# Measuring Usability of Requirements Using Social Network Service in Product Line Engineering

Sang-eun Park, Neunghoe Kim, Jung-been Lee, Geunhyung Lee and Hoh Peter In

Department of Computer Science, Korea University Seoul, South Korea

{jppowers, neunghoi, jungbini, skypepper, hoh\_in}@korea.ac.kr

### Abstract

Consideration of the quality of software requirements is a critical activity that is directly connected to the success of a software project. In the product line engineering domain, software requirements are a fundamental part of all the core assets that can be reused in other products. In order to reuse core assets in a family product, an activity that determines quality factors with respect to the software requirements is more important in this domain than in other domains. Although there are many researches on the identification of commonality and variability issues, the consideration of quality factors for satisfying the determined software requirements has not been actively studied. In this paper, we propose a systematic method that evaluates usability in terms of user opinions published on a social network service during the development of core assets in the field of domain engineering.

Keywords: Usability, Product Line Engineering, Social Network Service

# **1. Introduction**

Consideration of the quality of software requirements is a critical activity that is directly connected to the success of a software project. Even if the functional software requirements have been implemented, users' decisions change considerably depending on the quality of functions. In particular, in the product line engineering (PLE) domain, it is important to develop core assets that consider not only the functional but also the quality aspects of the software requirements. Moreover, in order to reuse these core assets in other products, these assets have to be developed through a reasonable level of decision. Therefore, a method that determines quality factors with respect to the software requirements is more important in this domain than in other domains.

Although there are many researches on the identification of commonality and variability issues [1, 2], the consideration of quality factors for satisfying the determined software requirements has not been actively studied. According to quality standard documents [3], there are six main software quality attributes: functionality, reliability, usability, efficiency, maintainability, and portability. Functionality is considered while determining the software requirements, and other attributes except for usability are determined on the basis of the internal company decisions. However, usability is an attribute that requires user opinions, and these opinions may change from time to time. Therefore, usability is considered along with the software requirements of a system that is already in use while developing core assets in the field of domain engineering. After the activation of social network services, research has been

conducted for analyzing social data; such research reveals the possibility of a new method of using social data for ensuring the usability of a product [10].

In this paper, we propose a systematic method that evaluates usability in terms of user opinions published on a social network service during the development of core assets in the field of domain engineering. This method improves the usability of the software requirements by the use of a social network service in the field of product line engineering. Further, it analyzes the user opinions published on the abovementioned service and supports the identification of commonality and variability through a quantitative evaluation of usability. Moreover, it helps to reach a reasonable decision for checking the consistency of the existing software requirements.

The rest of this article is organized as follows: Section 2 describes the background of product line engineering and social network service, and Section 3 introduces the proposed method for the measurement of the usability of requirements by using a social network service in detail. Finally, Section 4 presents the conclusions and future work plans.

# 2. Background

# 2.1. Product Line Engineering (PLE)

Product line engineering (PLE) is defined as a set of software intensive systems built from a common set of core assets. PLE is based on core assets such as the architecture, reusable software components, requirements, and documentation. Specific activities, such as core asset development and product development with the core assets under technical and organizational management, are essential for PLE [4].

PLE has the separation of processes between the domain and the application engineering processes proposed in [5]. The domain engineering process produces a platform that includes the commonality of applications and the variability to support mass customization.

This platform classifies the different kinds of development artifacts as domain artifacts and applications artifacts. Domain artifacts are reusable development artifacts created in the sub-processes of domain engineering. Application artifacts are the development artifacts of specific product line applications.

#### 2.2. Social Network Service (SNS)

A social network service (SNS) can be described as an online service or platform that reflects the relation of social human networks. The advent of ubiquitous environments and the development of the Internet have led to the development and widespread use of portable mobile devices (e.g., smartphones) [6]. This phenomenon has led to the creation of SNSs using which people can easily share their opinions and interests with each other.

Nowadays, the analysis of a social network is a complex domain because of the amount of stored data involved. SNSs have evolved into an appropriate medium for collecting various individual opinions based on the data from the participants. Hence, there have been many researches on various related areas [7]. A recent research showed that users posted product reviews on SNSs at least once a month, and one-fourth of these users responded to these reviews [8]. SNS users express their judgment, experience, and satisfaction/dissatisfaction without inhibitions.

# **3.** Measuring Usability of Requirements Using Social Network Service in PLE

This section presents a step-by-step description of the proposed method. The proposed method is an improved version of an existing method that manages domain requirements [9]. After deriving domain requirements by an existing method, we developed the proposed method that measures usability using SNS. Further, software requirements are generalized as a result by the analysis of usability with the existing method.



# Figure 1. Measuring Usability of Requirements Using Social Network Service

The overall architecture of the proposed method is depicted in Figure 1. According to this figure, the proposed method consists of five activities: usability keyword elicitation, SNS documents collection and refinement, usability measurement, context matrix analysis, and expert decision.

# Step 1: Usability Keyword Elicitation

This step determines the keywords related to usability in order to obtain usabilityrelated information from SNS. The input of this step is the result of the determination of domain requirements, which is the output of an earlier activity [9]. The output of this step is one or two keywords; these keywords are nouns extracted from the domain requirements and are used in the next step for gathering usability information.

#### Step 2: SNS Documents Collection & Refinement

This step collects and refines the documents on a social network by using the usability keyword list from Step 1. First, the required time period is selected on the basis of a standard including the collection starting date. The selected duration has to contain the latest information. Before the collection, a dictionary of words has to be built; this dictionary should include both positive and negative meanings of the listed words. Then, sentences that contain the keywords (both positive and negative connotations) are collected using the usability keywords. Next, these collected

sentences are reviewed to filter out unnecessary or incorrect information. Such a review leads to high measurement accuracy and a collection of required/relevant sentences.

### **Step 3: Usability Measurement**

This step measures the value of the user opinions on each requirement using the refined and filtered documents from Step 2. A positive opinion implies that the requirement is well-defined before this step, and a negative opinion implies that the requirement has certain issues with respect to usability. Usability is measured by quantitatively measuring the frequency of both negative and positive opinions. The following is the formula used for calculating usability:

# **Usability** = Number of positive opinions – Number of negative opinions

#### Ratio of usability (Negative opinions)

= Number of negative opinions/Total number of opinions that were collected \* 100

A positive value of usability implies that the requirements have surpassed the baseline with respect to usability. In contrast, a negative value of usability implies that the requirements are below the baseline with respect to usability. Further, the calculated value is added to the context matrix.

#### **Step 4: Context Matrix Analysis**

Usability/ N Ratio	System Reg	Property/ Ratio	System n1	System n2	System n3
2980(0.006%)	PR1	C(100%)	o	o	o
1764(1.02%)	PR2	C(100%)	0	o	o
3450(14.25%)	PR3	PV(67%)	0	x	0
1350(31.452%)	PR3i1		0	x	x
3450(14.25%)	PR3i2		0	x	o
4285(9.215%)	PR4	CV(100%)	o	o	o
1816(2.374%)	PR4i1		x	o	0
2431(16.45%)	PR4i2		o	x	o
1125(6.579%)	PR4i3		0	x	x
-658(52.784%)	PR5	P(67%)	0	x	0

Table 1. Example of Context Matrix

This step analyzes the commonality and variability of each domain requirement by using a context matrix. The calculated usability values are added to the existing context matrix. An example of the context matrix is given in Table 1. The existing method calculates property/ratio and determines commonality and variability by a comparison with other systems. However, usability that is connected directly to user opinions cannot be ignored in the process of developing core assets that can be reused. It is not necessary to select low-usability requirements as the core assets. Therefore, with the existing method identifying property/ratio, the proposed method determines commonality and variability considering usability and generalizes requirements if usability has a specific value over the baseline. However, if the domain requirements have a usability value that is less than the baseline value, the execution moves to the expert decision step. This baseline value is determined by a domain expert because each domain or project has different characteristics.

#### **Step 5: Domain Expert Decision**

In this step, domain experts analyze the classified domain requirements that have low usability and select low-usability requirements such as the log-in and log-out functionality. Because of the characteristics of the SNS, specific requirements that are natural can be collected as a low-usability requirement. Such requirements are skipped upon an expert review. In the case of low-usability requirements that constitute a variability, whether the requirements need to be modified or excluded from the list of core assets depends on the budget and time. Further, in the case of low-usability requirements that constitute a commonality, whether the requirements that need to be modified are determined and how to increase usability is studied.

# 4. Conclusions and Future Plans

The software product line domain requires reasonable judgment because of its aim to reuse core assets. In this paper, we proposed a systematic method that evaluates usability in terms of user opinions by using a social network service. This method supports reasonable judgment by reflecting the users' satisfaction/dissatisfaction during the development of the core assets. There are future research plans to apply the proposed method in smart-device domain and verify its effectiveness.

#### Acknowledgements

This research was supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2012M3C4A7033345), and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A2009021).

# References

- [1] M. Moon, K. Yeom and H. Chae, "An Approach to Developing Domain Requirements as a Core Asset Based on Commonality and Variability Analysis in a Product Line", IEEE Transactions on Software Engineering, vol. 31, no. 7, (2005).
- [2] S. Lee, J. Kim, C. Song and D. Baik, "An Approach to Analyzing Commonality and Variability of Features using Ontology in a Software Product Line Engineering", Proceeding of the Fifth International Conference on Software Engineering Research Management and Applications, (2007), pp. 727-734.
- [3] ISO/IEC JTC 1/SC 7/WG 6. ISO/IEC 9126-1:2001, Software Engineering Product quality Part1: Quality Model, (2007).
- [4] L. M. Northrop, "SEI's Software Product Line Tenets", IEEE Software, vol. 19, no. 4, (**2002**) July/August, pp. 32-40.
- [5] Pohl, Klaus, Böckle, Günter, Linden, Frank J, "Software Product Line Engineering", Springer (2005).
- [6] DMC Media, "SNS Users Around the World", http://www.dmcreport.co.kr, (2010).
- [7] O. Kwon and Y. Wen, "An Empirical Study of the Factors Affecting Social Network Service Use", Journal of Computers in Human Behavior, vol. 26, no. 2, (**2010**), pp. 254-263.
- [8] G. Colony, "Social Sigma Rising", http://blogs.forrester.com, (2010).
- [9] D. Park, D. Kim and K. Jung, "A Method to Manage Requirements Analyzing the Commonality and Variability in Product Line", Korea Information Processing Society, vol. 13-D, no. 7, (**2006**), pp. 909-922.
- [10] J. Hart, R. Charlene, T. Faisal, S. Corina and D. Alan, "Exploring the Facebook Experience: A New Approach to Usability", Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges, ACM, (2008).

# Authors



# Sang-eun Park

He is a M.S. candidate with the College of Information and Communications, Korea University. His research interests include requirements engineering, software engineering. He received the B.S. degree in Computer Science from Sangmyung University, in 2011.



# Neunghoe Kim

He is a Ph.D. candidate with the College of Information and Communications, Korea University. His research interests include requirements engineering, value-based software engineering, software engineering economics, and embedded software engineering. He received the M.S. degree in Computer Science from Korea University.



# Jung-been Lee

He is a Ph.D. Course in the Department of Computer Science and Engineering at Korea University in Seoul, Korea. His major areas of study are self-adaptive software, software architecture evaluation and potential defect analysis. He received the M.S. degrees in Computer Science and Engineering from Korea University in 2011.



# **Geunhyung Lee**

He is a M.S. candidate with the College of Information and Communications, Korea University. His research interests are network, and ubiquitous computing. He received the B.S. degree in computer science from Sangmyung University, in 2011.



# **Hoh Peter In**

He is the Vice Dean of the College of Information and Communication at Korea University in Seoul, South Korea. His primary research interests are embedded software engineering, social media platform and service, and software security management. He earned the most influential paper award for 10 years in ICRE 2006. He has published over 100 research papers. He was an Assistant Professor at Texas A&M University. He received his Ph.D. in Computer Science from the University of Southern California (USC).