J. Inf. Technol. Appl. Manag. 25(3): 17~28, September 2018
https://doi.org/10.21219/jitam.2018.25.3.017
ISSN 1598-6284 (Print)
ISSN 2508-1209 (Online)

The Activity-Oriented Usability Model of Software

Seokha Koh* · You-Jeong Koh**

Abstract

In this paper, an activity-oriented usability model is proposed. The usability model contains two types of characteristics: special-type characteristics of usability and sub-characteristics of usability. Workability, study-ability, and playability are, but do not exhaust, examples of special-type characteristic of usability. They correspond to working, studying, and playing using the software product, respectively. They represent the goal of using and can overlap each other. They are usability too by themselves.

Navigate-ability, data-prepare-ability, data-input-ability, response-wait-ability, output-examine-ability, and output-utilize-ability are typical examples of sub-characteristics of usability. They correspond to navigating, preparing data, inputting data, waiting response, examining output, and utilizing the output data, respectively. They are not usability by themselves. They constitute usability together as a group.

Assessing is the fundamental and indispensable aspect of quality. Without assessing, the concept of quality has little practical value. Satisfaction, effectiveness, and efficiency are the most typical sub-characteristics of usability in existing quality models, which correspond to the evaluation criteria of usability. In the activity-oriented usability model, however, only the user's satisfaction is included: Satisfaction is regarded as the operational definition of usability in the user's view. As the result, usability can be interpreted as the 'goodness for using, which is evaluated by the user.'

Three fundamental principles regarding software quality models are proposed too in this paper: Principles of Parsimony, Cohesiveness, and Inheritance. Discussions illustrate well that typical existing usability models violate these basic principles. Many authors have tried to define general usability models which can be applied to most kinds of software. The dream of the general and universal usability model, however, may be an illusion. The activity-oriented usability model is expected to serve as a prototype from which specialized usability models can be derived.

Keywords : Software Quality, Usability, Quality in Use

Received : 2018. 06. 25. Final Acceptance : 2018. 09. 10.

* Corresponding Author, Department of Management Information Systems, Chungbuk National University, 1 Chungdae-ro Seowon-gu Cheongju, Chungbuk, 28644, Korea, Tel : +82-43-261-7062, Fax : +82-43-273-2355, e-mail : shkoh@cbnu.ac.kr
** UX specialist, Reflexion Corporation, e-mail : kohyou10@gmail.com

1. Introduction

The quality of something is, according to Collins Cobuild Advanced Learner's English Dictionary, '*how good or bad it is*.' [1] In this paper, the quality of a software product is defined as 'how good or bad is the product?'

A software product should be good for being used. In this paper, 'How good is a product for being used' or, shortly, 'goodness for using' is called usability of the product. Since being used is the ultimate goal of all kinds of product including software, usability traditionally has been one of the most important issues regarding software quality.

There are diverse definitions of usability [Boehm et al., 1978; Foraker Labs, 2002; González Sánchez et al., 2009b; Grady, 1992; ISO 9126-1 : 2001, 9241-11, 25010 : 2011, 25021 : 2012, 25022 : 2016, 25023 : 2016; Microsoft Corporation, 2000; McCall et al., 1977; Nielsen, 2012; Shuja and Krebs, 2008; U.S. Department of Health and Humanity Services, 2017].

It is noticeable that González Sánchez et al. [2009b] regard playability for video game as a special sub-type of usability. Some authors view usability as the quality of a user's experience [Gonzalez Sanchez et al., 2009b; Microsoft Corporation 2000; U.S. Department of Health and Humanity Services, 2017] while other authors see it as the quality of the product [Herrera et al., 2010; ISO 9126-1, 9241-11, 25000 Series; Nielsen, 2016].

1) In this paper, italic font emphasizes that corresponding part is quoted with no or only slight changes from the cited literature.

Koh and Jiang [2017] combine these two approaches to define two types of usability: the instance usability as the goodness of an individual user experience and the product usability as the aggregation of all instance usability associated with a product. In this paper, the term usability is used to encompass quality in use, for example, that of ISO/ISE 25000 Series SQuaRE.

An activity model that contains two categories of software activity types which are associated with using the software product is proposed in section 2. A usability model based on the activity model, which will be called the activity-oriented usability model, is proposed in section 3. In section 4, three principles are proposed and the validity of existing models and the activity-oriented model are discussed against the principles. The results shows well that the activity-oriented usability model will serve as a robust prototype on which on which usability models specialized according to the goal of using can be built.

2. Types of Activity Associated with Using Software

Koh and his colleagues [Koh, 2016, 2017a, 2017b; Koh and Jiang, 2017] define software activity as the activity which is performed on the software product by a person or a group of persons. They define using as a type of software activity, which is performed through user interfaces of the software product. They distinguish using form other types of software activity which engage the interactions between the software products and persons such as studying the pro-

duct itself (increasing the user's knowledge or expertise about the product itself), testing (finding out whether the product performs as intended or required), or customizing user interface (making the product better to use).

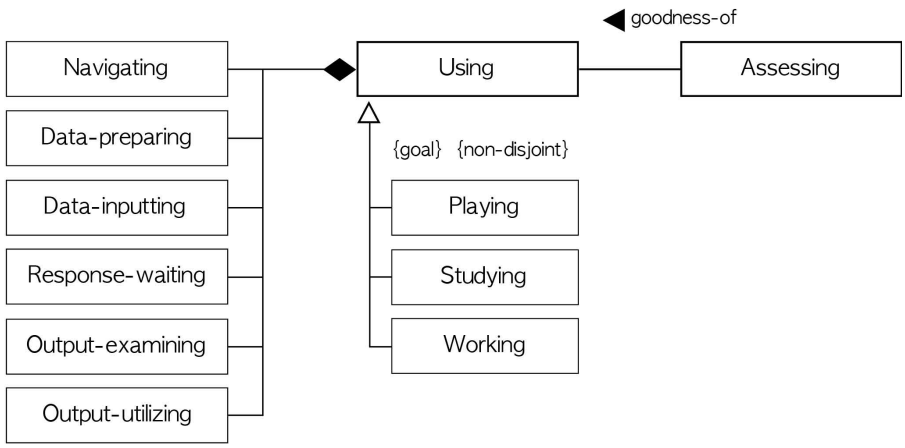
Working, studying, playing, navigating, preparing data, inputting data, waiting response, and examining output are typical types of software activity which are associated with using the software product [Koh and Jiang, 2017]. They can be classified into two categories: special types of using and component activities of using (refer <Figure 1>).

The classification criterion of working, studying, and playing is the goal of using. That is, people use the software product to accomplish some goals, for example, to do some tasks in the work place, to study some issues, or to play for fun. It is noticeable that these types are not mutually exclusive since the goals of using a software product can be multiple. Suppose that, for example, an education game software product is used both to study some topics and to play for fun [Koh and Jiang 2017]. Then, this instance of using is also an instance of studying and an instance of playing at the same time.

Navigating, preparing data, inputting data, waiting response, examining output data, and utilizing the output data are the typical examples of component activity type of using. A set of instances of these activity types constitute an instance of using. An instance of these activity types alone, however, seldom constitutes a whole instance of using. An instance of navigating alone, for example, seldom constitutes a whole instance of using.

It is noticeable that accessing is excluded from <Figure 1>. Accessing is the precondition of using. Accessing, however, can happen independently with using. A person can access a software product, for example, to test or customize the product. Accessing should be classified as a separate type of software activity independent of using.

Assessing is not a type of the software activity. Assessing or evaluating, however, is an essential and indispensable aspect of quality. According to Oxford Learner's and/or Living



〈Figure 1〉 The Activity Model Associated Using Software Products: A UML Diagram

Dictionaries, assessing means 'evaluating or estimating the nature, ability, of quality of,' and 'evaluating' means '*forming an opinion of the amount of something after thinking about it carefully*;' where empirical and theoretical means '*based on experiments or experience rather than ideas or theories*' and '*concerned with the ideas and principles on which a particular subject is based, rather than with practice and experiment*,' respectively. Assessment of the quality of a software product should be feedback into the development process or post-lifecycle change process of the product to improve the quality of the product. Without assessment, the notion of quality is of little use.

3. A Tentative Usability Model Based on the Activity Model

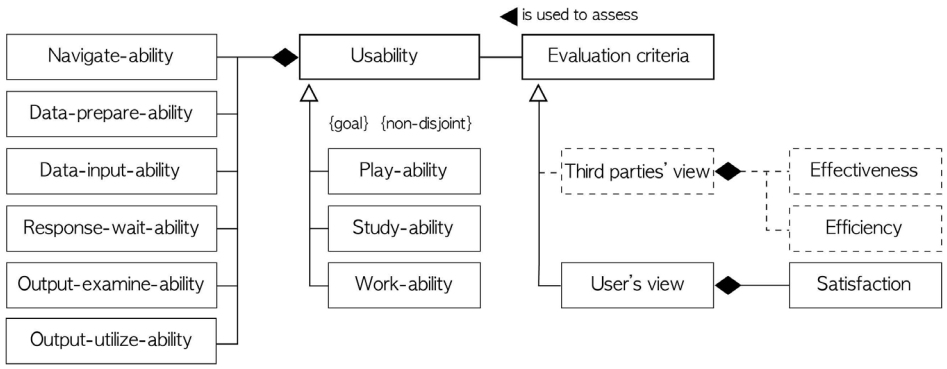
Effectiveness, efficiency, and satisfaction are the typical quality characteristics which are most frequently cited as sub-characteristics of usability in existing software quality models [Koh and Jiang, 2017]. However, they correspond to the criteria by which usability is evaluated (refer <Figure 2>). In this paper, they are not regarded as quality characteristics. As the result, they are not regarded as sub-characteristics of usability either.

Koh and his colleagues [Koh, 2016, 2017a, 2017b; Koh and Jiang, 2017] suggest to apply the suffix '-ability' to a software activity type (a verb) to denote the quality corresponding to the activity type; that is, the 'goodness for the corresponding activity.' They also propose to apply the suffix '-ability' to the software activity to avoid confusion. Adopting their naming scheme, two types of quality characteristics that correspond to the software activity categories in the activity model in <Figure 1> can be defined as follows:

- **Quality characteristic as a special type of usability**: The characteristic corresponding to a special activity type of using, for example, working, studying, or playing.
  - **Work-ability**: Goodness for working,
  - **Study-ability**: Goodness for studying.
  - **Play-ability**: Goodness for playing,
- **Sub-characteristic of usability**: The Sub-characteristic corresponding to a component type of using, for example, navigating, preparing data, inputting data, waiting response, examining output data, or utilizing the output data.
  - **Navigate-ability**: Goodness for navigating,
  - **Data-prepare-ability**: Goodness for preparing data to be input,
  - **Data-input-ability**: Goodness for inputting data,
  - **Response-wait-ability**: Goodness for waiting response,
  - **Output-examine-ability**: Goodness for examining output data.
  - **Output-utilize-ability**: Goodness for utilizing the output. data

Since special-type activities can be overlap each other, the special-type characteristics of usability can overlap each other too.

Usability can be evaluated either directly or indirectly:



〈Figure 2〉 A Tentative Usability Model: A UML Class Diagram

- **Sub-characteristics evaluation**: Evaluate navigate-ability (how good to navigate), data-prepare-ability (how good to prepare data), data-input-ability (how good to input data), response-wait-ability (how good to wait response), output-examine-ability (how good to examine output), and output-utilize-ability (how good to utilize the output data) first, and then aggregate them into usability.
- **Direct evaluation**: Usability is evaluated directly without evaluating its sub-characteristics.

Most existing quality models such as SQuaRE, adopt the direct evaluation. However, the sub-characteristics evaluation will provide much richer information as well as increase the validity and reliability of measurement.

As a special type of usability, workability, study-ability, or playability can be evaluated either directly or indirectly. Suppose that the goals of using a product are multiple. For example, a person may use an education game software product to both play and study. In this case, both playability (how good to play) and study-ability (how good to study) can be assessed separately, and then be aggregated into usability.

4. Discussions

Usability can be evaluated by many stakeholders. Koh and Jiang [2017] classify stakeholders into users and third parties:

- **User's view**: The usability assessed by the users.
- **Third parties' view**: The usability assessed by the stakeholders other than the users. This includes the usability measured by the objective criteria and procedure set officially.

Many existing sub-characteristics of usability correspond to the evaluation criteria of using [Koh and Jiang, 2017]. For example, effectiveness, efficiency, and their measures of SQuaRE's quality in use are essentially regarding the evaluation of third parties' view while satisfaction is regarding the evaluation of user's view.

Two different definitions can be made for the sub-characteristics according to the two views. For example, Koh and Jiang [2017] redefine effectiveness, efficiency and satisfaction under the user's view as follows:

- **Effectiveness**: The user's (subjective) evaluation on how well his/her goals are achieved.
- **Efficiency**: The user's (subjective) evaluation on how much the cost is expended.
- **Satisfaction**: The user's evaluation on how much he/she is satisfied.

On the other hand, they can be defined as follows under the third parties' view: For a business application product which is supposed to be used to support a predetermined set of tasks in a workplace,

- **Effectiveness**: The accomplishment of the tasks,
- **Efficiency**: The costs to accomplish the tasks,
- **Satisfaction**: The overall evaluation of the sponsor on how the product is satisfactory.

It is noticeable the definition of satisfaction of third parties' view is quite different with that of SQuaRE's quality in use.

A person can be satisfied for various reasons. Satisfaction means, according to Oxford Living Dictionaries, "*fulfillment of one's wishes, expectations, or needs, or the pleasure derived from this.*" The '*fulfillment of one's wishes, expectations, or needs*' is virtually the same as the typical definition of effectiveness. That is, a person is generally satisfied because his/her using experience is effective. One can be satisfied because his/her using experience is efficient too. Or one can be satisfied because his/her using experience is pleasing, comfortable, exciting, or etc. Each of these reasons can be included as a measurement item of satisfaction.

- **Overall measurement**: Ask the user the degree of overall satisfaction: for example, how much satisfactory was your recent using experience?
- **Itemized measurement**: Ask the user the degree of satisfaction for each aspect of satisfaction: for example, how much useful was your recent using experience?

SQuaRE uses both overall measurement and itemized measurement [ISO/IEC 25022 : 2016]. In itemized measurement, effectiveness or efficiency can be included as a measurement item of satisfaction. SQuaRE defines usefulness as the '*degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use*,' which is virtually the same as its definition of effectiveness. In this regard, Koh and Jiang [2017] define satisfaction as the synonym of usability in the user's view. In other words, satisfaction can be regarded as an operational definition of usability in the user's view.

In fact, SQauRE also take this approach. All the measures of usability in SQuaRE's product quality model are regarding user interface, while the factors that influence effectiveness and efficiency are included chiefly in functional suitability and performance efficiency [Koh 2017a]. This implies that SQuaRE thinks that the factors which mainly influence satisfaction, effectiveness, and efficiency are different and that the factors should be classified into separate groups. It will not be desirable to aggregate satisfaction, effectiveness, and efficiency into usability unless the correlation among them is sufficiently high.

Effectiveness and efficiency of Gonzalez Sanchez et al.'s [2009a] playability are virtually the same as the corresponding parts of SQuaRE, although their wording is slightly different [Koh and Jiang, 2017]. Suppose that a person has used a game software product just for fun. Then, it will be meaningless to measure effectiveness if satisfaction is measured too. Moreover, the time to play (that is, efficiency) may not matter at all in this case. So, the following issue arises:

- **Issue**: Which alternative is more useful?
  - **A1**: Let all major evaluation criteria of using included and in usability.
  - **A2**: Restrict the scope of usability to cohesive elements only: For example, define satisfaction as the operational definition of usability, and treat effectiveness and efficiency as distinctive and independent characteristics regarding using.

A1 is the approach that the current major usability models including SQuaRE adopt. However, our hypothesis is that A2 is more desirable for the reasons as follows:

- The inconsistency, vagueness, and confusions associated the current usability models: Koh and his colleagues [Koh 2016, 2017a, 2017b; Koh and Whang 2016; Koh and Jiang 2017] illustrate this point of argument well.
- The correlation among satisfaction, effectiveness, and efficiency may be low. If so, A1 will make the concept usability vague and of little use. This point of argument should be verified by extensive empirical studies.
- Usability may be more easily and systematically specialized for specific types of software if usability is evaluated consistently only in user's view: The third parties are very diverse according to software type by software type. So, combining the heterogeneous viewpoints with the homogeneous user's view will make it too complex to define a universal evaluation process.
- Satisfaction, effectiveness, and efficiency may be used for distinctive analyses and for purposes. For example, efficiency may be used in an economical or financial analysis in which satisfaction and effectiveness are not used. Or, only effectiveness may be used to confirm that the functional requirements are implemented properly.

- **Principle of Parsimony**: The quality characteristic should be defined as brief and precise as possible.
- **Principle of Cohesiveness**: The quality characteristic should consist of elements that fit together well and form a united whole.
- **Principle of Inheritance**: Every aspect of usability should be able to be inherited by its special-type characteristics, possibly, with proper specializations.

The principle of parsimony translates the famous law of parsimony to software quality version. The existing definitions of usability and quality in use are generally too long, complex, and abstract. They violate the principle of parsimony. Software quality is very elusive [Kitchenham and Pfleeger, 1996]. So, it should be defined as

specifically and precisely as possible.

The existing definitions of usability and quality in use including those of SQuaRE contain too many things that do not fit together well and do not form a united whole. It is one of the fundamental reasons of why the existing definitions of usability are so long, complex, and vague. The scope of usability should be sharpened to the extent for ordinary users and software engineers to understand and use easily.

The principle of inheritance is the fundamental principle of classification. Efficiency is excluded in Gonzalez Sanchez et al.'s [2009b] playability model. It is effectiveness, however, what is excluded in fact since effectiveness is defined virtually the same as the typical efficiency in their playability model. This seems very reasonable since effectiveness can be effectively addressed in the framework of satisfaction for game software. The only proper goal of playing is getting fun or being satisfied. Moreover, is it really important to shorten the time taken to play? It seems not. The economic or financial aspect is another question.

As the result, the usability model for general software cannot be specialized for game software for fun or pleasure. The usability model for general software should be redefined to be able to be specialized properly for special types of using.

5. Conclusions

In this paper, the term usability is used to encompass quality in use. The benefit of a model can increases as the complexity of the model increases. The cost associated with using the model, however, almost always increases as the complexity of the model increases. If the complexity increases over some threshold, the marginal cost can exceed the marginal benefit. This paper postulates that the typical current usability models including those of SQuaRE are too big and complex.

In this paper, a model of activities associated with using software products and a software usability model based on the activity model are proposed. The activity model contains two categories of software activity type: special-types of using and component activities of using. They are classified according to the goal of using and can overlap each other. The special-type activity is using by itself. Working, playing, and studying are, but do not exhaust, the examples. The component activities constitute using and each of them is not using. Navigating, preparing data, inputting data, waiting response, examining output data, and utilizing the output data are the examples.

The usability model presented in this paper is based on the activity model. It contains two types of characteristics associated usability: special-type characteristics of usability and sub-characteristics of usability. The special-type characteristics correspond to the special-type activities. Workability, playability, and study-ability are, but do not exhaust, the examples. They can overlap each other. They are usability too by itself. The sub-characteristics correspond to component activities. Navigate-ability, data-prepare-ability, data-input-ability, response-wait-ability, out-examine-ability. and output-utilize-ability

are the examples. They are not usability by itself.

Assessing is the fundamental and indispensable aspect of quality. The activity-oriented usability model contains evaluation criteria of usability. Evaluation criteria are classified further according to whose view they are subject to: those subject to user's view and those subject to third parties' view. Satisfaction, effectiveness and efficiency are the most typical sub-characteristics of usability in existing software quality models. In this paper, satisfaction is classified as the evaluation criterion of user's view while effectiveness and efficiency are classified as the evaluation criterion of third parties' view.

Satisfaction is regarded as an operational definition of usability and included in the activity-oriented usability model. Other existing sub-characteristics of usability including effectiveness and efficiency are not included in the model. As the result, usability can be interpreted as the goodness of using, which is evaluated by the users themselves. Effectiveness and efficiency are the concepts valid for the software product without user-interfaces while usability is valid only for the software product with user-interfaces. So, it is proper to separate effectiveness and efficiency from the user's satisfaction.

Three fundamental principles regarding the software quality model are proposed: Principles of Parsimony, Cohesiveness, and Inheritance. Discussions illustrate well that typical existing usability models violate these basic principles. The usability model should be compact and precise enough for ordinary users and software engineers to understand and use easily.

Usability is the most important but illusive concept regarding software quality. Many authors have tried to define general usability models which can be applied to most kinds of software. The dream of the general and universal usability model, however, may be an illusion. The activity-oriented usability model proposed in this paper can serve as a prototype to be specialized according to software type by type.

Empirical studies on the correlation among satisfaction, effectiveness, and efficiency are required to confirm the validity of the activity-oriented usability model. If the correlation is low, the activity-oriented validity model can be accepted to be more valid and useful than the existing usability models.

Theoretical and empirical studies on how the activity-oriented usability model can be specialized according to the goal of using are also required to confirm the validity and usefulness of the usability model. The general model of usability may be an illusion.

References

[1] Alnanih, R., Ormandjieva, O., and Radhakrishnan, T., "A New Quality-in-Use Model for Mobile User Interfaces", *Proceedings of Joint Conference of the 23rd International Workshop on Software Measurement (IWSM) and the Eighth International Conference on Software Process and Product Measurement (Mensura)*, 2013, pp. 165-170.

[2] Boehm, B. W., Brown, H., and Lipow, M., "Quantitative Evaluation of Software Quality", *TRW Systems and Energy Group*, 1978.

[3] Collins Cobuild Advanced Learner's English Dictionary, reference date: 05/16/2016.

[4] Foraker Labs, "Introduction to User-Centered Design", http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design/, 2002 (reference date: 17/04/2017).

[5] González Sánchez, J. L., Montero Simarro, F., Padilla Zea, N., and Guitierrez Vela, F. L., "Playability as Extension of Quality in Use in Video Games", Proceedings of I-USED'09: The Interplay between Usability Evaluation and Software Development, Proceedings of the Second International Workshop on the Interplay between Usability Evaluation and Software Development, Uppsala, Sweden, August 24, 2009a.

[6] González Sánchez, J. L., Zea, N. P., and Guitierrez, F. L., "Playability: How to Identify the Player Experience in a Video Game", Proceedings of IFIP Conference on Human-Computer Interaction: Human-Computer Interaction-INTERACT 2009, 2009b, pp. 356-359.

[7] Grady, R. B., *Practical Software Metrics for Project Management and Process Improvement*, Prentice-Hall, 1992.

[8] Herrera, M., Moraga, M. A., Caballero, I., and Calero, C., "Quality in Use Model for Web Portals (QiUWeP)", In: Daniel F., Facca F.M. (eds) *Current Trends in Web Engineering. ICWE 2010, Lecture Notes in Computer Science*, Springer: Berlin, Heidelberg, Vol. 6385, 2010, pp. 91-101.

[9] ISO/IEC 25010 : 2011, *Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models*, ISO, 2011.

[10] ISO/IEC 9126-1 : 2001, *Software Engineering-Product Quality-Part I: Quality Model*, ISO, 2001.

[11] ISO/IEC 9241-11 : 1998, *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)-Part 11: Guidance on Usability*, ISO, 1998.

[12] ISO/IEC 25021 : 2012, *Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Quality Measure Elements*, ISO, 2012.

[13] ISO/IEC 25022 : 2016, *Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Measurement of Quality in Use*, ISO, 2016.

[14] ISO/IEC 25023 : 2016, *Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Measurement of system and Software Product Quality*, ISO, 2016.

[15] Kitchenham, B. and Pfleeger, S. L., "Software Quality: The Elusive Target [special issue section]", *Software*, IEEE, Vol. 13, 1996, pp. 12-21.

[16] Koh, S. and Jiang, J., "What should Using a Software Product and Usability of the Software product be?", *Journal of Information Technology Applications & Management*, Vol. 24, No. 3, 2016, pp. 73-92.

[17] Koh, S. and Whang, J., "A Critical Review on ISO/IEC 25000 SQuaRE Model", *Proceedings of the 15th International Conference on IT Applications and Management: Mo-*

bility, Culture and Tourism in the Digitalized *World*, (ITAM15), 2016, pp. 42-52.

[18] Koh, S., "Cause-and-Effect Perspective on Software Quality: Application to ISO/IEC 25000 Series SQuaRE's Product Quality Model", *Journal of Information Technology Applications & Management*, Vol. 23, No. 3, 2016, pp. 71-86.

[19] Koh, S., "The Checklist for System and Software Product Quality Implied in the Product Quality Model of ISO/IEC 25000 Series SQuaRE", Proceedings of 17th International Conference on IT Applications and Management: Babolsar, Iran, 22-23 February 2017, pp. 126-136.

[20] Koh, S., "The Principle of One Quality View and Division of Product Quality Model of ISO/IEC 25000 Series SQuaRE", *Asian Journal of Information and Communications*, Vol. 9, No. 1, 2017b, pp. 87-101.

[21] McCall, J. A., Richards, P. K., and Walters, G. F., *Factors in Software Quality*, Vol. I, II, and III, US Rome Air Development Center Reports, US Department of Commerce, USA, 1977.

[22] Microsoft Corporation, "Usability in Software Design", https://msdn.microsoft.com/en-us/library/ms997577.aspx, October 2000 (reference date: 17/04/2017).

[23] Nielsen, J., "Usability 101: Introduction to Usability", Nielsen Norman Group, https://www.nngroup.com/articles/usability-101-introduction-to-usability/, Jan. 4, 2012 (reference data 17/04/2017).

[24] Oxford Learner's Dictionary (reference date: 05/16/2016).

[25] Shuja, A. K. and Krebs, J., *IBM Rational Unified Process Reference and Certification Guide: Solution Designer*, IBM Corporation, 2008.

[26] U.S. Department of Health and Humanity Services, "Usability Evaluation Basics", https://www.usability.gov/what-and-why/usability-evaluation.html, reference date: 17.04/ 2017.

Author Profile

Seokha Koh

Seokha Koh is the professor of the Department of MIS, Chungbuk National University. His current primary research areas include Software Quality Management, Business Process Modeling, Software Architecture, Project Management, and Software Engineering.

You-Jeong Koh

You-Jeong Koh is the UX specialist who works in Reflexion Corporation.