

Article

# Value-Oriented Requirements: Eliciting Domain Requirements from Social Network Services to Evolve Software Product Lines

Nazakat Ali  and Jang-Eui Hong \* 

Department of Computer Science, Chungbuk National University, Cheongju 28644, Korea;  
nazakatali@selab.cbnu.ac.kr

\* Correspondence: jehong@chungbuk.ac.kr; Tel.: +82-43-261-2261

Received: 14 August 2019; Accepted: 18 September 2019; Published: 20 September 2019



**Featured Application:** The proposed approach can be better used in the software requirements engineering field, more specifically in software product line Engineering. The proposed approach can elicit user requirements from social network sites and has the ability to identify variable and common requirements.

**Abstract:** Social network services allow a large population of end-users of software products to publicly share their concerns and experiences about software systems. From a software engineering perspective, such data can be collected and analyzed to help software development organizations to infer users' emerging demands, receive their feedback, and plan the rapid evolution of software product lines. For the evolution of software product lines, organizations supplement emerging requirements in their products to meet user's needs and also to retain their dominance in the market. Therefore, social network services, being a communication channel, have supported a number of software development activities such as requirements engineering. It has supported software development organizations to cope with numerous limitations of the traditional requirements engineering approaches by eliciting, prioritizing, and negotiating user requirements. However, these approaches do not consider eliciting requirements in terms of variability and commonality while identifying requirements. To address this issue, we have proposed a social network service-based requirement engineering process. It considers the attributes of users' opinions to determine variability and commonality. In order to justify our proposed approach, a controlled experiment was conducted on a sample set of end-users on Facebook and Twitter. The experimental results show that the team using the proposed approach performed better in terms of efficiency and effectiveness than the team that used a traditional requirements engineering approach.

**Keywords:** social networks; requirement engineering; variability and commonality; software product lines; requirements elicitation; text analytics

## 1. Introduction

The World is generic and the user demands are rapidly changing as time goes by [1,2]. The fundamental factor in the success of a software product line (SPL) is to understand what users really want [1,2]. SPL is a paradigm where applications are derived from core assets which are built in domain engineering [3].

In order to fulfil user requirements, it is necessary for SPL organizations to know the opinions on the products that they have already floated on the market. The opinion may tell about user satisfaction which measures how software product, provided service, and overall user experience

either falls short of or meets user expectations. Therefore, it is critical to monitor users' feedback in order to know what users really want. Otherwise, it will remain a dream to be ahead in a competitive software market and the privileges of organization's yesterday could be superseded by tomorrow's growing challenges. Social Network Services (SNS) are applications that connect users to each other by creating personal information profiles. Personal information may include photos, video, audio, location, likes, dislikes and area of interest [4]. In SNS, individuals are related (directly or indirectly) to each other based on common interest, friendship, or trust, etc. [5]. Using SNS, billions of users do their activities by building relationships, posting their opinions, or talking about their particular experiences. Recently, SNS have gained substantial attention from academia and industrial researchers. It has unlocked innovative ways of research in various fields like marketing, economics, arts, and information technology (IT) [6]. SNS users are growing exponentially [7] which provides highly diverse data that has brought revolution in many areas of data science. From the software engineering point of view, SNS have laid the foundation of unprecedented opportunities for software organizations to get instant feedback from a large number of their end-users [8]. Such feedback can be used to elicit new requirements and to plan the rapid evolution of software in the next release. SNS have been used in various phases of the software development lifecycle, such as project planning, Requirements Engineering (RE), and implementation of software projects [9]. In RE, SNS have demonstrated their ability in supporting and improving several process activities, such as requirements gathering [7,9], negotiation [10], stakeholder identification and prioritization [11,12] and many others.

Several studies have used SNS to elicit user requirements from SNS. These [13–15] are representative examples where SNS are utilized to conduct RE activities. Guzman et al. [13] used Twitter to automatically identify, group, and rank tweets that are potentially relevant to software evolution. The tweets were classified into improvement request and others. Williams et al. [14] have proposed an approach that extracts tweets from Twitter to elicit user requirements. The authors have classified tweets into bug reports, user requirements and others. They have extracted 949 user requirements and 1061 bug reports for 10 selected software products. Pagano et al. [15] have presented an empirical study which extracts user feedback from app stores. The authors have examined the usage of feedback features, the content of feedback, and its impact on the end-users. Mughal et al. [16] proposed an SNS-based process for minimizing the in-group bias as well as identifying stakeholders and eliciting user requirements. However, these research articles do not consider requirements engineering for SPL. The RE process for SPL should be in line with the aim of SPL. SPLs are designed aiming for short time-to-market, quality, and rapid development. Therefore, to successfully cope with short time-to-market time, the RE process used should be fast, easy and inexpensive.

In this paper, we propose an SNS-based RE process to conduct RE activities. Our proposed approach collects user tweets and comments from Twitter and Facebook respectively. Some natural language processing (NLP) techniques such as cleaning [17], parts-of-speech (POS) tagging [18], WordNet lemmatization [19], and N-grams [20] are applied to get potential candidate terms that can be interpreted as user requirements in later stages. The requirements are then analyzed to get variable requirements (VR) and common requirements (CR) from the initial set of requirements.

Generally, we can collect user demands for a specific theme from various social networks. However, measuring values in software engineering is complex but also essential for the systematic study of values in software engineering [21]. Therefore, we also focused on other information of SNS users which we called personal attributes, for instance, 😊 (emoticon), country, gender, and language, to draw more vigorous conclusions. The requirements which are elicited by considering personal attributes are called value-oriented requirements in our study. The definition of value-oriented requirements is derived from the study of Thew et al. [22]. The authors mentioned that 'socio-political' issues such as values, people's feelings and emotions are often considered as problems in the RE process. The authors presented a value-based RE method that first introduced a taxonomy to deal with socio-political issues in RE to complement prevailing analysis of non-functional requirements. The authors introduced new considerations into the RE process by paying attention to individual users' values, motivations

and emotions. We selected 'gender' and 'country' attributes due to the fact that different regions (countries) can have gender-specific values or values in general that can be useful to better understand users' demands. However, the study of Thew et al. [22] addressed value-based RE, meaning that the engineering process considers values to perform development activities. Our value-oriented RE approach identifies users' needs as one of the requirements, not process activities.

Mostly, this research article contributes in three-fold. First, we present an SNS-based RE process to conduct RE activities for SPL. Our approach focusses on VR and CR which is not addressed in related work. Second, we propose a methodology to determine VR and CR considering the influencing factors of tweets or comments. Third, the validity of this proposed approach is measured by conjoining with Facebook and Twitter, which allows users to express their opinion without any temporal and spatial constraints. Furthermore, a controlled experiment consisting of two teams each consisting of 15 members was performed to test the proposed approach and its applicability compared to traditional RE process. The smartwatch domain was selected as a subject in our controlled experiment to exercise our proposed approach.

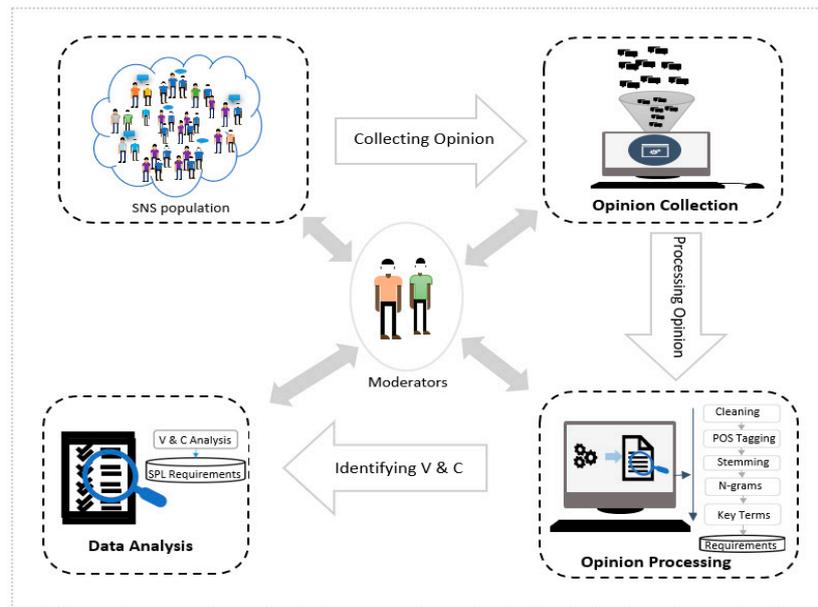
We envision the relevance of this study within new software paradigms like cloud computing, mobile applications, and software ecosystems. In software ecosystems, anonymous end-users are not within instant reach of software development organizations and support provided by conventional RE approaches is insufficient. Therefore, smart wearables developers, cloud service developers, game developers or "Internet of XYZ" ecosystems could use our RE approach to elicit user requirements.

This paper is organized as follows: Section 2 discuss the SNS-based RE process for SPL while in Section 3 controlled experiment design and results are described. In Section 4, we analyze the outcomes from both RE approaches. In Section 5 we discuss threats to the validity of this study while Section 6 describes related work and finally, Section 7 concludes this article with some future research directions.

## 2. Proposed Approach

Social networks are frequently used in the software industry, especially to involve users during market surveys to ensure the usefulness of requirements that they have developed. SNS are also useful because social media encourages discussions on specific issues, and it is accessible to everyone to participate wherever and whenever they want [23]. Additionally, the SNS have the capability to promote a collaborative and cooperative environment to get new ideas [24].

We propose an approach to elicit user requirements from popular SNS such as Facebook and Twitter. The aim of this proposed approach is getting instant feedback of users in order to evolve software systems quickly. In our approach, we first extract the most discussed keywords from SNS for a related domain and then make a domain-specific query(s) to start a discussion on the selected SNS. User opinions are collected and analyzed to know their current demands. By applying some NLP techniques (WordNet lemmatization, POS tagging, N-gram) opinions are processed and candidate terms are extracted and interpreted as requirements. The key terms are a major source to elicit user requirements [25]. However, extracting key terms from an unstructured corpus is a challenging task. The extracted requirements are then analyzed to decide about VR and CR. Our SNS-based approach is comprised of three steps, which are, opinion collection, opinion processing, and data analysis phase. In the opinion collection step, we collect users' opinions from SNS users. Figure 1 shows an overview of our approach.



**Figure 1.** SNS-based RE Process. (SNS: Social Network Services, RE: Requirements Engineering).

In this section, the proposed approach is described generally while Section 3 explains our proposed approach in detail with a controlled experiment.

2.1. *Opinion Collection*

In this phase, the domain keywords which are mostly discussed on SNS are identified. Through keyword frequency, the extent of users’ need is understood. We have selected keywords with the highest frequency for further processing. Table 1 shows an example of keywords extracted from Facebook.

**Table 1.** Example of keywords extraction from Facebook.

Keywords	Smartwatch Apps	Smartwatch Price	Smartwatch Design
Frequency	1900	189	2030

The keywords were extracted from a specific Facebook page (smartwatch iwatch) and Twitter by using hashtags (#smartwatch design, #smartwatch app, etc.). In this phase, we have also investigated SNS to know which of their features would allow us to extract customer requirements. The investigation results are shown in Table 2.

**Table 2.** Offered features of selected SNS.

Requirements	Facebook	Twitter	Instagram
Easy access	✓	✓	✓
Post queries	✓	✓	X
Comment on—queries	✓	✓	X
User need—approval	✓	X	X

Based on our investigation results (Table 2), we selected Facebook and Twitter. Both Facebook and Twitter are reachable for all users. Users can easily post their opinions and also can get feedback. While Instagram is easy to access, it is a platform to post photos and videos for pre-approved followers, which is why we dropped it.

The requirements engineers join selected communities and become members because users with a common interest can join the communities without any hurdle. After that, requirements engineers

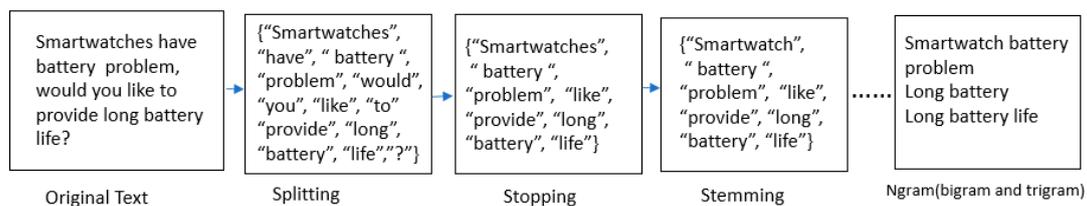
post their queries or questions on selected SNS. The followers of the designated communities start exchanging their opinions and start a discussion. For Facebook, users need approval to join the communities and their group of interests if needed. SNS enables the requirements engineers to get the opinions of users with diverse cultures, ethnicities, and occupational backgrounds.

After identification of domain keywords, query selection, and SNS selection, the opinions of users from the targeted SNS are collected. Two Facebook pages (Smartwatch iwatch, Wearable-iwatch) were used to collect user feedback. For Twitter, SmartWatch and NextGeneration wearables were used. There are various kinds of online communication channels and the ways of collecting data differ from channel to channel. For example, SNS such as Facebook and Twitter offer open application programming interface (API) for accessing and extracting corpus. Furthermore, the data extracting methods can be different from one SNS to another SNS. For example, possible methods like crawling and clipping can also be used to extract data from web-based applications. We only used open APIs for Facebook and Twitter, respectively [26,27]. Before the collection of data (opinions), we took the consent of users to avoid privacy issues.

As one of the valuable requirements, we also collect information such as nationality (geographical information), emoticon, and user device information, and gender. The information of nationality is used for knowing country-specific requirements, and the emotions were also useful to determine user preference for specific functionality of a product. The device information can also be collected to optimize software development.

## 2.2. Opinion Processing

The corpus that is extracted from SNS in Section 2.1 is used as input for the data processing phase. For data cleaning, special characters, punctuations, and numbers are filtered out from the corpus using natural language processing (NLP). Secondly, WordNet lemmatization was utilized to group together different forms of words to make them single items. For example, words such as “show, showing and shown” are considered as the basic word “show”. The WordNet lemmatization helps to reduce the amount of extracted corpus. Thirdly, we have used part-of-speech tagging (POST) from the NLTK (Natural Language Toolkit), and lastly, the N-gram technique [28] is used to extract the terms from the corpus. We are interested in extracting the phrases which might be candidates for requirements. Therefore, we used bigrams and trigrams to extract the combination of nouns, adjectives, and verbs. Figure 2 shows opinion filtering by NLP. We filtered the corpus by applying NLP techniques and manually assigned the labels (attributes) to filtered opinions.



**Figure 2.** The demonstration of Natural Language Processing (NLP) steps to get candidate terms for requirements.

Assigning attributes helps to refine the opinions as well as the objectivity and traceability of user requirements [29]. The attributes are divided into three classes: Redundant, Positive, and Irrelevant. Each attribute class is named after its connotation. The gathered corpus can be ‘Positive’, or ‘Redundant’. Beyond these two classes (Positive and Redundant), the corpus falls in the category of ‘Irrelevant’. ‘Positive’ attribute refers to those requirements which are current demands of users or the opinions which tell about the features offered by parallel competitors. ‘Redundant’ attributes figure out those features which our SPL system offers already, while ‘Irrelevant’ attribute tells us about those comments which are meaningless for us and beyond our consideration.

Examples of initial opinion documentation with attributes and region of the users are shown in Table 3. Through the discussion, we have noticed that users sometimes use a variety of words, synonyms, abbreviations, or use various terminologies for the same things. In order to accurately understand end-user needs, the opinions which are expressed in different ways need to be restored. For example, opinions like, “BTW, the problem with the design of smartwatches are unattractive logos. Why companies don’t gafi” need to be restored to the original text to extract the requirements from it. We infer and rewrite these opinions in an understandable way to transform them into user requirements, as shown in Figure 3.

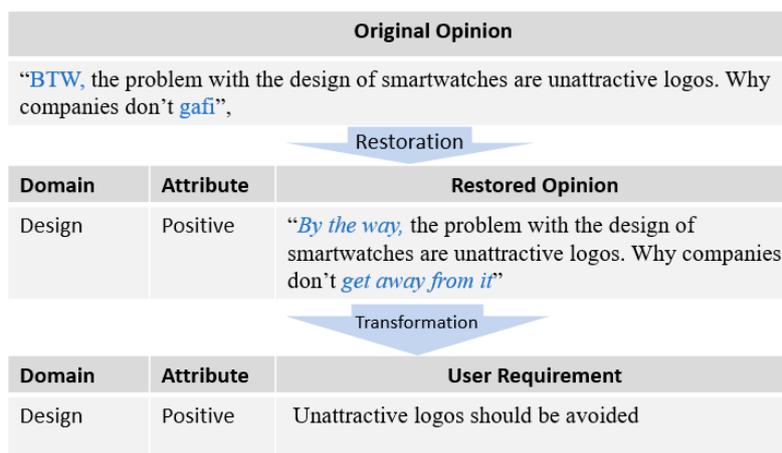


Figure 3. Example of restoration and transformation of collected opinions.

This transformation is carried out using two steps:

Restoration – The SNS users use jargons and abbreviations to express their opinions, this makes it difficult to understand user opinion, and therefore, it is important to restore these jargons and abbreviations and form correct sentences. In Figure 3, abbreviated opinions are corrected by following this step.

Transformation—Once the opinions are restored and form grammatically correct phrases, then attributes are assigned and transformed into standard requirement form, as shown in Figure 3.

Syntactical complexity makes this opinion analysis difficult to understand. However, we noticed that most users’ opinions on an SNS were generally not syntactically complicated. The aforementioned NLP techniques are utilized to remove the garbage. At this stage, the opinions are now converted into requirements and we present them in such a way that becomes more understandable.

For identifying VR and CR, the frequency of requirements is also documented. Table 4 (in data analysis) shows an example of requirements that we presented in an understandable way.

This phase is a semi-automated phase where candidate terms for requirements are identified automatically from the corpus and then these terms are converted into initial requirements manually.

### 2.3. Data Analysis

In this section, requirements are analyzed to determine VR and CR. The SPL engineers assure that requirements must be quantifiable and detailed. The success of any SPL depends on the correct identification of domain requirements in the RE phase [30].

The attributes of tweets and comments such as number of likes, number of replies (discussion) for a comment or a tweet and frequency (duplication) were selected to determine the VR and CR. These factors are called influence factors in our study. The numbers of likes, replies, and frequency or duplication are selected as influence factors to determine the VR and CR. A higher influenced tweet or

comment can tell that the majority of people are taking interest in it. Therefore, we propose following Equation (2) to determine the VR and CR of requirements.

$$I_{op} = (W_{Likes} * N_{Likes}) + (W_{Duplicates} * N_{Duplicates}) + (W_{Replies} * N_{Replies}) \tag{1}$$

$$I_{op} = \sum_{k=1}^3 W_k * N_k (op) \tag{2}$$

where,  $W_k$  is the weighted value for factor of influence, i.e.,  $W_1$  is the weight for Likes,  $W_2$  is the weight for Duplication or similarity and  $W_3$  is the weight for Replies.

**Table 3.** Examples of initial opinion documentation.

Sub-Domain	ID	Opinions	Region	Attributes	Gender
Apps	1	So, a watch that tells you the time when you touch a certain button would not be cool? You could talk text to the person or send voice messages back and forth.	South Korea	Positive	Male
	2	The watch costs \$300 and is currently available in English and Korean. I suggest to develop a watch with Spanish, Arabic, French, German, and Italian capabilities in the near future.	Japan	Positive	Male
	3	My friend died due to blood pressure, we would save his life if there would be blood pressure measuring app. and informed him timely	Pakistan	Redundant (already existing requirement)	Male
	4	Love this! My son is blind, and sadly most of these cool gadgets are ridiculously expensive. Would you like to include voice functionality in it?	USA	Positive	Female
Design	5	Oh! this will be a great birthday present for my friend	Thailand	Irrelevant	Female
	6	this is a brilliant idea	China	Irrelevant	Male
	7	I want to swim with my watch. Water resistant watch is necessity of the day	Germany	Positive	Male
	8	I want to wear smartwatch with every cloth. I don't like green as I am Chinese lolzz	China	Positive	Female

The  $N_k (op)$  represents the numbers of Likes, Duplicates and Replies that a tweet or a comment receives in discussion. The values for Twitter attributes (such as like, retweet, followers and friends, content category, duplicates and sentiment) are determined by a comprehensive survey in Reference [22]. The authors have examined the aforementioned six tweet attributes and rated their importance by a comprehensive survey. In the authors' survey, 84 participants were involved and 76% of the total participants were software developers, 15% were project managers, 4% were product owners and 15% were software engineers. Of the total participants, 38% reportedly had 6–10 years of experience, 30% had 3–5 years of experience, 12% had 1–2 years of experience and 10% had 11–15 years of experience. Furthermore, 66% of the total participants were from industry, 19% were from institutions, 14% were from both industry and research and 1% performed it as a leisure activity. The authors reported that 39% of the total participants were considering tweet attributes while prioritizing the tweets, 44% of the total participants were reportedly sometimes considering tweet attributes and only 17% did not consider tweet attributes while analyzing tweets. This result demonstrates that software engineering researchers and practitioners consider tweet attributes significantly while analyzing tweets.

Therefore, based on Reference [31], the weight for each factor in  $W_k$  is determined by consulting with four our domain experts. The domain experts argued that the attribute Like is very important to determine the importance of a tweet or a comment because the Like hit shows users' interest in the expressed topic. Therefore, the weight for Likes is agreed to be 0.5. The domain experts also argued that Replies attribute of a tweet or a comment can also be important because it tells about

the brainstorming or a discussion about a topic expressed by users. Therefore, weight for Replies is agreed to be 0.4. The domain experts also argued that Duplication or similarity attribute of a comment or a tweet is very critical in determining variability and commonality of a requirement. The more duplication of a comment or a tweet means that more users demand one requirement which can be a common requirement. The level of similarity or duplication can be different for different opinions. Therefore, the weights for Duplication or similarity is decided by calculating using the Jaccard similarity coefficient [32] to extract VR and CR. Equation (3) is used to calculate the Jaccard coefficient.

$$Jaccard(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|} \quad 1 \leq i \leq j \leq x \quad (3)$$

where,  $x$  is the number of comments or tweets. Comments or tweets  $C_i$  is considered as a similar comment or tweet of  $C_j$  if  $Jaccard(C_i, C_j) \geq \alpha$  where,  $\alpha$  is a well-defined threshold. We used  $\alpha = 0.50$  to decide whether it is a duplicated requirement or not. The terms which got  $\alpha = 0.50$  or above were enlisted as duplicate requirements. For instance, we calculate the VR and CR for the following requirement: "The smartwatch should be **water resistant**" expressed in a comment as shown in Figure 4.

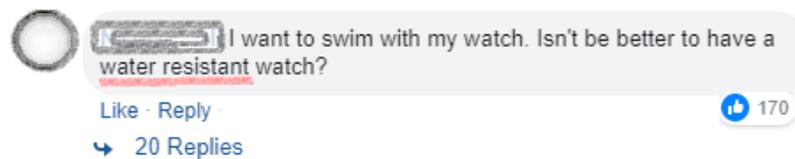


Figure 4. User comment demanding water-resistant smartwatch.

The VR and CR of this requirement is determined by applying Equation (1).

$$\begin{aligned} I_{op} &= (0.5 * 170) + (0.4 * 20) + (0.6 * 24) \\ &= (85) + (8) + (14.4) \\ &= 107.4 \end{aligned}$$

where, 0.5 is the determined weight for Like, 0.4 for Reply influential factor while 0.6 is the similarity value which is determined by applying Equation (2). The  $I_{op}$  is calculated by multiplying the weight of each influencing factor with its corresponding number, as shown in the above example. The above example requirement has 170 likes, 20 replies and 24 duplicated requirements by users. Therefore, the value of  $I_{op}$  is 107.4 which is quite a high value. Based on this value, we classify the above requirement as a common requirement. Based on our continuous experimentation with 89 requirements and discussion with domain experts we have defined a threshold for the  $I_{op}$  value to decide about VR and CR. The threshold value was decided to be 75, meaning that if the value of  $I_{op}$  would be less than the threshold value, then those requirements would be classified as variable requirements, otherwise they would be a common requirement. Table 4 shows an example of variable and common requirements.

**Table 4.** Example of elicited Value-Oriented Requirements.

Sub-Domain	Comment Id	Opinions	I <sub>op</sub>	Com/Variability
Apps	R1	Smartwatch should provide <b>multilingual</b> interface.	72	Variable
	R2	Smartwatch should provide early <b>warning app</b> e.g., earthquake and tsunami etc.	62	Variable
	R3	The application should support <b>blind people</b> to hear time and receive text messages in voice.	71	Variable
	R4	Smartwatch should provide application to monitor <b>heart rate</b> .	68	Variable
	R5	Smartwatch should provide sleep monitor app for users.	123	Common
Design	R6	<b>Solar technology</b> should be used to charge battery of smartwatch.	156.2	Common
	R7	The smartwatch should be <b>water-resistant</b> .	107.4	Common
	R8	The smartwatch should provide at least 18-h <b>battery life</b> .	189	Common
	R9	<b>Aesthetics</b> must be considered for female smartwatches.	74	Variable

#### 2.4. Moderators

The moderator is an important role in this study. The roles of moderators include overall scheduling, providing guidelines to carry out the experiments, results analysis and facilitating the teams with required resources. The authors of this study acted as moderators and were responsible for overseeing the whole process. The moderators take outputs from both the SNS-based RE approach and the traditional RE approach to analyze the effectiveness and efficiency of the proposed approach. The final output of the SNS-based RE approach is a set of SPL requirements including VR and CR.

### 3. Experiment Design

To validate our proposed approach, we have performed a controlled experiment on the smartwatch domain where participants were divided into two teams. One team executed the task by using the SNS-based RE approach while the other team executed the task by following the traditional RE approach. The experiment was performed to measure the effectiveness of the proposed approach for SPL evolution. The experiment was performed by following the guidelines of Reference [33]. For comparison, both approaches were conducted in parallel for a specified period of time and results were analyzed to know the effectiveness of our proposed approach. In our controlled experiment, the participants were divided into two teams, each consisting of 15 members. Team T1 executed the task by utilizing the proposed approach and Team T2 executed the assigned task by utilizing the traditional RE approach (interview and survey). Each team was comprised of 10 graduate students with a research interest in RE, 3 software engineers with 4 years of industrial experience and 2 domain experts. The experiment is intended to measure efficiency and effectiveness of the proposed approach along with sentiment in terms of participants' feedback.

#### 3.1. Hypothesis Formulation

In order to test our proposed approach, we have formulated the following hypotheses:

**Hypothesis 1 (H1).** *The proposed approach identifies SPL requirements effectively and efficiently.*

This hypothesis is further divided into two:

**Hypothesis 1a (H1a).**  $\mu$  (Time taken by T1) >  $\mu$  (Time Taken by T2).

**Hypothesis 1b (H1b).**  $\mu$  (Valid Requirements gathered by T1) <  $\mu$  (Valid Requirements gathered by T2).

**Hypothesis 2 (H2).** *The proposed approach helps to identify VR and CR effectively and efficiently.*

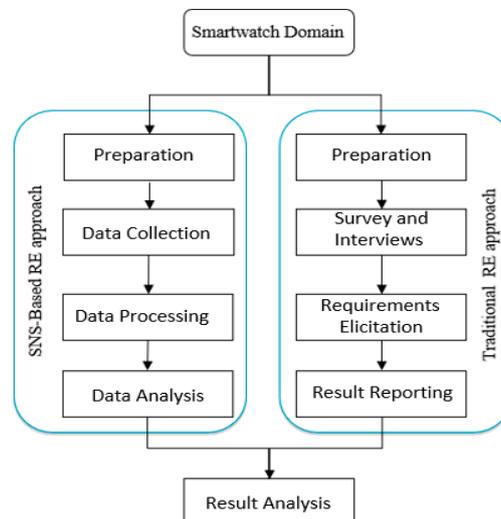
This hypothesis is further described as:

**Hypothesis 2a (H2a).**  $\mu$  (Time taken by T1 to determine VR and CR) >  $\mu$  (Time taken by T2 to determine VR and CR).

**Hypothesis 3 (H3).** *More end-users involved in the SNS-based RE process.*

### 3.2. Experimental Process

As mentioned above, two teams participated in the controlled experiment. One team (T1) used the SNS-based proposed approach to conduct RE activities conjoining Facebook and to elicit user requirements. Another team (T2) conducted RE activities without using the proposed approach. Figure 5 shows the process of our controlled experiment.



**Figure 5.** Experimental Process.

### 3.3. Proposed Approach

Our research method consists of three steps: opinion collection, opinion processing, and data analysis, as mentioned in Section 2. Team T1 was responsible for executing RE tasks with the SNS-based proposed approach. In this section, we follow the experimental process to perform our task.

#### 3.3.1. Preparation

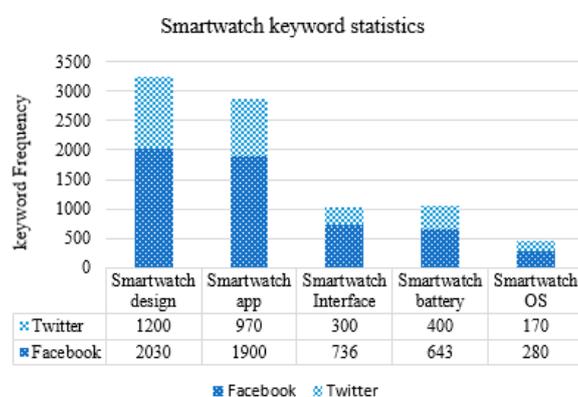
In the preparation phase, the domain experts gave roles to the participants. The roles were given based on participant's knowledge in the related domain. The roles were given as follows:

- Domain Experts (2)
- Project Manager (1)
- Team Leader (1)
- Requirements Engineers (3)
- Data Processing Engineers (3)
- Quality Assurance Engineers (2)
- Developers (4)

Team T1 was comprised of two domain experts, one project, manager, one team leader, three requirements engineers, three data processing engineers, two quality assurance engineers, and four

developers. One requirements engineer's role was duplicated and participated as a developer in team T1. During the preparation phase, keywords related to the smartwatch domain were extracted from Facebook and Twitter using their APIs (Application Programming Interface). Related keywords from two Facebook pages (Smartwatch iwatch, Wearable-watch) were extracted. For Twitter, five hashtags (#smartwatch design, #smartwatch app, #smartwatch interface, #smartwatch battery, #smartwatch OS) were used to extract keywords. The hashtag (#) allows users to search for specific keywords on Twitter. Searching through hashtags becomes very effective when Twitter is viewed at a massive scale to get opinions of masses towards a specific topic, such as knowing peoples' opinions of a certain public figure (#Obama) or certain events (#elections) [34]. The two Facebook pages such as Smartwatch iwatch and Wearable-watch were selected because these pages were dedicated to discussing only smartwatch-related topics.

Figure 6 shows the statistics of extracted keywords for the smartwatch domain. It shows the frequency of specific keywords on Facebook and Twitter.



**Figure 6.** Statistics of smartwatch domain keywords.

Based on the frequency of extracted keywords, smartwatch design and smartwatch apps keywords were selected with a frequency of 3230 and 2870 respectively, as shown in Figure 6. After keywords selection, we (in consultation with two domain experts) developed two queries to initiate discussion on SNS to collect users' opinions for the smartwatch domain: one for the application sub-domain and the other for the smartwatch design sub-domain as mentioned below:

Q1: Does smartwatch provide enough applications for you?

Q2: What does your experience say about smartwatch design?

The Q1 was expected to be either yes or no. If yes, then members of T1 do not need to go further. In the case of no, "what is your need" has been exploited. The Q2 has been formed to address the needs of end-users. By using Q2, end-users were expected to report bugs, the shortcomings of features, etc. The "smartwatch design" refers to hardware and software and other services provided in terms of features. In response to Q2, users were expected to share their experiences about their smartwatches in terms of problems (bugs, shortcomings, etc.) they faced or the advantages they got.

After developing domain-related queries, well-known SNS were investigated against the criteria mentioned in Section 2 and Facebook and Twitter were selected as candidate SNS to perform the experiment. After selection of SNS, queries were posted on selected SNS and users started the discussion and exchanged their opinions on the posted questions. For example, some users reported that "My Apple Watch is still malfunctioning after a restart" and "I find many problems with this DM98 smart watch. Firstly, the sim is not supported. It shows mobile network not available whereas the same sim card works properly on any mobile handset (in India). Secondly, the weather shows the location of Alaska by default and it can't be changed to my location or any other location. Thirdly, the voice input keyboard is not supported. Dear researchers, if you have solutions for any of the above

issues please let us know”, respectively. The users reported bugs in an effective way. Such kinds of information were very useful in order to elicit user requirements.

The requirement engineers of team T1 were actively involved in discussion with users to make things clear when necessary. Requirements engineers of team T1 followed those users who expressed their opinion in a vague way. For instance, a user stated, “it should support multifunction imo”. Here, requirements engineers followed such type of users to elicit what users really want. The requirements engineers asked in reply “what function do you think it should support”. The user replied it should support a multi-lingual function.

### 3.3.2. Data Collection

After posting related queries in the preparation phase, requirements engineers started to extract user opinions from 7 March 2018 to 15 March 2018. Both Facebook and Twitter provide APIs to extract the opinions. Therefore, requirements engineers of T1 used these APIs to extract data from both Facebook and Twitter, respectively. A brief description of the purpose of this study was posted along with queries to let users know the purpose of this activity.

Initially, requirements engineers collected 12,000 opinions from Facebook and Twitter for further processing. Figure 7 shows the number of opinions collected per day over the course of the data collection process. We observed that users were more active on the first four days of the data collection process.



**Figure 7.** Number of opinions collected per day.

Requirements engineers of team T1 also collected information such as nationality (geographical information), emoticons, and gender of those users who were engaged in the discussion. The information of nationality was used for providing multi-lingual information, and the emotions were also useful to determine user preference for specific functionality of a product. A total of 8373 users from 14 countries took part in the SNS-based RE process. Figure 8 shows the gender distribution of users who participated in the SNS-based RE process. The gender attribute is usually used for gender-driven design in software engineering [35].

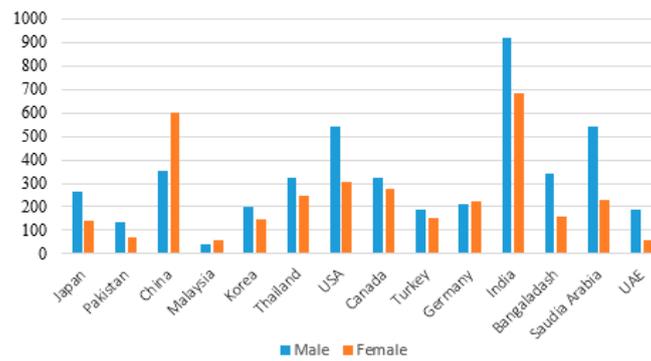


Figure 8. Gender distribution of participants in the SNS-based RE process.

The gender attribute was useful to provide a gender-inclusive design environment in this experiment. For example, Figure 9 shows the relationship between interface requirements with respect to the region (Chinese people do not like green color due to culture). Figure 10 shows sample corpus with discussion, and comment likes.



Figure 9. Gender- and Culture-sensitive requirements.

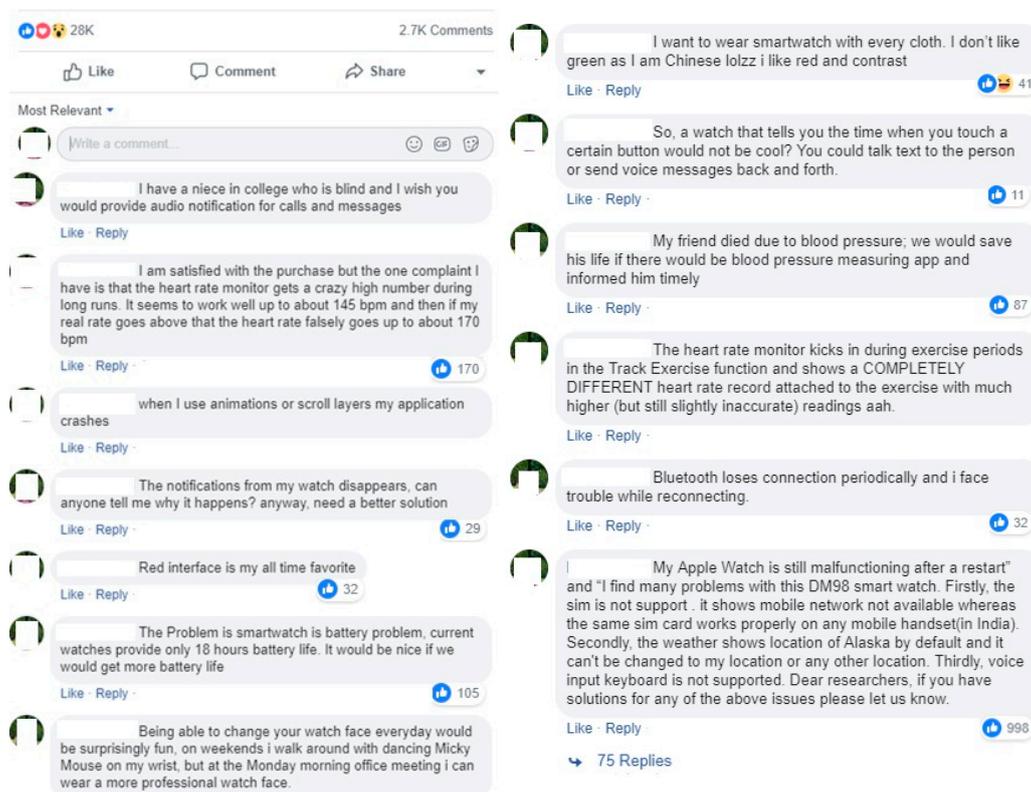


Figure 10. Sample Corpus with discussion and comment likes.

### 3.3.3. Data Processing

After opinions extraction, the opinions were cleaned to remove garbage. Subsequently, NLP resources such as stop words was generated for the smartwatch domain. WordNet lemmatization was utilized to group together the different forms of words to make them single items. The main objective of this phase was to elicit the requirements that users are most interested in. In the end, bigram and trigram were extracted to get the candidate requirement terms from the corpus. The requirements engineers interpreted those extracted keywords into requirements.

All the extracted requirements were not useful or genuine requirements. There were some irrelevant and redundant requirements as well. Therefore, compactness and redundancy pruning were utilized to remove the irrelevant and redundant requirements, respectively. Table 5 shows the volume and ratio of our extracted requirements' polarity such as positive, irrelevant, and redundant. The frequency (number of occurrences) of requirements were recorded during data processing to help in the data analysis phase.

**Table 5.** Volume and ratio of our extracted requirements' polarity.

	Attributes	Count	Percentage
Total		12,000	100%
	Redundant	1470	12.25%
	Irrelevant	2400	20%
	Positive	8130	67.75%

Since the discussion was driven by posting specific questions about the smartwatch domain, therefore, about 68% of opinions got the positive attribute. The positive opinions were selected to extract user current demands related to the smartwatch domain, especially smartwatch apps, and design sub-domains.

### 3.3.4. Data Analysis

In the data analysis phase, the elicited requirements were analyzed to extract user needs in terms of VR and CR. That is, the requirements with the positive attribute were processed further to extract VR and CR from the requirements. Equation (1) was used to extract VR and CR.

For evaluation of the extracted requirements, a human tagger manually went through all the requirements and made a requirement list that shows common and variable requirements of smartwatch domain (apps and design). Table 6 shows common and variable requirements of smartwatch domain along with total extracted requirements.

**Table 6.** Common and Variable Requirements Collected by T1.

Sub-Domain	Number of Requirements	Common Requirements	Variable Requirements
Apps	128	91	32
Design	99	76	28
Total	227	183	44

## 3.4. Traditional Approach

RE is one of the most important phases in the software development cycle. It is mostly concerned with elicitation, documentation, and maintaining stakeholder's requirements [36]. Meeting stakeholders' needs is considered the baseline of any quality software system [37].

In this section, we elaborate on the RE process carried out by team T2 to elicit user requirements for the smartwatch domain.

### 3.4.1. Preparation

In parallel with our proposed approach, the traditional requirements elicitation approach (survey and interview) was carried out through an online survey and interview to elicit users' requirements for the smartwatch domain. Traditional requirements elicitation protocols were followed to conduct this survey [38,39]. The team T2 was responsible for executing this task. In the preparation phase, the domain experts gave roles to the participants. The roles were given as follows:

- Domain Experts (2)
- Project Manager (1)
- Team Leader (1)
- Requirements Engineers (5)
- Quality Assurance Engineers (2)
- Developers (4)

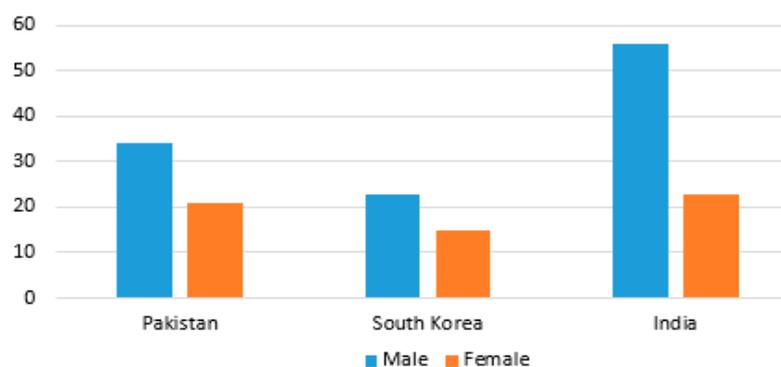
Team T2 was comprised of two domain experts, one project manager, one team leader, five requirements engineers, two quality assurance engineers, and four developers. After assigning roles, requirements engineers made a questionnaire in consultation with the project manager and domain experts to elicit user requirements. Domain experts told requirements engineers to include Q1 and Q2 as fundamentals questions while making a questionnaire for the survey. The participants were also told to record the response in terms of the number of people per day. The project manager asked the requirements engineers to target users as much as possible.

Following are some sample questions that were used by team T2 to elicit user requirements:

- Does the smartwatch provide enough applications for you?
  - If no, what type of application do you want?
- What does your experience say about smartwatch design?
- Do you prefer gender-specific interface design?
  - If yes, what do you prefer?
- How long should your smartwatch battery last?

### 3.4.2. Survey and Interview

Requirements engineers of team T2 targeted 350 users from Pakistan, South Korea, and India for an online survey. The questionnaire was sent through emails of the targeted users. 172 out of 530 users responded. Out of total respondents, 41 respondents were subsequently selected to know their requirements by an online interview. Figure 11 shows the gender distribution of users who participated in interviews and surveys.



**Figure 11.** Gender distribution of participants in the traditional RE process.

The response from the users was quite slow and team T2 members had to wait for user response. Time for each activity was recorded to know the overall time spent on whole activities. Finally, 74 requirements were collected, out of which 64 were functional and 11 were non-functional requirements.

The requirements engineers of team T2, in consultation with domain experts and the project manager, decided VR and CR of requirements based on the frequency of requirement. Meaning that if a requirement is demanded by the majority, it would be a potential candidate to be a common requirement. The requirements which were less frequent were decided as variable requirements. Along with this, domain experts used their own domain knowledge to determine VR and CR. Finally, 59 requirements were classified as common requirements and 15 requirements were classified as variable requirements.

### 3.4.3. Result Reporting

After performing the requirements elicitation task, team T2 forwarded their final requirements document to moderators. Table 7 shows user requirements collected by team T2 using the traditional RE approach. Team T1 gathered 74 requirements out of which 43 were for the smartwatch App domain and 31 were for the smartwatch Design domain. Furthermore, 47 requirements were determined as common requirements and 15 requirements were classified as variable requirements.

**Table 7.** Requirements elicitation by the traditional RE approach.

Sub-Domain	Number of Requirements	Common Requirements	Variable Requirements
Apps	43	37	11
Design	31	22	04
Total	74	47	15

## 4. Result Analysis

In this section, we analyzed the collected data from T1 and T2. In our controlled experiment, there is only one independent variable to measure. The independent variable is the approach used to execute RE activities. However, T1 and T2 are two levels of independent variables. Team T1 conducted RE activities using the proposed approach while team T2 executed the task with traditional RE approach.

The variables that change when independent variable shifts from level one to level two (T1 and T2) are called the dependent variable in this study. The effectiveness, efficiency, and sentiment are three dependent variables which change when the independent variable changes. Effectiveness of independent variables was measured in terms of collection of requirements and identification of variable and common requirements. The effectiveness compares which of the two teams collected more SPL requirements in terms of VR and CR.

The efficiency was measured in terms of time taken by teams (T1 and T2) to accomplish their task. Time of team T1 can be compared with the time taken by T2. Hence, we can determine which of two teams take more time and vice versa.

The sentiment is measured objectively by taking feedback from participants of T1 and T2 following the experiment. In the following subsection, we will measure effectiveness, efficiency, and sentiment.

### 4.1. Effectiveness

In order to measure effectiveness, we compared the documented requirements of team T1 and team T2 for the smartwatch domain. Therefore, after receiving the requirements document from both teams, which team collected more useful SPL requirements was then analyzed. The summary of collected requirements is shown in Table 8. The total requirements given by team T1 are 227 out of which 175 are functional requirements and 52 are non-functional requirements, and also team T2 identified 167 common requirements and 60 as variable requirements. Similarly, team T2 identified a total of 74 requirements out of which 63 were functional requirements while 11 were non-functional

requirements. Team T2 classified 59 requirements as common requirements and 15 were classified as variable requirements.

**Table 8.** Comparison of Team “T1” and Team “T2” over Requirements.

	T1				T2			
	FR	NFR	CR	VR	FR	NFR	CR	VR
<b>App</b>	101	27	91	32	39	4	37	11
<b>Design</b>	74	25	76	28	24	7	22	4
<b>Total</b>	175	52	167	60	63	11	59	15

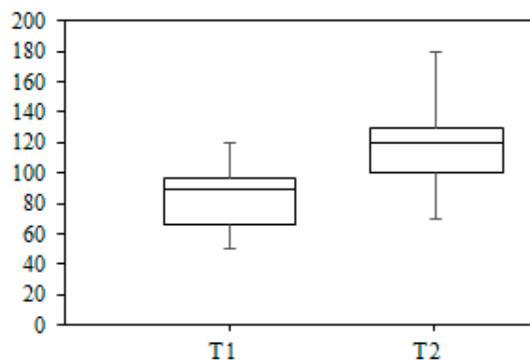
FR: Functional Requirements, NFR: Non-Functional Requirements, CR: Common Requirements, VR: Variable Requirements.

Table 8 shows that team T1 collected more requirements for the smartwatch domain. This is due to the fact that team T1 approached more users through SNS and collected more requirements. It is widely believed that more accurate user requirements can be gathered by involving more users [40–42]. It is also noted that team T1 classified more requirements as variable and common requirements because they were easily decided due to the provided criteria, as mention in Equation (1).

#### 4.2. Efficiency

In order to measure the efficiency, time taken by each team was recorded. The recorded data shows that the time taken by team T2 was more than the time taken by team T1. The boxplot is used to express the results to show the time comparison between team T1 and team T2.

Figure 12 shows the boxplot for team T1 and team T2. The boxplot for team T1 has a median of 89 min. The whiskers show that minimum time taken by a member of team T1 to identify a requirement was 50 min and the maximum time taken by a member a member of T1 was 120 min. For team T2, boxplot shows that maximum time taken by a member of team T2 was 180 min and minimum time taken by a member of team T2 was 70 min. The median time taken by members of team T2 was 120 min.



**Figure 12.** Boxplot for total time taken by two teams to execute the RE task.

The time taken by team T2 was more because members of team T2 waited for the response from users whom they sent a request to do a survey. Also, requirements engineers of team T2 conducted interviews to elicit requirements which also took more time, while the team T1 used the SNS-based approach to conduct RE activities and they found a huge population on SNS at once and did not take more time to elicit user requirements. Therefore, the results demonstrate that the SNS-based proposed RE approach takes less time to conduct overall activities.

Figure 13 shows the boxplot in order to compare the time taken by both teams to identify VR and CR. The median time for team T2 was 38 min for each participant to determine VR and CR. The whiskers show that minimum time taken by a participant of team T2 was 36 min and maximum time taken by participants of team T2 was 80 min to identify VR and CR. This was due to the fact that

there were more conflicts among participants as to whether the requirements should be categorized as common or variable requirements. The domain experts were frequently involved to help to resolve the conflicts.

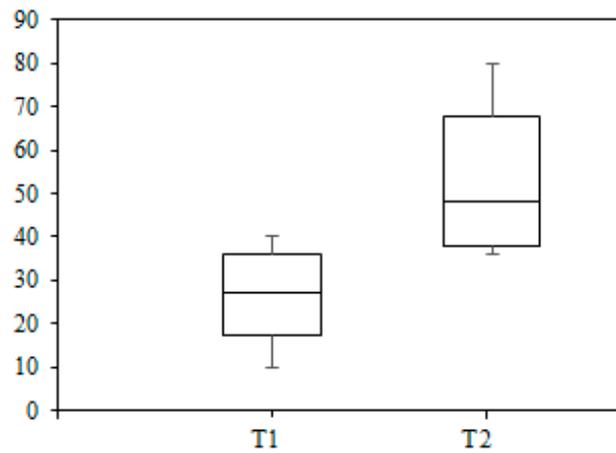


Figure 13. Boxplot for total time taken by two teams to determine Variability and Commonality.

The boxplot also shows that participants of team T1 took less time to identify variable and common requirements. The participants in team T1 took 27 min median time.

The maximum time taken was 40 min and the minimum time taken was 10 min to determine VR and CR. The median time was a little closer to the mean time of team T2 because the requirements engineers found some keywords which were different but were either synonyms or variations of an already existed keyword. Therefore, the requirements engineers had to decide about the VR and CR of requirement in this case. The overall time taken by team T1 was less because they determined VR and CR of requirements based on the predefined equation (Equation (1)).

#### 4.3. Sentiment

In order to know the sentiment of participants in the SNS-based approach and the traditional RE approach, we collected data by asking the opinion regarding their satisfaction about the performance of the proposed approach. The opinion was expressed by following a Likert scale from 1 (very low) to 5 (very high) concerning (i) how the proposed approach is effective and efficient and (ii) how easy it was to implement. The mean of these two scores is shown in Table 9. The mean value of team T1 for effective and efficient (EE) is 4.26 and 3.76 for easiness of the SNS-based proposed approach. The feedback shows that the proposed approach is effective and efficient, but participants still prefer the traditional RE approach because of its easiness. The feedback mean-value of team T2 for the traditional RE regarding efficiency and effectiveness was 3.86. The mean value for easiness was 4.40. In summary, the proposed approach was globally judged positive by team T1.

Table 9. T-test for Sentiment.

Participants		Effective and Efficient	Easy
T1	Mean	4.2667	3.7667
	N	15	15
	Std.Deviation	0.49522	0.59362
T2	Mean	3.8667	4.4000
	N	15	15
	Std.Deviation	0.66726	0.60356

#### 4.4. Hypothesis Testing

In order to test the formulated hypotheses, we start from the first hypothesis i.e., null hypothesis H10 and alternate hypothesis H1.

##### 4.4.1. First Hypothesis

The formulated H10 null hypothesis states that:

a)  $\mu$  (Time taken by T1) =  $\mu$  (Time Taken by T2)

b)  $\mu$  (Valid Requirements gathered by T1) =  $\mu$  (valid requirements gathered by T2)

For the testing the first part of the null hypothesis H10, we used a t-test on our data using the IBM SPSS statistical tool. We take a 95% confidence interval, therefore, statistical significance is set to  $\alpha = 0.05$ . As a result of our t-test, we generated the following results. Table 10 shows general information such as the number of participants in each team, mean time for both teams, standard deviation, and standard error mean for our data set.

**Table 10.** Group Statistics: Mean Time for two Teams.

	Team	N	Mean	Std.Deviation	Std.Error Mean
Time	T1	15	82.2667	18.87730	4.87410
	T2	15	114.2000	25.64928	6.62262

Table 11 shows upper and lower bounds on the 95% confidence interval, Mean difference, standard-error difference, and other important significant levels of dependent variable efficiency of both teams T1 and T2. As shown in Table 11, the significance value for the SNS-based proposed approach is 0.001, compared to the traditional RE approach carried out by team T2. The significant value was less than  $\alpha = 0.05$ , which means that team T1 has shown more efficiency of eliciting user requirements than team T2.

**Table 11.** T-test for Efficiency.

t	df	Sig.(2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
					Lower	Upper
-3.883	28	0.001	-31.93333	8.22289	-48.77715	-15.08851

Figure 12 also shows that team T2 has taken more time than team T1 to conduct RE activities for smartwatch domain. The boxplot shows that team T1 has a median 89 min and team T2 has a median 120 min. This means that the median of T1 is less than the median of T2.

It is also shown in Table 10 that the mean time for team T1 was less than the mean time of team T2. Therefore, we can describe it as:

$\mu$  (Time taken by team T1) = 89 min

$\mu$  (Time Taken by team T2) = 120 min

Hence,

$\mu$  (Time taken by T1) <  $\mu$  (Time Taken by T2)

Thus, we reject the part a of the null hypothesis H10 because it states that  $\mu$  (Time taken by T1) =  $\mu$  (Time Taken by T2).

For testing part b of the null hypothesis H10, we refer to Table 8. As shown in Table 8, the total functional requirements collected by team T1 were 175 and total non-functional requirements collected by team T1 were 52 requirements. Functional requirements gathered by team T2 were 63 and non-functional requirements were 11. So,

Total requirements collected by team T1= 227

Total requirements collected by team T2= 74

Hence, it clearly tells that valid requirements gathered by team T1 were more than the valid requirements collected by team T2. Therefore, it can be stated as:  $\mu$  (Valid Requirements gathered by T1) >  $\mu$  (valid requirements gathered by T2). Therefore, based on this evidence, we reject the part *b* of the null hypothesis H10 which states that  $\mu$  (Valid Requirements gathered by T1) =  $\mu$  (valid requirements gathered by T2).

We have also seen that part *a* and part *b* of the null hypothesis H10 are rejected. Hence, the null hypothesis H10 is rejected and the alternate hypothesis H1 is accepted.

#### 4.4.2. Second Hypothesis

In this subsection, we test the null hypothesis H20 which states that:

$$\mu \text{ (Time taken by T1 to determine VR and CR)} = \mu \text{ (Time taken by T2 to determine VR and CR)}$$

As mentioned above, the time to determine VR and CR by both teams was recorded. We applied a t-test on that data, and we took a 95% confidence interval, therefore, statistical significance was set to  $\alpha = 0.05$ . As a result of our t-test, we generated the following result.

Table 12 shows upper and lower bounds on the 95% confidence interval, Mean difference, standard-error difference, and other important significant levels of dependent variable efficiency of both teams T1 and T2 in order to determine variable and common requirements. The significant value for the SNS-based proposed approach was 0.001, as compared to identifying the variable and common requirements by the traditional RE approach. Table 12 shows that the significant value (0.001) is less than  $\alpha = 0.05$ , meaning that team T1 has shown more efficiency in determining variable and common requirements than team T2.

**Table 12.** T-test for Efficiency in determining VR and CR.

t	df	Sig.(2-tailed)	Mean Difference	Std. Error Difference	95 % Confidence Interval of the Difference	
					Lower	Upper
-4.310	16	0.001	-27.44444	6.36784	-40.94366	-13.94523

Figure 13 also shows that the median time taken by team T1 was 27 min and the median time for team T2 was 48 min. This means that the time taken by team T2 to determine variable and common requirements was less than team T2. Therefore, we can describe it as:

$$\mu \text{ (Time taken by T1 to determine VR and CR)} = 27 \text{ min}$$

$$\mu \text{ (Time taken by T2 to determine VR and CR)} = 48 \text{ min.}$$

Hence,  $\mu$  (Time taken by T1 to determine VR and CR) <  $\mu$  (Time taken by T2 to determine VR and CR). Thus, we rejected the null hypothesis H20 because it states that  $\mu$  (Time taken by T1 to determine VR and CR) =  $\mu$  (Time taken by T2 to determine VR and CR) and accepted the alternative hypothesis H2.

#### 4.4.3. Third Hypothesis

The third null hypothesis states that:

H30:  $\mu$  (Number of users involved in the SNS-based RE process) =  $\mu$  (Number of users involved in traditional RE process)

To test this hypothesis, we refer to Figure 8 for data collection.

$$\mu \text{ (Number of users involved in SNS-based RE process)} = 8373$$

$$\mu \text{ (Number of users involved in traditional RE process)} = 172$$

Therefore,  $\mu$  (Number of users involved in the SNS-based RE process) >  $\mu$  (Number of users involved in the traditional RE process). Hence, we reject the null hypothesis H30 and accept the alternate hypothesis H3.

## 5. Threats to Validity

Our Experimental results show that the proposed SNS-based RE approach can effectively elicit user requirements from SNS. In this section, we discuss a few threats to the validity of this research and the way we tried to improve those threats.

A potential threat to the presented study's internal validity is the human judgement in the data analysis phase, where final requirements have been analyzed by moderators along with domain experts. This might lead to experimental bias as humans tend to be biased in their judgement. However, it is infrequent in requirements interpretation and analysis to employ humans to manually interpret and classify requirements. Therefore, the bias threat which arises from using humans is not imminent, it can be avoided or mitigated by setting up a common mechanism to determine VR and CR. In order to mitigate this threat, we used Equation (1) to reduce bias.

The threats to external validity affect the generalizability of experimental results [43]. A possible threat emerges due to the fact that our dataset was limited in size and we have used limited Facebook pages and tweets to extract the user requirements. A bigger dataset from more SNS sources might affect the results. However, we found various types of user requirements within our experimental scope. Therefore, even if the data set was not so big in our experiment, we have confidence in our research results. Additionally, we chose Facebook and Twitter for gathering the specific domain concerns, even though there are other SNS like Instagram, LinkedIn, and ResearchGate. Facebook and Twitter are the two most common and general-purpose social networks where billions of users are present. Therefore, we selected only Twitter and Facebook as platforms for our experiment.

Another threat to validity is that SNS are not built on the perspective of RE. Therefore, comments and tweets might not include meaningful data for software development organizations. The SNS are also used for marketing and advertisement purposes by the software development organization. Therefore, Facebook comments and tweets might only include marketing messages rather than user feedback. Therefore, moderators started the discussion on SNS by posting related queries so that the discussion would be on a focused issue rather than a general discussion. However, the SNS is a major source to get users' instant feedback from diverse end-users, and also a place to put up a variety of opinions freely.

One of the threats to the validity is a construct validity which is a degree to which variables measure the things they are supposed to. In our study, we identified three dependent variables and one independent variable to measure our SNS-based approach. Obviously, other variables can also be identified to measure other facets as well. Therefore, it is not possible to capture all dimensions in a single controlled experiment. In our study, effectiveness, efficiency and sentiment were the three variables because these measures sufficiently support to prove our hypothesis.

## 6. Related Work

The RE research community has started to consider SNS for enhancing the participation of end-users in the development of software systems. Reaching out to a huge population to collect the needs of anonymous users can guarantee the quality of ecosystems [37]. Traditional RE approaches, for instance, interviews, day-long seminars, and surveys are very expensive when stakeholders are globally distributed. To address this issue a number of web-based approaches are proposed [10–12]. The SNS-based strategies [44] have proved their effectiveness in the RE field [13–15,45,46].

Finkelstein and Lim et al. [10–12] proposed web-based tools that easily elicit and prioritize requirements recommended by stakeholders. These tools automatically prioritize requirements based on the stakeholders' rating. The roles of shaper, facilitator, and moderator have similarly been proposed to induce the requirements and eventually prioritize them.

SNS such as Twitter and Facebook brought revolution in many data science fields by producing a huge amount of valuable data every day [44]. Such a kind of users' feedback can be used to elicit user requirements. Therefore, researchers have considered SNS as a better alternative to conduct RE activities [9,10,47–52].

Andrew et al. [39] have enlisted the potentials of SNS for software engineering and the effectiveness of SNS to understand needs of end-users. Margaret et al. [53] argued that SNS can support software development activities such as RE, development, testing, and documentation.

Seyff et al. [9] used SNS to support requirements elicitation, prioritization, and analysis. The authors conducted three experimental studies to demonstrate the effectiveness of SNS in the field of RE. Results show that the popular SNS, i.e., Facebook can support distributed RE. But the participants of the experiments are limited to the students of a university which does not represent the potential users across the globe.

Lee et al. [48] also used SNS to elicit user requirements from Facebook and accessed a wide range of users. The authors conducted a case study to demonstrate the validity of their approach as compared to traditional RE approaches and extracted 67 user requirements within their domain. They also validated the usage of SNS for the elicitation of user requirements but this RE process did not consider the product line perspective including VR and CR analysis.

Seo et al. [49] proposed an approach based on SNS to analyze the impact of customer requirements. The authors used network characteristics of users to prioritize user requirements. Singer et al. [51] presented a survey study where authors interviewed 27 software developers in order to know the capabilities of SNS in software development. Authors revealed that developers often rely on various online resources, such as Twitter to keep themselves informed about their systems. They also revealed that Twitter helps software developers to stay aware of industrial changes and technology changes. It was also mentioned that developers use Twitter for learning as well.

Guzman et al. [45] have proposed an approach which uses Twitter as a platform to elicit user requirements. The authors presented an exploratory study to investigate the content of 10,986,494 tweets about 30 software applications. The authors have classified tweets into 22 categories including feature shortcomings, bug reports, feature request, and feature strength, etc.

Ming et al. [50] have used popular SNS such as StackOverflow, a popular Q and A software community site where programmers and software engineers share their knowledge and experience to elicit user requirements. The collected data from StackOverflow was used to extract requirements requests to help software developers.

Guzman et al. [13] presented a study for classifying, ranking and grouping tweets for software evolution. The authors used Twitter as a platform to elicit user requirements. The proposed approach classifies tweets into two categories including improvement requests and others. The authors used 68,108 tweets for their experiment. In summary, the proposed approach was able to classify tweets automatically into improvement request and other categories with an F-measure of 0.79.

Table 13 shows a summary of studies that are relevant to our research. The “●” symbol represents that the respective column aspect is present in the author’s research and “○” symbol represents that the respective aspect is not considered in the author’s research.

**Table 13.** Summary of related work that considers aspects related to our research.

Reference	SNS	Moderator	Requirements Elicitation	VR and CR Analysis
[9]	●	●	●	○
[52]	●	○	○	○
[49]	●	○	●	○
[51]	●	○	●	○
[50]	●	○	●	○
[31]	●	○	●	○
[16]	●	●	●	○
[12]	●	●	●	○
[11]	●	●	●	○
[10]	●	○	●	○

Table 13. Cont.

Reference	SNS	Moderator	Requirements Elicitation	VR and CR Analysis
[45]	●	○	●	○
[15]	●	○	●	○
[14]	●	○	●	○
[13]	●	○	●	○
[48]	●	●	●	○
[54]	●	○	●	○
[55]	●	○	●	○

SNS: Social Network Services, VR: Variable Requirements, CR: Common Requirements.

N. Seyff et al. [54] presented a conceptual solution to elicit crowd-based user requirements. The conceptual framework has three basic components that could work together to support continuous elicitation and negotiations: (1) CrowdFeed is proposed to allow users to communicate feedback regarding the services and software products they use and to actively participate in negotiation, (2) Requirements and Sustainability Service clusters, classifies and analyses user feedback which is received from CrowdFeed component, (3) the Requirements and Sustainability Integrator component supports visualization and assessment of effects on sustainability.

Ali et al. [55] proposed an approach that aims to mine SNS such as twitter and Facebook data to elicit user requirements. The authors collected 30,633 tweets from Twitter and 18,482 comments from Facebook in order to determine user requirements. Topic modeling was used to discover topics that exist in their dataset. Classification algorithms such as support vector machine, multinomial Naive Bayes, and random forest were also used to classify SNS data into bug reports, new feature demands, quality attributes (nonfunctional requirements) and irrelevant opinions. The authors found that 47% of opinions were classified as irrelevant opinions while 24% were reported to be bug reports, 20% were new feature demands or requests and 9% opinions were discussing quality attributes. The authors [56] also proposed an approach to elicit recurring requirements for SPLs.

In conclusion, the related work on requirements elicitation using SNS has the confinement that the research conducted for requirements elicitation and analysis has not addressed the issue of eliciting requirements from a larger group of users considering VR and CR. The survey [57] reveals that there are some RE approaches of SPL which are used to extract requirements from web repositories, however, the related work has not considered SNS (Facebook and Twitter) to elicit user requirements to evolve SPL systems.

To summarize the contribution from related work presented in this section, we argue that the RE community has not considered SNS yet to elicit customer requirements due to the investigation complexity of VR and CR [58].

## 7. Conclusions and Future Work

With the rapid expansion of internet innovations, SNS have become a useful source for gathering customer opinions. Taking advantage of this big opportunity, software requirements engineers have started considering SNS to elicit user requirements in order to evolve the SPL products quickly. Users' opinion is essential for SPL organizations to better understand the general responses of end-users to their products and improve their products with the passage of time. In addition, users' opinion enables SPL organizations to figure out the specific needs of individual users that can facilitate subsequent SPL strategies to meet the specified needs of the individual user.

This paper presented an SNS-based RE process to elicit user requirements from SNS considering SPL systems. The key superiority of this article is that it is considering VR and CR while other research is focused on gathering requirements and stakeholder prioritization. We have proposed a model to determine VR and CR by considering the attributes of user opinion which is a distention of this research.

The applicability of the SNS-based RE process for elicitation of requirements to evolve SPL products is shown through a controlled experiment performed by using Facebook and Twitter. The participants in the experiment were grouped into teams each consisting of 15 members with various roles. The task was to elicit user requirements for the smartwatch domain. One team conducted the task by using the SNS-based RE process conjoining with Twitter and Facebook and the other group executed the task with the traditional RE process, i.e., by interview and online surveys. We gathered the data from both teams and performed tests on the provided data to know the efficiency and effectiveness of the SNS-based RE process. The results from the experiment show that the proposed SNS-based RE process effectively and efficiently elicits user requirements considering VR and CR.

As mentioned above, our proposed approach has two parts: (1) identifying requirements using SNS and (2) classifying requirements into variable requirements and common requirements using Equation (1). If we consider VR and CR as general plain requirements, then our approach can be used as a general approach to elicit users' requirements from SNS. Our approach can also be used in reused-based software development which can expedite the time to market, reduce laborious work and improve productivity and quality.

This research has limitations which will be addressed in future work. There are moderator roles and domain expert roles who are in fact human beings. There is a chance of human error in interpreting the extracted keywords into requirements. Therefore, automatic tools should be developed with the help of NLP to process the users' opinion. Eventually, this will enable us to minimize the role of moderators and domain experts.

The extended studies connected with our research can consider a technique to dynamically map between elicited requirements to the components of domain architecture.

**Author Contributions:** Conceptualization, N.A. and J.-E.H.; Methodology, N.A.; Validation, J.-E.H.; Statistical Analysis, N.A.; Data Collection, N.A.; Writing—Original Draft Preparation, N.A.; Writing—Review and Editing, J.-E.H.; Supervision, J.-E.H.; Experimental Administration, J.-E.H.; Funding Acquisition, J.-E.H.

**Funding:** This research was supported by Next-Generation Information Computing Development Program through the NRF of Korea funded by the Ministry of Science, ICT (NRF-2014M3C4A7030505, NRF-2017M3C4A7066479).

**Acknowledgments:** We thank all members of the SE laboratory for their support.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Böckle, G.; Pohl, K.; van der Linden, F. A framework for software product line engineering. In *Software Product Line Engineering*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 19–38.
2. Marques, M.; Simmonds, J.; Rossel, P.O.; Bastarrica, M.C. Software product line evolution: A systematic literature review. *Inf. Softw. Technol.* **2019**, *105*, 190–208. [[CrossRef](#)]
3. Ali, N.; Choi, Y.; Kim, S.; Hong, J. Which factors should be considered for domain testing? *Int. Conf. Conver. Technol.* **2016**, *6*, 910–911.
4. Obar, I.; Wildman, S.S. Social media definition and the governance challenge: An introduction to the special issue. *Telecommun. Policy* **2015**, *39*, 745–750. [[CrossRef](#)]
5. Kaplan, A.M.; Haenlein, M. Users of the world, unite! The challenges and opportunities of Social Media. *Bus. Horiz.* **2010**, *53*, 59–68. [[CrossRef](#)]
6. Ellison, N.B.; Boyd, D.M. Social network sites: Definition, history, and scholarship. *J. Comput. Mediat. Commun.* **2007**, *13*, 210–230.
7. Ali, N.; Hong, J.E. Using Social Network Service to determine the Initial User Requirements for Small Software Businesses. *Int. J. Appl. Bus. Econ. Res.* **2017**, *15*, 259–268.
8. Mao, K.; Capra, L.; Harman, M.; Jia, Y. A survey of the use of crowdsourcing in software engineering. *J. Syst. Softw.* **2017**, *126*, 57–84. [[CrossRef](#)]
9. Seyff, N.; Todoran, I.; Caluser, k.; Singer, L.; Glinz, M. Using popular social network sites to support requirements elicitation, prioritization and negotiation. *J. Internet Serv. Appl.* **2015**, *6*, 7. [[CrossRef](#)]

10. Kukreja, N. Winbook: A social networking-based framework for collaborative requirements elicitation and WinWin negotiations. In Proceedings of the 34th International Conference on Software Engineering, Zurich, Switzerland, 2–9 June 2012; pp. 1610–1612.
11. Lim, S.L.; Quercia, D.; Finkelstein, A. StakeNet: Using social networks to analyse the stakeholders of large-scale software projects. In Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering, Cape Town, South Africa, 1–8 May 2010; Volume 1, pp. 295–304.
12. Lim, S.L.; Damian, D.; Finkelstein, A. StakeSource2.0: Using social networks of stakeholders to identify and prioritise requirements. In Proceedings of the 33rd International Conference on Software Engineering (ICSE), Honolulu, HI, USA, 21–28 May 2011; pp. 1022–1024.
13. Guzman, E.; Ibrahim, M.; Glinz, M. A little bird told me: Mining tweets for requirements and software evolution. In Proceedings of the IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 4–8 September 2017; pp. 11–20.
14. Williams, G.; Mahmoud, A. Mining Twitter feeds for software user requirements. In Proceedings of the IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 4–8 September 2017; pp. 1–10.
15. Pagano, D.; Maalej, W. User feedback in the appstore: An empirical study. In Proceedings of the 21st IEEE international requirements engineering conference (RE), Rio de Janeiro, Brazil, 15–19 July 2013; pp. 125–134.
16. Mughal, S.; Abbas, A.; Ahmad, N.; Khan, S.U. A Social Network Based Process to Minimize In-Group Biasedness During Requirement Engineering. *IEEE Access* **2018**, *6*, 66870–66885. [CrossRef]
17. Esuli, A.; Sebastiani, F. Training data cleaning for text classification. In *Conference on the Theory of Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 29–41.
18. Li, C.; Liu, Y. Joint POS tagging and text normalization for informal text. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 1263–1269.
19. Lemmatization. Available online: <https://www.datacamp.com/community/tutorials/stemming-lemmatization-python> (accessed on 21 May 2019).
20. Rani, T.; Anuradha, K.; Reddy, V. Sentiment Classification on Twitter data using Word N Gram Model. *Int. J. Technol. Eng. Sci.* **2016**, *4*, 7032–7035.
21. Winter, E.; Forshaw, S.; Ferrario, M.A. Measuring human values in software engineering. In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Oulu, Finland, 11–12 October 2018; pp. 1–4.
22. Thew, S.; Sutcliffe, A. Value-based requirements engineering method and experience. *Requir. Eng.* **2018**, *23*, 443–464. [CrossRef]
23. Ruhe, G.; Wohlin, C. *Software Project Management in a Changing World*; Springer: Berlin/Heidelberg, Germany, 2014.
24. Sánchez, R.A.; Cortijo, V.; Javed, U. Students' perceptions of Facebook for academic purposes. *Comput. Educ.* **2014**, *70*, 138–149. [CrossRef]
25. Arora, C.; Sabetzadeh, M.; Briand, L.; Zimmer, F. Automated extraction and clustering of requirements glossary terms. *IEEE Trans. Softw. Eng.* **2017**, *43*, 918–945. [CrossRef]
26. Graph API. Available online: <https://developers.facebook.com/docs/graph-api/reference/v3.3/object/comments> (accessed on 10 March 2018).
27. Srivastava, J. How to Extract Tweets from Twitter in Python. Available online: <https://github.com/PacktPublishing/Python-Social-Media-Analytics> (accessed on 10 March 2018).
28. Wang, X.; McCallum, A.; Wei, X. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 697–702.
29. Neiva, D.; de Almeida, F.C.; de Almeida, E.S.; Meira, S.L. A requirements engineering process for software product lines. In Proceedings of the 2010 IEEE International Conference on Information Reuse & Integration, Las Vegas, NV, USA, 4–6 August 2010; pp. 266–269.
30. Pak, A.; Paroubek, P. Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the LREC, Valletta, Malta, 17–23 May 2010; Volume 10, pp. 1320–1326.

31. Guzman, E.; Ibrahim, M.; Glinz, M. Prioritizing user feedback from twitter: A survey report. In Proceedings of the IEEE/ACM 4th International Workshop on CrowdSourcing in Software Engineering (CSI-SE), Buenos Aires, Argentina, 22 May 2017; pp. 21–24.
32. Niwattanakul, S.; Singthongchai, J.; Naenudorn, E.; Wanapu, S. Using of Jaccard coefficient for keywords similarity. In Proceedings of the International Multiconference of Engineers and Computer Scientists, Hong Kong, China, 13–15 March 2013; pp. 380–384.
33. Claes, W.; Per, R.; Martin, H.; Magnus, C.O.; Björn, R.; Wesslén, A. *Experimentation in Software Engineering: An Introduction*; Kluwer Academic: London, UK, 2006; pp. 26–51. Available online: <http://books.google.com/books> (accessed on 22 November 2018).
34. O'Connor, B.; Balasubramanyan, R.; Routledge, B.R.; Smith, N.A. From tweets to polls: Linking text sentiment to public opinion time series. In Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, Washington, DC, USA, 23–26 May 2010; pp. 122–129.
35. Dray, S.; Daniela, B.; Brock, A.; Peters, P.A.; Bardzell, S.; Druin, A.; Burnett, M.M.; Churchill, E.F.; Williams, G.; Holtzblatt, K.; et al. Perspectives on gender and product design. In Proceedings of the CHI'14 Extended Abstracts on Human Factors in Computing Systems, Toronto, ON, Canada, 26 April–1 May 2014; pp. 53–56. [[CrossRef](#)]
36. Hujainah, F.; Bakar, R.B.A.; Abdulgaber, M.A.; Zamli, K.Z. Software Requirements Prioritisation: A Systematic Literature Review on Significance, Stakeholders, Techniques and Challenges. *IEEE Access* **2018**, *6*, 71497–71523. [[CrossRef](#)]
37. Dar, H.; Lali, M.I.; Ashraf, H.; Ramzan, M.; Amjad, T.; Shahzad, B. A Systematic Study on Software Requirements Elicitation Techniques and its Challenges in Mobile Application Development. *IEEE Access* **2018**, *6*, 63859–63867. [[CrossRef](#)]
38. Zowghi, D.; Coulin, C. Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and Managing Software Requirements*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 19–46.
39. Pacheco, C.; García, I.; Reyes, M. Requirements elicitation techniques: A systematic literature review based on the maturity of the techniques. *IET Softw.* **2018**, *12*, 365–378. [[CrossRef](#)]
40. Kujala, S. User involvement: A review of the benefits and challenges. *Behav. Inf. Technol.* **2003**, *22*, 1–16. [[CrossRef](#)]
41. Kujala, S.; Kauppinen, M.; Lehtola, L.; Kojo, T. The role of user involvement in requirements quality and project success. In Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05), Paris, France, 29 August–2 September 2005; pp. 75–84.
42. Shafiq, M.; Zhang, Q.; Akbar, M.A.; Khan, A.A.; Hussain, S.; Amin, F.E.; Khan, A.; Soofi, A.A. Effect of project management in requirements engineering and requirements change management processes for global software development. *IEEE Access* **2018**, *6*, 25747–25763. [[CrossRef](#)]
43. Dean, A.; Daniel, V.; Danel, D. *Design and Analysis of Experiments*; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1.
44. Storey, M.A.; Singer, L.; Cleary, B.; Figueira, F.F.; Zagalsky, A. The (r) evolution of social media in software engineering. In Proceedings of the on Future of Software Engineering, Hyderabad, India, 31 May–7 June 2014; pp. 100–116.
45. Guzman, E.; Alkadhi, R.; Seyff, N. A needle in a haystack: What do twitter users say about software? In Proceedings of the 24th International Requirements Engineering Conference (RE), Beijing, China, 12–16 September 2016; pp. 96–105.
46. Ali, N.; Kim, S.; Hong, J.E. Listen closely, respond quickly: Enhancing conformity of SPL domain requirements through SNS. In Proceedings of the 2016 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2–4 November 2016; pp. 1–5.
47. Begel, A.; DeLine, R.; Zimmermann, T. Social media for software engineering. In Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, Santa Fe, NM, USA, 7–8 November 2010; pp. 33–38.
48. Kim, Y.; Kim, N.; Lee, D.; In, H.P. Customer Requirements Elicitation based on Social Network Service. *KSII Trans. Internet Inf. Syst.* **2011**, *5*, 1733–1750.
49. Seo, D.M.; Kim, N.H.; Lee, D.H.; Kim, J.D.; In, J.S. Social Network Service-based impact analysis of customer requirements. *Int. J. Multimed. Ubiquitous Eng.* **2012**, *7*, 245–254.
50. Xiao, M.; Yin, G.; Wang, T.; Yang, C.; Chen, M. Requirement acquisition from social q&a sites. In *Requirements Engineering in the Big Data Era*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 64–74.

51. Wen, B.; Luo, Z.; Liang, P. Distributed and collaborative requirements elicitation based on social intelligence. In Proceedings of the 2012 Ninth Web Information Systems and Applications Conference, Haikou, China, 16–18 November 2012; pp. 127–130.
52. Leif, S.; Figueira Filho, F.; Storey, M.A. Software engineering at the speed of light: How developers stay current using twitter. In Proceedings of the 36th International Conference on Software Engineering, Hyderabad, India, 31 May–7 June 2014; pp. 211–221.
53. Storey, M.A.; Treude, C.; van Deursen, A.; Cheng, L.T. The impact of social media on software engineering practices and tools. In Proceedings of the FSE/SDP workshop on Future of software engineering research, Santa Fe, NM, USA, 7–8 November 2010; pp. 359–364.
54. Seyff, N.; Betz, S.; Groher, I.; Stade, M.; Chitchyan, R.; Duboc, L.; Penzenstadler, B.; Venters, C.; Becker, C. Crowd-focused semi-automated requirements engineering for evolution towards sustainability. In Proceedings of the 2018 IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, 20–24 August 2018; pp. 370–375.
55. Ali, N.; Hwang, S.; Hong, J. Your Opinions Let us Know: Mining Social Network Sites to Evolve Software Product Lines. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 4191–4211. [[CrossRef](#)]
56. Ali, N.; Hong, J.E. Creating adaptive software architecture dynamically for recurring new requirements. In Proceedings of the International Conference on Open Source Systems & Technologies (ICOSST), Lahore, Pakistan, 18–20 December 2017; pp. 67–72.
57. Alves, V.; Niu, N.; Alves, C.; Valença, G. Requirements engineering for software product lines: A systematic literature review. *Inf. Softw. Technol.* **2010**, *52*, 806–820. [[CrossRef](#)]
58. Chen, L.; Ali Babar, M.; Ali, N. Variability management in software product lines: A systematic review. In Proceedings of the 13th International Software Product Line Conference, San Francisco, CA, USA, 24–28 August 2009; pp. 81–90.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).