Jokela, T., *Performance rather than capability problems. Insights from assessments of usability engineering processes.* Product Focused Software Process Improvement, 2005: p. 115-127

[11] Lindgaard, G., *Usability testing and system evaluation: A guide for designing useful computer systems.* 1994: Chapman & Hall London,, UK.

[12] Durrani, Q.S., S.A. Qureshi, Lindgaard, G. *Usability Engineering Practices in SDLC.* 1994.

[13] London: Chapman and Hall, *Usability testing and system evaluation - A guide designing usable computer systems*, 1994.

[14] Ferre, X. *Integration of usability techniques into the software development process*. in *International Conference on Software Engineering (Bridging the gaps between software engineering and human-computer interaction)*. 2003.

[15] Fischer, H. *Integrating usability engineering in the software development lifecycle based on international standards.* in *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*. 2012: ACM

[16] Heiskari, J., et al. *Bridging the gap between usability and requirements engineering*. in *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International*. 2009: IEEE.
[17] Carlshamre, P. and M. Rantzer, *Business: Dissemination of Usability: Failure of a Success Story.* Interactions, 2001. **8**(1): p. 31-41.
[18] Seffah, A., R. Djouab, and H. Antunes. *Comparing and reconciling usability-centered and use case-driven requirements engineering processes.* in *Australian Computer Science Communications*. 2001: IEEE Computer Society.
[19] Durrani, Q.S. and S.A. Qureshi, *Usability Engineering Practices in SDLC.* 2012.
[20] Ketola, P., *Integrating usability with concurrent engineering in mobile phone development*. 2002: University of Tampere.
[21] C. Ravindranath Pandian, *Applied software Risk Management,* Auerbach Publications, 3rd edition, 2007.
[22] Linda Westfall, *Software Risk Management*, International Conference on Software Quality, San Diego, California, February 8-10, 2011.
[23] Karolak, Dale Walter, *Software Engineering Risk Management*, Computer Society Press, 1996.
[24] Barry W. Boehm, *Software Risk Management Principles and Practices*, Defense Advanced Research Projects Agency, IEEE Software, 8(1): 1991, PP. 32-41.
[25] Cooper, D. and Chapman, C., *Risk Analysis for Large Projects Models, Methods and Cases*. John Wiley, UK, 1987.
[26] Hayes, R. W., Perry, J. G., Thompson, P. A. and Willmer, G., *Risk Management in Engineering Construction Implications for Project Managers.* Project Management Group, UMIST, Thomas Telford Ltd., UK, 1987.
[27] Richard Fairley, *Risk Management for Software Projects*, Journal IEEE Software, Volume 11 Issue 3, pp. 57 – 67, May 1994.
[28] Carter, B., Hancock, T., Morin, J. and Robin, N., *Introducing RISKMAN: The European Project Risk Management Methodology*. NCC Blackwell Limited, UK, 1994.
[29] Ian Summerville, *Software Engineering*, Addison Wesley, 7th Edition, 2007.
[30] Tummala, V. M. Rao, Nkasu, M. M. and Chuah, K. B., *A systematic approach to risk management.* Journal of Mathematical Modeling and Scientific Computing, 1994, 4, 174±184.
[31] Tummala, V. and Burchett, J., *Applying a risk management process (RMP) to manage cost risk for an EHV transmission line project.* International Journal of Project Management, volume 17, pp.223-35.1999.
[32] Halima M. Mofleh And Ammar. Z, *A Framework For Software Product Risk Management Based On Quality Attributes And Operational Life Cycle (SPRMQ)*, 2010.

[33] Obrand, L., Augustsson, N., Holmstrom, J., & Mathiassen, L., *The Emergence of Information Infrastructure Risk Management in IT*, 2012. 45th Hawaii International Conference on Services.
[34] Wei Xiang, *Software Engineering Project Management*, Study Book, Published by University of Southern Queensland, Austalia, 2nd Edition, 2007.
[35] Jakub Miler, *A Method of Software Project Risk Identification and Analysis*, PhD Thesis, Faculty of Electronics, Telecommunications and Informatics, GDAŃSK University of Technology, 2005.
[36] Liu S, Zhang J, Keil M, Chen T. *Comparing senior executive and project manager perceptions of IT project risk: a Chinese Delphi study*.Info Syst J 2010;20:319–55.
[37] Pressman, R.S.: *A practice's approach*, 6th edn. (2005).
[38] Lobato, L. L., da Mota Silveira Neto, P. A., do Carmo Machado, I., de Alemida, E. S., & de Lemos Meira, S. R., *Risk management in software product lines: An industrial case study*, 2012 .
[39] Hillson, D., *Extending the risk process to manage opportunities.* International Journal of Project Management, 2002. **20**(3): p. 235-240.
[40] Mokhtari, K., et al. *A preliminary research on risk management for marine industry applications, Quality, Reliability, Risk, Maintenance and Safety Engineering (ICQR2MSE)* International Conference, 2012.
[41] Lobato, L.L., et al. *Risk Management in software engineering: A scoping study*, Evaluation & Assessment in Software Engineering (EASE 2012), 16th International Conference, 2012.
[42] Dubey, Gulati, A., A. Rana, *Integrated Model for Software Usability*, Vol. 4 No. 03 March 2012 International Journal on Computer Science and Engineering (IJCSE).
[43] Hafizah Yahya and Rozilawati Razali, *Usability Models for Electronic Government – A Preliminary Review*, UMT 11th International Annual Symposium on Sustainability Science and Management 09th – 11th July 2012, Terengganu, Malaysia.
[44] Platt, A.B. *The usability risk. in Reliable Distributed Systems*, Proceedings of the 18th IEEE Symposium on. 1999.
[45] Abran, A., et al., *Usability Meanings and Interpretations in ISO Standards.* Software Quality Journal, 2003. 11(4): p. 325-338.
[46] Constantine, L.L. and Lockwood, L.A.D. 1999. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centred Design*, New York: Addison-Wesley.
[47] A.Seffah, et al., QUIM : *A Framework for Quantifying Usability Metrics in Software Quality Models*, IEEE, 2001.

[48] M. Khan, et al. : *Development of usability evaluation framework for haptic systems, in VR Innovation* (ISVRI), 2011 IEEE International Symposium on, 2011, p. 343-344
[49] U. N. D. o. E. a. S. A. U. *Nations in: United Nations E-Government Survey*, 2010
[50] W. Chutimaskul, et al.: *The quality framework of e-government development*, ICEGOV '08 Proceedings of the 2nd international conference on Theory and practice of electronic governance p. 105-109, 2008.
[51] McFarlan W., Portfolio *Approach to Information Systems*. Journal of System Management, January, 12-19, 1982.
[52] Altom, T., *Usability as risk management.* Interactions, 2007. **14**(2): p. 16-17.
[53] Josang, A., et al. *Security Usability Principles for Vulnerability Analysis and Risk Assessment*. in *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*. 2007.
[54] Boëhm, *Characteristics of software quality*, Vol 1 of TRW series on software technology, North-Holland, Amsterdam, Netherlands, 1978.
[55] B. Shackel, *Usability – Context, framework, definition, design and evaluation*. In Human Factors for Informatics Usability, ed. Brian Shackel and Simon J. Richardson, 21–37. New York, Cambridge University Press, 1991.
[56] N. Bevan, J. Kirakowski & J. Maissel, *What is usability?*, Proceedings of the 4th International Conference on HCI, 651–655, 1991
[57] R. B. Grady, *Practical Software Metrics for Project Management and Process Improvement*, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.
[58] Institute of Electrical and Electronics Engineers. IEEE standard glossary of software engineering terminology, IEEE std. 610.12-1990.Los Alamitos, CA: Author, 1990
[59] J. Nielsen, *Usability engineering*. London: Academic Press, 1993.
[60] J. Preece, D. Benyon, G. Davies, L. Keller and Y. Rogers, *A guide to usability: Human factors in computing*. Reading, MA: Addison-Wesley, 1993.
[61] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland and T. Carey, *Human-computer interaction. Reading*, MA: Addison-Wesley, 1994.
[62] A. Dix, J. Finley, G. Abowd and R. Beale, *Human-Computer Interaction*, 2nd ed. Prentice-Hall, 1998.
[63] M. Donyaee and A. Seffah , QUIM: An Integrated Model for Specifying and Measuring Quality in Use, Eighth IFIP Conference on Human Computer Interaction, Tokyo, Japan, 2001.
[64] L. Bass and B. E. John, *Linking usability to software architecture patterns through general scenarios*. Journal of Systems and Software, 66 (3), 187-197, 2003.

[65] B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison Wesley, Boston, MA, 2005.
[66] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey and V. Moret-Bonillo, *Usability: A Critical Analysis and a Taxonomy*, International Journal of Human-Computer Interaction, 26(1), 53–74, 2010.
[67] S. K. Dubey, A. Sharma and A. Rana, *Usability evaluation in object oriented software systems using fuzzy logic approach*, International Journal of Computer Science Issues, 2011, "in press".
[68] International Organization for Standardization. ISO 9241-11:1998, *Ergonomic requirements for office work with visual display terminals (VDTs), Part 11: Guidance on usability*. Geneva, Switzerland: Author, 1998.
[69] ISO 9126: *Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for their Use*. Geneva, 1991.