# Identifying Mitigation Plans in Reducing Usability Risk Using Delphi Method

Jayaletchumi T. Sambantha Moorthy, Suhaimi bin Ibrahim, Mohd Naz'ri Mahrin

**Abstract**—Most quality models have defined usability as a significant factor that leads to improving product acceptability, increasing user satisfaction, improving product reliability, and also financially benefitting companies. Usability is also the best factor that balances both the technical and human aspects of a software product, which is an important aspect in defining quality during software development process. A usability risk consist risk factors that could impact the usability of a software product thereby contributing to negative user experiences and causing a possible software product failure. Hence, it is important to mitigate and reduce usability risks in the software development process itself. By managing possible usability risks in software development process, failure of software product could be reduced. Therefore, this research uses the Delphi method to identify mitigation plans for reducing potential usability risks. The Delphi method is conducted with seven experts from the field of risk management and software development.

**Keywords**—Usability, Usability Risk, Risk Management, Risk Mitigation, Delphi Method.

## I. INTRODUCTION

USABILITY is defined in various quality models as an important factor in ensuring the development of a quality and usable software product. Ignorance and unawareness about the concept of usability and failure to address usability during the software development process leads to an overall failure of the software product due to poor quality and high usability problems. Usability problem is perceives as an aspect of the system and/or a demand on the user which makes it unpleasant, inefficient, onerous, perturbing or impossible for the user to achieve their goals in typical contexts of use [1].

Presence of usability problems are reflected when usage level of software products is low [2] and other reported usability problems are such as high number of broken links and slower accessibility speed [3]; less 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 and inappropriately used usability methods [4]. Generally, existence of usability problem could lead to failure of a system [5].

Usability professionals have expressed that by integrating usability closely with software development process [6], a more usable software products with less usability problems can be developed. There are many efforts to integrate a formal

Jayaletchumi T. Sambantha Moorthy, Suhaimi bin Ibrahim and Mohd Naz'ri Mahrin are with the Universiti Teknologi Malaysia (UTM), Kuala Lumpur, 54100 Malaysia (e-mail: tsmjayaletchumi2@live.utm.my; suhaimiibrahim@utm.my and mdnazrim@utm.my ).

usability process, standard, techniques and practices into software development process to improve the interaction and quality of the systems (e.g. [7]-[10]. However, software developers face difficulties when new usability practices are introduced during development process [11]. Some usability approaches are only integrated in requirement and design phase [12]. In fact, their practical implementation is largely missing [7]. Usability practices as well are not part of requirement engineering [9], so developers are often given an incomplete, confusing, and sometimes contradictory requirement. As a result, many development teams are facing difficulties in avoiding and minimizing usability problems.

Besides this, various usability evaluation activities such as inspection, empirical testing, and metrics for usability standards in computing has been integrated into software development process to measure and improve usability of software [13]. However, it only evaluates a completed system and does not intervene at earlier stages of development process [14]. An International standard, ISO 13407 [15] also had proposed a framework for integration of usability in all phases of software development process. This standard uses the approach of User-Centered Design (UCD) that focuses specifically on developing usable system. Research works are still in progress to introduce the best methods for reducing usability problems while increasing the rate of successful and usable software products but however, current practices face various challenges in reducing the usability problems.

Alternatively, the concept of risk management can be used to control usability problems even though these problems cannot be totally eliminated. In the concept of risk management it is important to deal with possible usability problems before they occur and by managing the usability risks involved in software development, the possible failure of a software product could be reduced. Hence, a development team needs to identify plans to manage and mitigate potential usability risks at the earlier phases of development process itself. Currently, development teams are facing challenges in understanding the ways to mitigate usability risks in the early stages of the development process. If development teams continue to develop software products without mitigating usability risks during software development process, the chances of producing less usable software products are higher.

Usability risk was first introduced in the context of e-commerce and World Wide Web services [16]. Some studies which are related to mobile applications have mentioned the term 'usability risk' in their studies [17-19].The search for the term 'usability risk' shows that this term is not widely used.

Most of the research done is regarded as a usability problem and not on usability risk.

This paper puts forward results from the Delphi method that identifies mitigation plans for potential usability risk which can help a development team to mitigate potential usability risks that may occur during Software Development Lifecycle (SDLC). This research uses the Delphi method whereby a number of experts from software development and risk management field are selected as respondents. A mitigation plan for some usability risks is presented in this paper. Project managers, quality managers, risk management teams and software development teams may benefit with the results of this research in ensuring identified usability risks are mitigated in each phase of the software development lifecycle. This contributes to the development of more usable and productive software products.

## II. METHODOLOGY OF RESEARCH

The research presented in this paper is conducted in five steps as discussed below:

### Step1. Construction of Questionnaire for Delphi Study

The questionnaire is designed based on guidelines provided by Mark Kasunic [20]. This research considers the attributes of usability as factors for producing usable software products. The usability risk that has been identified and derived from the usability attributes and the sub attributes defined in software quality models [21], are used as constructs for this questionnaire. The initial list of usability risk has been published our previous works [22].

### Step2. Pilot Study of Questionnaire

Piloting the Delphi study is an essential step to ensure earlier detection and correction of inflexibility in terms of the design of the questionnaire, the measurement and the analysis method, before the actual Delphi process is begun. This is an important precautionary step to increase the validity and reliability of the results that would be obtained from the Delphi Study. The participants for the pilot study are selected using the characteristics of expert sampling and snowball sampling. These participants are to meet with some of the predefined criteria for experts and are asked to recommend other suitable participants for this pilot study. Even though the participants are not wholly to the definition of experts in the actual Delphi study, they are still selected based on their experience and knowledge of software development and/or handling risk management, which is the core criteria and therefore is considered sufficient for a pilot study. A sample of nine participants is selected to participate in the pilot study with four of the participants being ICT practitioners and the rest are from a pool of academicians. All the participants have experience in the area of software development and/or risk management and are experienced in dealing with software development projects. These selected participants for the pilot study are required to respond to the questionnaires which are sent to them by email. These respondents are also required to provide specific feedback on the questionnaire constructs, the

rating scale, the required time in responding and also to other related challenges. The duration between iterations is observed in the pilot study.

### Step 3. Expert Selection Process

The expert selection process is the main critical phase in the Delphi Study because the quality of experts chosen determines the quality of the results to be obtained [23]. The main aim of the expert selection process is to identify individuals in the Public Sector in Malaysia that match the criteria that are included in this study. The expert selection process involves three stages namely the Definition of Experts, the Number of Experts and the Actual Selection Process.

### Step4. Delphi's 1st Round

The questionnaires are sent by email to the panel of experts and each expert is given sufficient time (2 weeks) to complete and return the questionnaire with their responses. Since their responses are solely their own without any influence from the members of the selected panel of experts, the quality of the feedback given on the mitigation plans for usability risks is totally based on their own experience. The responses on the mitigation plans from Delphi's 1st Round are then compiled in a report and used in Delphi's 2nd Round.

### Step5. Delphi's 2nd Round

The analyzed responses from the Delphi's 1st Round are consolidated and summarized in the form of a report and then given to the experts. This is an opportunity for the experts to further review the pool of responses and change their responses if necessary. During this process, any ambiguous responses given by the experts are neglected. Fig 1 shows the overall methodology of the research.
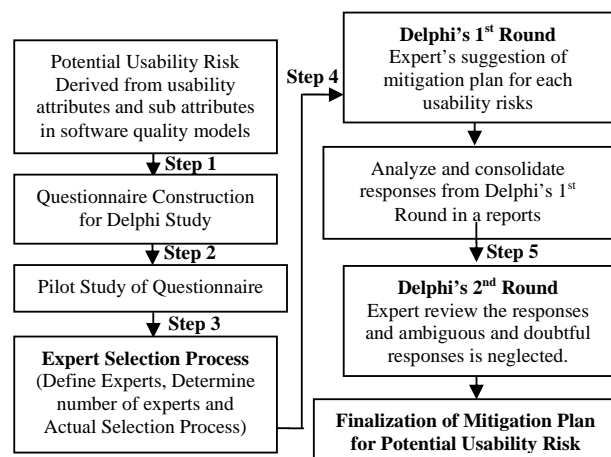


Fig. 1 Methodology of Research

## III. RESULT AND DISCUSSION

### Step 1. Construction of Questionnaire for Delphi Study

In our previous study [22], the integrated usability attributes were identified as Effectiveness, Efficiency, Satisfaction, Comprehensibility and Safety. The usability risk factors were

derived from the usability attributes and their sub attributes. These usability risk factors were used to determine the potential usability risk. Along with usability risks suggested by ICT practitioners, a final list of 42 potential usability risks were determined and the Delphi method was then used to identify the mitigation plans for the potential usability risks.

*Step 2. Pilot Study of Questionnaire*

The respondents are required to provide some suggestions on the possible mitigation steps for the given usability risks. From the nine selected participants, only 7 respondents participated in this pilot study. Based on the feedback given, instructions of the questionnaire are refined to be clearer, more detailed and precise. It is believed that the refined instructions would increase the understanding and ability of the respondents in answering the questionnaire effectively.

*Step 3. Expert Selection Process*

Since the credibility of the Delphi study lies in its ability to identify experts that fulfill a defined criteria which is suitable for the Delphi study [24], a strict set of criteria is used to ensure the respondents are real 'experts' in the field under investigation [25]. Besides that, the initial decided number of experts to participate in this Delphi study is reduced from ten to seven. The number of experts is limited to seven due to time constraints in conducting this study. The same experts are sustained for the duration of the research and their views are given equal weightage.

To obtain the targeted sample size of experts in this research, the experts are purposively sampled, using a combination of the expert sampling and the snowball sampling. Since the number of experts with experience in software development and/or risk management is unknown and it is difficult to locate the required experts among the people, snowball sampling is used to penetrate the unknown population and the selection of experts are done on referral basis. A total of 10 experts are approached and given a specific form to compile their experience and knowledge. After reviewing the details of the experts, only seven experts are found to be suitable and therefore fulfill the required criteria. The number of experts is small but diverse. Initially, each expert is briefed on the nature, goal, and purpose of the study and as to how the outcome of the Delphi Study is going to prove useful.

*Step 4. Delphi's 1st Round*

The questionnaire was sent by email to the panel of experts and each expert was given sufficient time (2 weeks) to complete and revert their responses. Since experts responses are not influenced by each other, the quality of feedback given on mitigation plan for usability risks is totally depended on their experiences. Responses on mitigation plan from Delphi's 1st Round were then, compiled in a report and used in Delphi's 2nd Round.

*Step 5. Delphi's 2nd Round*

The compiled mitigation plans are sent to the experts to get their acceptance or rejection on any unrelated information.

The changes recommended are analyzed and the amendments made in the initial stage on the mitigation plans are for troubleshooting potential usability risks.

At the end of the Delphi study, the mitigation plan for each usability risks is identified. For illustration purposes, the identified mitigation plan for some usability risk is as listed in Table I.

TABLE I
MITIGATION PLAN FOR USABILITY RISK

| No | Usability Risk | Mitigation Plan |
|---|---|---|
| 1 | Low percentage of task accomplishment | • Prepare to perform effective User Acceptance Testing (UAT) and other functional testing of the software before it is released for operational use.<br>• Establish realistic project timeline to enable developer ensure full task accomplishment<br>• Have proper dedicated development team to increase task accomplishment<br>• Review the user requirements and functional requirements. Enhance the system if required.<br>• Provide sufficient training (programming skill) to developer, if required to ensure complete task accomplishment in software.<br>• Perform simulation runs<br>• Plan for familiarization sessions |
| 2 | Inappropriateness of the task output | • Establish testing plan–functional, System Acceptance Testing<br>• Stakeholder / User engagement during software development<br>• Instil requirement sign-off process to ensure<br>• Have walk trough/inspection regularly<br>• Establish Quality Team and build in Quality Assurance (QA) process during development<br>• Review the data flows and functional requirements. |
| 3 | Incorrect task execution | • Prepare software testing plan and conduct proper independent testing<br>• Establish Quality Assurance (QA) process during development<br>• Review the data flows, functional requirements and coding. |
| 4 | Incomplete functionalities to perform a task | • Review user requirements and system design specifications to ensure all functionalities are identified and to establish a comprehensive URS<br>• Include right expert in the software development process.<br>• Establish quality assurance (QA) process during development<br>• Design verification by business owners and establish sign-off design document<br>• Use *Use Case* and Functionality Acceptance Testing to test on the software functionality<br>• Identify potential users of system and involve some of them during system functional testing |
| 5 | Low usage by user with disabilities | • Adopt the W3C Web Content Accessibility Guidelines to add features that facilitates usage by users with disabilities)<br>• Establish Quality Assurance (QA) process during development<br>• Review and prepare complete Requirement Engineering by building this requirement in the URS.<br>• Include this feature in systems/application design |
| 6 | Lack of cultural diversity in user interface | • Identify potential users of system, engage the users and include their expectation of requirements.<br>• Requirement Engineering (RE) and Systems/ application design should include cultural diversity features in user interface.<br>• Quality assurance should be implemented during development |

| No | Usability Risk | Mitigation Plan |
|---|---|---|
| 7 | Inability to adapt to changing user preferences and environment | • RE and Systems/application design should include features of user's changing preferences<br>• Quality assurance should be implemented during development<br>• Build flexibility in system design (not so rigid) |
| 8 | Lack of user control | • RE and Systems/application design should include requirements for user control<br>• Quality assurance should be implemented during development<br>• Define user roles and security requirements of system early in study<br>• Build in flexibility in system design (not so rigid) |
| 9 | High ratio of failure resulted from human errors | • Quality assurance should be implemented during development<br>• Create more drop-down menu and radio buttons to decrease failure from human errors<br>• Plan comprehensive functional testing.<br>• Have proper, complete and adequate operational documentation. Carry out regular audits<br>• Develop sufficient test cases to cover all possible scenarios<br>• Prepare proper functional testing plan and comprehensive test scripts |
| 10 | High ratio of failure resulted from execution errors | • Quality assurance should be implemented during development<br>• Sufficient time given for unit and integration testing<br>• Ensure the testers have the domain knowledge during testing.<br>• Check and review system's configuration and coding |

For example, the risk of, '*Incomplete functionalities to perform a task*' can be mitigated when the user requirements and the system design documents establish all required functionalities. Furthermore, including the right experts and establishing quality assurance process during the software development process would probably mitigate this risk. A verification of the design by business owners and by using *Use Cases* and *Functionality Acceptance Testing,* the possibility of the occurrence of usability risk could be prevented.

## IV. CONCLUSION

This paper explains in detail the process of identifying mitigation plans for usability risks using the Delphi method. The Delphi method involves a panel of experts who provide feedback on topics under discussion, based on their experience and knowledge and this method is useful in identifying mitigation plans because the results are found to be effective. The following are the major contributions and conclusions of this research:

- The proposed approach is an instrument for identification of mitigation plans for usability risks which can be used during the software development process.
- Mitigation plans provided in this paper are to mitigate usability risks. Hence, in future the mitigation plans for other risks can also be developed and validated by using the approach proposed in this paper.

### REFERENCES

[1] Lavery, D.a.C., Gilbert, "Representing Predicted and Actual Usability Problems," in *Proc.of International Workshop on Representations in Interactive Software Development*, 1997. QMW College, London

[2] 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, pp. 283-293.

[3] Isa, W.A.R.W.M., Suhami, M.R., Safie, N.I., and Semsudin, S.S., "Assessing the Usability and Accessibility of Malaysia E-Government Website," *American Journal of Economics and Business Administration*, 2011, no. 3(1), pp. 40-46.

[4] Jokela, T., "Performance rather than capability problems. Insights from assessments of usability engineering processes," 2005.

[5] Seffah, Ahmed, Donyaee, Mohammad, Kline, Rex B, and Padda, Harkirat K, "Usability measurement and metrics: A consolidated model," *Software Quality Journal*, 2006. vol 14(2), pp. 159-178.

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

[7] Durrani, Q.S. and S.A. Qureshi, "Usability Engineering Practices in SDLC," 2012.

[8] 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.

[9] Heiskari, J., Kauppinen, M., Runonen, M., & Mannisto, T., "Bridging the gap between usability and requirements engineering," in *17th IEEE International Requirements Engineering Conference RE'09*, 2009.

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

[11] Carlshamre, P. and M. Rantzer, "Business: Dissemination of Usability: Failure of a Success Story," *Interactions*, vol 8(1), pp. 31-41, 2001.

[12] Seffah, A., R. Djouab, and H. Antunes, "Comparing and reconciling usability-centered and use case-driven requirements engineering processes," in *Australian Computer Science Communications*, IEEE Computer Society, 2001.

[13] Humayoun, S., "Incorporating Usability Evaluation in Software Development Environments," *KI - Künstliche Intelligenz*, vol 26(2), pp. 197-200, 2012.

[14] Lilja, K.K., K. Laakso, and J. Palomki, "Using the Delphi method," in *Proceedings of PICMET'11, Technology Management in the Energy Smart World (PICMET)*, IEEE, 2011.

[15] ISO, "ISO13407 : Human-centred design processes for interactive systems," 1999.

[16] Platt, A.B., "The usability risk," in *Proc. of the 18th IEEE Symposium on Reliable Distributed Systems*," 1999.

[17] Ketola, P., "Integrating usability with concurrent engineering in mobile phone development," University of Tampere, 2002.

[18] Jones, T.C., "Assessment and Control of Software Risk," New York: Prentice-Hall, 1994.

[19] Jin, B., Ko, S., Mun, J., & Ji, Y, "A Study for Usability Risk Level in Physical User Interface of Mobile Phone," in *Usability and Internationalization. HCI and Culture*, N. Aykin (Ed), Springer Berlin Heidelberg, pp. 327-335, 2007.

[20] Kasunic, M., "Designing an effective survey," DTIC Document, 2005.

[21] Dubey, S.K., A. Gulati, and A. Rana, "Integrated Model for Software Usability," *International Journal on Computer Science and Engineering*, vol 4(3), 2012.

[22] Sambantha Moorthy, J.T., S. Ibrahim, and M. Naz'ri Mahrin, "Identifying Potential Usability Risk During Software Development Process," *Open International Journal of Informatics*, Vol 1, pp. 29-45, 2013.

[23] Day, J. and M. Bobeva, "A generic toolkit for the successful management of Delphi studies," *EJBRM*, vol 3(2), pp. 103-116, 2005.

[24] Miller, G., "The development of indicators for sustainable tourism: results of a Delphi survey of tourism researchers," Tourism management, vol 22(4), pp. 351-362,2001.

[25] Singh, A.J. and M.L. Kasavana, "The impact of information technology on future management of lodging operations: A Delphi study to predict key technological events in 2007 and 2027," Tourism and Hospitality Research, vol 6(1), pp. 24-37, 2005.