Supporting Requirements Elicitation Practices

Mohd Ilias M. Shuhud[1], Alexander Richter[1], and Aishah Ahmad[2]

[1] Institute for Software Technology,
Bundeswehr University Munich, Germany
{ilias.shuhud,alexander.richter}@unibw.de
[2] Faculty of Computer Science and Mathematics,
Universiti Teknologi MARA, Malaysia
aishah@tmsk.uitm.edu.my

Abstract. In this paper, we examine the practices in requirements elicitation activities from the perspective of a developer of software projects. By doing so, we want to contribute to a better understanding of how the main activities between stakeholders can be supported by IT, particularly social software. We have interviewed six key persons from five different software projects and identified the potential roles of social software to improve in five main activities of requirements elicitation. We present these critical points in the context of the cases and discuss them across the cases.

Keywords: Requirements elicitation, practices, social software.

1 Introduction

Requirements elicitation can be broadly seen as the "*process of identifying needs and bridging the disparities among the involved communities for the purpose of defining and distilling requirements to meet the constraints of these communities*" [1, p. 26]. It is one of the most critical and complex collaborative tasks in software development. It involves stakeholders who benefit from or develop the system, such as end users, developers, legislators, and decision makers [2]. The activities include understanding the application domain, analyzing the stakeholders, choosing appropriate approaches and techniques, and eliciting requirements from identified sources [3]. Conventional methods for identifying requirements include interviews, surveys, focus groups, document analysis, and prototyping. Studies show that collaboration in requirements elicitation is often challenged by the difficulties of establishing a common understanding, implementing effective communication, cultural and business differences, getting the relevant stakeholders on board, ineffective knowledge management, and ineffective conflict management [4, 5, 6, 7, 8]. In recent years, a set of new tools has been developed that seems appropriate to support requirements elicitation activities technically [9]. These new tools, such as blogs, wikis, and social networking platforms—in general termed social software—are said to be capable of facilitating complex task management in collaborative software development [10]. However, there is a lack of studies that focus on how the tools actually support requirements elicitation.

P. Antunes et al. (Eds.): CRIWG 2013, LNCS 8224, pp. 306–321, 2013.

More specifically, a lack of "*understanding of actual work practices led to problematic perspectives of how highly distributed work should be set up in the software industry*" [11, p. 1]. Thus, a careful investigation of the domain characteristics is highly recommended.

With this empirical study, we want to achieve a better understanding of how different stakeholders collaborate in the requirements elicitation process. In the same vein, we would like to examine how this process is currently supported by information technology. We are particularly interested in how existing collaboration practices can be supported by social software and how this support can be extended. Therefore, we have interviewed six key persons from five different software projects. From the knowledge derived in the study, we contribute to a better understanding of how social software can improve collaboration in requirements elicitation. Our research questions are:

1. How do requirements analysts and other stakeholders collaborate in requirements elicitation activities?
2. How can social software support the establishment of a common understanding in requirements elicitation activities?

The remainder of this paper is organized as follows: In the next section, we describe our research methodology. In Section 3, we present the state of the art of stakeholders, main activities, handling elicitation, existing problems, and social software in requirements elicitation from related works. The results from the empirical study are summarized in Section 4. We discuss the results across the cases in Section 5 and present our conclusion and future work in Section 6.

2 Methodology

2.1 Review of Related Studies

We conducted a review of related studies to provide conceptual understandings of the subject matter with respect to the practices of requirements elicitation. We followed the guidelines of [12] for reviewing literature, which consist of five main phases: (1) definition of scope, (2) conceptualization of the topic, (3) a literature search, (4) a literature analysis and synthesis, and (5) the research agenda.

In defining the scope (1), the guidelines follow the work of [13]. We set the focus to observe research outcomes from other studies related to activities in requirements elicitation, the actors of the activities, and problems that exist in the activities. We aim to integrate the outcomes found in the review and organize them conceptually from a neutral perspective. The review is written for general scholars, and it covers only significant sources concerning the topic. In drawing a broad conception (2), we define key terms such as requirements elicitation, stakeholder, problems, and social software. In searching for the relevant literature, some of the keywords used to search (3) were "requirements elicitation," "requirements elicitation collaboration/collaboration tools," "requirements elicitation practices/techniques/methodologies," "problems/issues/challenges in requirements elicitation," and "stakeholders in requirements elicitation." The online databases used include IEEE Xplore, the ACM Digital Library, Springerlink, emeraldinsight, EBSCOhost, SAGE, JSTOR, and Google Scholar. We also did a backward search by referring to literature cited by certain articles, and we received article recommendations from experts in the field (4). Finally, we synthesized the literature and developed insightful questions to construct the agenda for this research study (5).

2.2 Empirical Data Collection and Analysis

The second step was conducting semi-structured interviews. We did six interviews with key informants from five different software projects. We named the sample projects Project A to Project E for confidentiality reasons. Four of the selected projects are from Germany, and one is from Malaysia (Project E). Each project in Germany is represented by a key informant, while two informants from the Malaysian software project participated. The key informants are called I1 (Informant 1), I2, and so on. Semi-structured interview is suitable for understanding subjective theories for everyday knowledge [14]. To ensure that the researcher would address the point of interest, we prepared an interview guide consisting of twenty nine open-ended questions. However, additional questions could be addressed during the sessions to understand related subjects of interest better [14]. Most of the interviews were conducted in English except for one (Project E), which was conducted in both English and Malay. Three interviews were conducted face to face, and five were conducted via Skype Internet telephony. Each interview was recorded and transcribed. The interview transcripts were then sent to the key informants for validation.

We did an interview analysis of the collected data. Codification was performed to support the connotations of the data. We coded the collaboration practices of stakeholders in requirements elicitation based on the transcribed interviews. We used the Atlas.ti version 7 software to assist in managing the empirical data, including transcription, codification and categorization. After the analysis of the transcript, five small case studies were conducted to provide a better picture of the practices from each project in requirements elicitation.

3 Related Studies

In this section, we define and explain the stakeholders and practices involved in requirements elicitation. Next, we focus on the handling of collaboration and problems in requirements elicitation.

3.1 Stakeholders in Requirements Elicitation

In software engineering, stakeholders are "*people or organisations who will be affected by the system and who have a direct or indirect influence on the system* *requirements*" [15]. The term "stakeholder" does not refer to a specific person but indicates the role [2]. The roles of stakeholders in the requirements engineering process can be divided into four main groups: users, developers, legislators, and decision makers. Users can include the employees, managers, suppliers, and customers who will mainly use the system. Developers are responsible for setting requirements definitions and for software development as a whole. Some examples of developers are programmers, designers, requirements analysts, and testers. Legislators produce guidelines or standards that the socio-technical system must comply with and may include certified bodies, public agencies, councils, legal advisories, and security executives [2]. The final group of stakeholders is decision makers, who make decisions about system requirements. Decision makers can come from both the user and developer sides, such as managers, directors, and financial controllers. The identification of stakeholders can be done by considering internal and external parties in the organization [28]. Every stakeholder may have different needs for information systems due to their different roles and work practices.

3.2 Practices in Requirements Elicitation

As the basis to examine the practices in requirements elicitation, we use five main activities as suggested by [3]: understanding the application domain; identifying the source of requirements; analyzing stakeholders; choosing techniques, approaches, and tools to use; and eliciting requirements from stakeholders and other sources. These five main activities are appropriate for our research aims, as it provides a clear indication of the work that requirements analysts and other stakeholders normally conduct in the requirements elicitation process.

Understanding the Application Domain

"*Requirements elicitation is about learning and understanding the needs of users and project sponsors with the ultimate aim of communicating these needs to the system developers*" [3]. To support requirements elicitation activities, it is necessary to understand the application domain. An in-depth investigation of the domain characteristics, including political, organizational, and social elements, as well as constraints on the system, is critical in determining the appropriateness of software projects [16]. Analyzing human and technical factors are critical in strategizing how they can be supported by technology [11]. The information system should be able to support diverse work activities, the types of information associated with those activities, and human-computer interactions. These concerns are in line with the Multiview methodology of [17] because the methodology highlights the importance of including five critical views to be analyzed in software development: (1) analysis of human activity, (2) analysis of information, (3) analysis and design of socio-technical aspects, (4) design of human-computer interfaces, and (5) design of technical aspects. These views are necessary and appropriate in analyzing and designing software systems, as they fulfill both human and technical needs.

Identifying the Source of Requirements

Identifying the source of requirements is an important task in collecting information about the organization and its environment. Besides stakeholders as the primary source of requirements, other reliable sources, such as current systems and processes and documentation (e.g. manuals, forms, and reports), can help analysts to identify the requirements [3]. The information available from these sources enables analysts to understand the current work practices, system process, and problems.

Analyzing Stakeholders

Analyzing stakeholders is critical to determine the right requirements sources. The identification of key users and domain experts is always included in analyzing stakeholders [3]. Relevant stakeholders can be internal and external to an organization. There are four steps in eliciting basic requirements from the identification of stakeholders [18]: stakeholders need to be identified and prioritized according to their roles and levels of influence; each stakeholder's profile will be collected to allow further understanding of the stakeholder; the prediction of requirements will be done based on learned profiles; and the requirements must be prioritized.

Selecting Techniques, Approaches, and Tools

Techniques such as interviews, observation, task analysis, workshops, and prototyping are among the common techniques used by software practitioners [16, 3]. Depending on the project type, some may tend to use more than one technique. According to [3], interviews, domain analysis, and group work are the most appropriate in determining the major practices in requirements elicitation. Additional approaches such as those based on goals, scenarios, viewpoints, and domain knowledge are also employed in some projects, though interview and group work are still the most popular techniques.

Eliciting Requirements from Stakeholders and Other Sources

After the application domain is well understood, sources of requirements are identified, relevant stakeholders are analyzed, and techniques or approaches have been selected, the elicitation of software requirements will begin [3]. At this stage, the scope of the system will be defined and a detailed extraction of users' needs will be performed [19]. Effective communication between analysts and other stakeholders is critical in this stage. The information gathered will be used in the next stage of requirements engineering, which is requirements specification, where the functionalities will be finalized.

3.3 Handling Requirements Elicitation

In handling collaborative work during requirements elicitation, the objectives of the collaboration should be clearly stated and well structured. For instance, [3] proposes three phases of collaboration in requirements workshops: (1) the scoping phase, (2) the high-level phase, and (3) the detail phase. Scoping is the process by which appropriate stakeholders are identified, problems are described, and the mission and vision are agreed upon. In the high-level phase, using the results of the scoping phase, relevant documentation will be reviewed and related work practices and system operation will be observed to give the analyst a basic understanding of the work domain, the software requirements, and the software project direction. In detailed workshops, analysts refine the results from earlier phases and validate them with the stakeholders. Each determined work practice is decomposed into individual process using use case analysis, for example. "*Documentation using natural language with graphical description should iteratively improve and incrementally develop during all phases*" [20]. This practice is important so that the document can be validated and agreed upon by all stakeholders to reach a common understanding. Furthermore, the final document can be a useful reference in the next stage, requirements specification, where functional requirements are determined.

3.4 Problems in Requirements Elicitation

Problems and challenges can affect the quality of requirements. The most common problems identified in previous work are problems of scope [5], communication and common understanding, e.g. [4], domain knowledge [16, 15, 21], and stakeholders [3]. The final software requirements are the results of collaboration among stakeholders, which is challenged by conflicts and contradictions among them [22].

The process of requirements elicitation is not only associated with technical issues but also involves social and communication issues among the stakeholders of the project, who play a significant role in the requirements elicitation process [5, 23]. Several studies have highlighted the importance of maintaining effective communication between stakeholders and the development team in generating quality requirements, e.g. [21]. Further, communication skills and the relationships between stakeholders determine the quality of the requirements [16] because the requirements will be documented and further addressed in the subsequent phases of development. Effective communication is the answer to the need to establish a common understanding of requirements elicitation. Without a common understanding, various conflicts can arise among stakeholders and may result in fluctuation in the requirements due to different interpretations of them. It has also been found that the main point of communication during the software requirements phase is to discuss changes in requirements [6]. In distributed software development, this problem is even more challenging, as the participants are not in the same location or time zone and do not have the same language skills and culture.

The barriers to effective communication in requirements elicitation include the effectiveness of existing means of interaction, terminology (e.g. software jargon), team size (e.g. too few or too many stakeholders), lack of awareness, limited knowledge flow, and social and organizational issues [8]. Further, [8] suggests that documentation (e.g. specification documents) is a poor communication channel and that relying too much on it will only widen the gap between the analysts and other stakeholders. As mentioned earlier, geographical distance also can affect the quality of communication among stakeholders, especially in distributed teams [7]. However, the effective implementation of collaborative tools can assist in addressing the communication quality. Finally, informal communication can bridge the gap between software developers and other stakeholders in defining requirements [8].

3.5 Social Software in Requirements Elicitation Collaboration

Effective and quality communication tools are essential to ensure successful software development collaboration [24]. It is claimed that informal communication and asynchronous communication can strengthen the relationship between the developer and other stakeholders [6]. At present, a growing number of studies (e.g. [9, 25, 18, 24, 26]) are observing the use of a new type of application called social software such as Facebook, Twitter, wikis, and blogs, to mediate collaboration in software development. This new form of Web applications, which has remarkably changed people's informal communication [31, 32], is becoming more significant in software development processes. For example, [24] attempts to develop new social networking platforms to facilitate requirements engineering collaboration for distributed teams. Meanwhile, in [18], a social networking platform is developed to improve the stakeholder identification process, predict possible requirements, and prioritize them. Both [24] and [18] claim that their platform has been tested in several projects and received promising feedback from users. Notwithstanding these first insights, further research must be done to establish the real potential of social software in this field. We need to understand better how they support the requirements elicitation process so that we can improve collaboration.

4 Summary of the Cases

Five software projects are involved in the empirical study. Projects A, B, C, and D are funded innovative projects in Germany, while Project E is a commercial project in Malaysia. All of the informants interviewed are key persons in the software projects, such as project managers, technical coordinators, business analysts, and technical leaders, who are directly involved in requirements elicitation processes.

The main objective of Project A was to build a community platform within a large-scale corporation. It was carried out by three universities and two leading companies in Germany. The universities' stakeholders consist of researchers, including professors, post-doctoral researchers, and doctoral researchers from three disciplines: psychology, information science, and information systems. The participants from the two partner companies include the members of the board of directors, various operation and management executives, and selected employees.

Project B is a software project to develop a social networking platform for military medical officers and trainee officers at a university. The administration of the medical branch appointed a research institute at a university to steer and develop a social networking platform to address the issue. The institute is represented by researchers who will manage and develop the platform. They require strong support from medical officers and trainee officers to gather requirements.

Project C is the development of a mashup solution that collects data from social media services such as Facebook and Twitter with certain filters. The data generated by the mashup can be used by other software applications, such as tablet computer or smartphone applications. The interface between the mashup and software applications is also among the elements that the project must address. At the moment, there are three applications of other projects that use the mashup service. This is a service-oriented architecture project.

The mission of Project D was to develop a solution to support elderly people's daily life after demographic developments. The objective is to lower the barrier of access to the benefits of social software for elderly people with a specially designed tablet computer based on the Android operating system. The stakeholders of the project consist of developers of different aspects (e.g. the hardware design, the human-computer-interaction, and the operating system) from ten partners of four different countries (Germany, Spain, Italy, and Austria). Also involved is a partner who wrote the business plan, a partner who conducted project coordination, and elderly people from Spain and Germany.

Project E is the enhancement of a retail collection module in a banking information system for a Malaysian bank that was affected after the restructuring of the Retail Collection Center. The module is used by more than 200 employees who make collection calls and conduct recovery processes and legal actions nationwide. It is also used by managers for managerial and reporting purposes. Recently, the bank's management decided to centralize the whole nation's retail collection in one center. The restructuring directly affected the existing information systems, including the module to be implemented. Table 1 below summarizes all the studied projects; however, we do not discuss their actual elicited requirements.

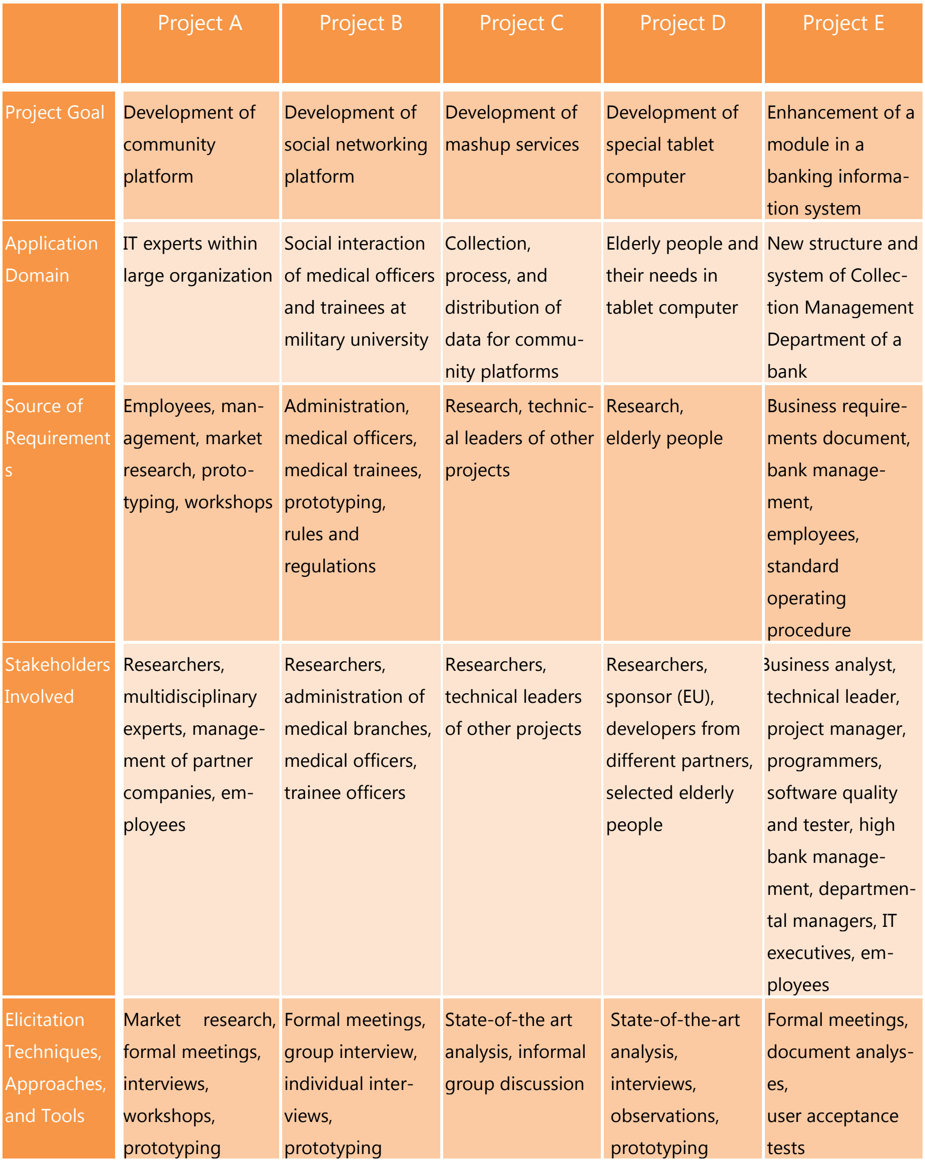
5 Discussion across the Cases

In this section, we compare the results across the cases to reveal insights concerning practices by stakeholders in the requirements elicitation process. We analyze the cases based on the five activities summarized by [3]. We will propose potential roles that can be played by social software, and we further address the concept for a prototype in Section 6.

5.1 Understanding the Application Domain

There are different motivations in understanding the application domain among software projects. For instance, in innovative projects such as Projects A, C, and D, the process of understanding the application domain was initiated by researchers after a certain period of analysis inspired by research areas of interest and before executing a project. The application domains were studied in detail, supported with certain theories and reviews of related work.

Table 1. Summary of the Cases



	Project A	Project B	Project C	Project D	Project E
Project Goal	Development of community platform	Development of social networking platform	Development of mashup services	Development of special tablet computer	Enhancement of a module in a banking information system
Application Domain	IT experts within large organization	Social interaction of medical officers and trainees at military university	Collection, process, and distribution of data for community platforms	Elderly people and their needs in tablet computer	New structure and system of Collection Management Department of a bank
Source of Requirements	Employees, management, market research, prototyping, workshops	Administration, medical officers, medical trainees, prototyping, rules and regulations	Research, technical leaders of other projects	Research, elderly people	Business requirements document, bank management, employees, standard operating procedure
Stakeholders Involved	Researchers, multidisciplinary experts, management of partner companies, employees	Researchers, administration of medical branches, medical officers, trainee officers	Researchers, technical leaders of other projects	Researchers, sponsor (EU), developers from different partners, selected elderly people	Business analyst, technical leader, project manager, programmers, software quality and tester, high bank management, departmental managers, IT executives, employees
Elicitation Techniques, Approaches, and Tools	Market research, formal meetings, interviews, workshops, prototyping	Formal meetings, group interview, individual interviews, prototyping	State-of-the art analysis, informal group discussion	State-of-the-art analysis, interviews, observations, prototyping	Formal meetings, document analyses, user acceptance tests

The business and user needs must be carefully defined before the requirements elicitation for their software takes place. However, this type of project requires adequate support from their partners to establish an equal understanding of the project proposed to benefit them. In Project A, the developer team conducted market research to identify the status quo, including understanding which solutions are well accepted by people. Meanwhile, Project C requires more technical knowledge; the application domain is around the mashup itself and the people who needed its services. Therefore, besides a theoretical understanding, Project C involves a lot of collaborative programming work among the team to understand how the solution will work. In Project D, the project team used interviews and observation of tablet usage by elderly people. They learned about the elderly people's perceptions toward and needs in a tablet computer.

> "*First of all, we did market research… What kind of communities are up there? What kind functions do they support? So we can have a market overview of what is out there and what is used and what users really accept. Because sometimes you cannot say from scratch which kind of functions users will accept. It's really hard to tell...*" —I2

Project B was initiated by the administration of the medical branches at a military university, who perceived that the passive social interaction among the medical staff members and trainees need to be resolved. The appointed software institute is developing a solution with a social networking platform. The team learned about the application domain through individual and group interviews with medical officers and trainees introduced by the administration office. Similarly, in Project E, a commercial project, the process of understanding the application domain was motivated by a request from the client or user side (bank) through a document called "business requirements" to the software vendor. The business analyst (or requirements analyst) from the vendor side is the champion for the application domain knowledge. In general, the understanding of the application domain in a software project can be initiated on the developer side or the client side. The developer has to possess the most accurate understanding of the application domain, as the knowledge derived will be used as a foundation for the next steps. However, the rest of the stakeholders can also contribute. In relation to that, we find that knowledge obtained in this stage should be easily accessible by all stakeholders to minimize conflicts and stimulate a common understanding [16, 21, 6]. The knowledge derived from fieldwork such as interviews and meetings or from state-of-the-art analysis such as extensive readings and reviews can be stored in a platform like a wiki. The other stakeholders who are not collecting the information can obtain knowledge from the wiki. Such a platform is actually exists in Projects A, B, C, and D. However, the platform is only actively used in Projects B and C, and they find that the platform is very useful in promoting shared understanding among stakeholders.

5.2 Identifying Source of Requirements

In identifying the sources of requirements, we discovered in some projects that this activity was conducted in parallel with the understanding of the application domain. Besides state-of-the-art analysis (for innovative projects), the sources of requirements can be identified through discussion with the decision makers during early-stage meetings. For example, Projects A, B, and E received lists of who would be participating in the projects. Detailed profiles of the different types and backgrounds of users and documentation also can be obtained. For Project D, the main source of requirements is elderly people from Spain and Germany, whom the project team observes and interviews to gather requirements. In the case of Project C, we learned that the source of requirements depended on other software projects that use the mashup services to provide new requirements from time to time. In Project E, as mentioned in section 5.1, the software vendor received a business requirements document from the top management of the bank. From this source, the development team can predict other potential sources of requirements, such as the managers at the Credit Management Department, the end users, and affected software module (collection).

> *"We used the contacts at the administration of the medical branch. They were able to tell us the key persons we should ask, and then we did a bit of background research and asked them about the product contacts"* —I2

For this activity, we observe that the potential role of social software is to gather the sources in one space and communicate them. All stakeholders of the project should know where requirements come from, as they can contribute by proposing other potential sources to be taken into account, such as additional contacts (e.g. end users and managers) and documentation (e.g. business requirements, reports, forms, standard procedures, guidelines, and manuals), as highlighted by [3].

5.3 Analyzing Stakeholders

In analyzing stakeholders, most developers, except for those of Projects C and D, collaborate with their partner or client to identify relevant stakeholders, especially the actual users. In the beginning, lists of users or employees are provided by the management, but the developer can examine the list and then request more relevant stakeholders when necessary.

> *"...we also got detailed profiles of certain user types; for example, many of our medical officers are not used to using Internet technology... we select the most suitable users..."* —I2

For Project C, the user stakeholders and the main source of requirements are the project managers or technical coordinators of software projects that are using Project C's solution (mashup). Thus, it is very easy for this project to identify and analyze the relevant stakeholders. Meanwhile, Project D has to determine suitable elderly people as the users. They finally selected about 20 people from Spain and about 20 from Germany to participate in the tablet computer testing. Identifying partner developers is a challenging task, especially in selecting an interactive designer for the tablet. As shown in [3], the stakeholders can be internal or external to the organization, which shows that stakeholders may come from outside or have indirect involvement. Generally, each surveyed project has all the stakeholders defined by [2]. However, we did not clearly identify the legislative stakeholders, especially for Projects A, C, and D. However, according to the work, a legislative stakeholder may also be in the form of documents, such as operation and quality manuals. For example, Project C has to ensure data privacy under military rules, and Project D has to follow new Central Bank guidelines.

> *"There are new guidelines from the Central Bank that we were told by them (bank's management) to follow, for example, about check management. A copy of a check received by a customer must be sent to them in a data file..."* —I5

Based on the practices involved in analyzing stakeholders, we believe that the list of relevant stakeholders with their details should be available in a space that other stakeholders of the project can see. Knowing who is participating will help in coordinating the collaboration, especially for projects that involve external stakeholders and stakeholders in different locations, as in Projects A and D [6]. It can also helps in predicting requirements based on their profiles [18]. Legislative stakeholders of non-human form, such as policies or regulations, should be accessible by the project stakeholders in the space. Here is where social software can play a role. We will address this issue further in Section 6.

5.4 Selecting Techniques, Approaches, and Tools

All of the software projects except for Project E decided to use interviews along with other techniques, such as observations, workshops, formal meetings, group discussion, and prototyping, to elicit requirements. Several prototype versions were created, distributed, and tested among user stakeholders in Projects A and D. Project A began with a workshop to provide an overview of the project. Meanwhile, Project E mainly used a series of formal meetings with the management and the IT department of the bank to gather the requirements. Apart from that, documents such as the new standard operating procedure for the new structure of the Credit Management Department will be analyzed in detail. Test cases were prepared for software testing and quality assurance purposes.

> *"We did a lot of interviews, and we didn't follow a clear top-down communication; we were communicating at different levels. We have one guy at XY who was an employee doing the IT management systems before, and he has explained his experience with the previous system."* —I1

In terms of collaboration tools, all software projects except for Project E have a wiki platform, but as we mentioned before, only Projects B and C are actively utilizing the platform especially for progress update. Informants from Projects B and C claimed that a wiki is very useful in allowing stakeholders to keep up with the latest updates on work in progress from the side of the developer and the other stakeholders. In relation to that, informants from all the projects spoke about the use of email as the main asynchronous communication tool and stated that Skype is the most frequently used synchronous tool in discussion.

In this activity, we proposed that the selected techniques, approaches, and tools to elicit requirements should also be mentioned to all stakeholders because the objectives of any selected techniques, approaches, and tools should be clearly stated and well-structured to achieve the goals [3]. The information can be included in a wiki by describing, for example, why a certain workshop format was selected, how the workshop would be conducted, who would be participating in the workshop, and what is expected from the workshop.

> *"… you have to have the overall goal that everyone has to understand. Otherwise, you do some minor or very detailed changes or you try to develop some requirements at a very detailed level, and later they don't matter because they don't understand; they don't fit the overall idea. So what we did at first was really to get the overall idea so that everyone understood this basic idea in the kickoff event and workshops."* —I1

5.5 Eliciting Requirements from Stakeholders and Other Source

In this activity, active interaction between the developer and the identified sources of requirements will begin to determine the actual and accurate needs. In Project D, for example, the developer conducted a direct observation of elderly people in the use of conventional tablet computer, followed by open interviews at a convenient location like a restaurant. The information gathered will be analyzed and translated into requirements. When a prototype of the tablet computer has been developed, it will be tested by elderly people to obtain feedback or new requirements. This iterative process will continue until the users are happy with the special tablet computer. The process in Project A is nearly identical, as different levels of prototypes for the community platform were distributed to gather the requirements from selected employees at partner companies. Project B elicited requirements through interviews and extraction of the problems that can be solved using the social networking platform.

In Project E, the elicitation of requirements from formal meetings is done in collaboration among a business analyst, a technical leader, and a project manager. After analyzing the business requirements, the business analyst will consult with the technical leader and the project manager to decide whether the request is feasible to implement. To verify and validate the requirements, several meetings were held with the bank side and included the sponsors, managers from the credit management department, and people from the IT department. Later, a technical team (e.g. programmers, systems analysts, and quality analysts) will be formed to finalize the requirements until they are signed off on.

However, some issues were reported during the elicitation process. In Project B, it is reported that the idea of open communication on a social networking platform is in conflict with the military rules. Thus, some modifications have to be made to follow the rules. In the case of Project E, the problem is inconsistency in explaining certain requirements by different representatives. A series of meetings was arranged during the requirements elicitation, but managers who are participating in the project sometimes failed to attend due to other commitments and sent someone else. However, I5 sometimes found that there were conflicts between requirements provided earlier and the most recent versions. Moreover, verifying requirements via email with the original manager who provided the requirements sometimes takes a long time to get a response. This issue can cause delays in the development work. The involvement of people from the management can sometimes ease the confirmation of requirements but can also slow down the process. Other than the domain knowledge issue, communication breakdown also contributes to this problem [16]. After looking at the practices and issues from the case, we proposed that elicited requirements should be published in one space so that all the stakeholders can read them and respond to them. Communication issues and inconsistency in requirements can be reduced when every stakeholder has the same access to the elicited requirements. If other information, such as a list of stakeholders and sources of requirements, is also available in the space, it can minimize conflicts between requirements and other variables.

> *"…we have a problem in finalizing requirements; the managers sometimes have other commitments, and they are unable to attend the meetings. Although they have representatives, the information sometimes is not consistent and confuses us [….] we can reconfirm and get clarification, but this takes some time."* —I5

In summary, we propose that the outcomes from all five activities in requirements elicitation, which are mainly performed by the developer, are useful to extend to other stakeholders. The shared outcomes from requirements elicitation activities among stakeholders can stimulate a common understanding among them. As a result, collaborative issues in the requirements elicitation process and the later stages can be reduced. This is where the potential role of social software lies in establishing a common understanding among different stakeholders in requirements elicitation. In the final section, we will briefly address our future work to develop a wiki prototype to extend the outcomes of this research.

6 Conclusion and Future Work

In this paper, we examine the practices of requirements elicitation activities. Five software projects were studied to gain insights from the main activities identified by [3]. Conducting these activities is the role that requirements analysts or business analysts play. However, we propose that the outcomes from the activities should be shared among other stakeholders to establish a common understanding of the requirements and that social software has the potential to support this. It was reported that most of the projects have their own IT in supporting their entire project management; however, the manipulation of the technology for requirements elicitation is limited. In addition, some projects did not fully utilize their social software platform. To extend the results of this research, we will develop a wiki prototype for requirements elicitation in our future work.

The wiki should collect and communicate the outcomes from the five main activities in [3], making them accessible to all project stakeholders. For instance (recap from 5.1 to 5.5), in understanding application domain, the activity will generate knowledge like work environment, processes, standard procedures, routines, issues and challenges within the application domain. In addition to that, the current or final list of elicited requirements with descriptions will produced after requirements elicitation process. We propose that, these outcomes from five main activities should be shared through a wiki platform so that it can assist in establishing common understanding among stakeholders.

In this research study, we have a limited number of interventions. For example, we have only one sample of a commercial project and only one informant interviewed for each software project except for Project E, which has two informants. We will ensure more samples in future data collection for our future work.

References

1. Software Engineering Institute, Requirements Engineering and Analysis Workshop Proceeding, Software Engineering Institute, Pittsburgh (1991)
2. Sharp, H., Finkelstein, A., Galal, G.: Stakeholder Identification in the Requirements Engineering Process. In: 10th International Workshop on Database & Expert Systems Applications, DEXA (1999)
3. Zoughi, D., Coulin, C.: Requirements Elicitation: A Survey of Techniques, Approaches and Tools. In: Engineering and Managing Software Requirements, pp. 19–46. Springer, Heidelberg (2005)
4. Coughlan, J., Lycetter, M., Macredie, R.: Communication Issues in Requirements Elicitation: A Content Analysis of Stakeholder Experiences. Information and Software Technology 45(8), 525–537 (2003)
5. Christel, M., Kong, K.: Issues in Requirements Elicitation (CMU/SEI-92-TR-012) (retrieved April 06, 2013), from the Software Engineering Institute, Carnegie (2013)
6. Damian, D., Marczack, S., Kwan, I.: Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centered Social Networks. In: 15th IEEE International Requirements Engineering Conference, IRE, pp. 59–68 (2007)
7. Damian, D., Zowghi, D.: Requirements Engineering Challenges in Multi-site Software Development Organizations. Requirements Engineering Journal 45(8), 149–160 (2003)
8. Al-Rawas, B., Easterbrook, S.: Communication Problems in Requirements Engineering: A Field Study. In: Westminster Conference on Professional Awareness in Software Engineering (1996)
9. Begel, A., Deline, R., Zimmermann, T.: Social Media Afor Software Engineering. In: Workshop on the Future of Software Engineering Research (2010)
10. Treude, C.: The Role of Social Media Artifacts in Collaborative Software Development. PhD Thesis, University of Victoria (2012)
11. Avram, G., Wulf, V.: Guest Editorial: Studying work practices in Global Software Engineering. Information and Software Technology 53(9), 949–954 (2011)
12. vom Brocke, J., Simons, A., Riemer, K., Plattfaut, R., Cleven, A.: Reconstructing the Giant: On the Importance of Rigor in Documenting the Literature Search Process. In: 17th European Conference on Information Systems (2009)
13. Cooper, H.: Organizing Knowledge Syntheses: A taxonomy of literature reviews. Knowledge in Society 1, 104–126 (1988)
14. Flick, U.: An Introduction to Qualitative Research, 4th edn. Sage Publication, London (2009)
15. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. John Wiley, Chichester (1998)
16. Marnewick, A., Pretorius, J.H., Pretorius, L.: A Perspective on Human Factors Contributing to Quality Requirements: A Cross-case Analysis. In: IEEE International Conference on Industrial Engineering and Engineering Management, IEEM (2011)

17. Avison, D., Wood-Harper, A.: Multiview: An Exploration in Information Systems Development. MGrawHill (1990)
18. Lim, S.L., Finkelstein, A.: StakeRare: Using Social Network and Collaborative Filtering for Large-scale Requirements Elicitation. IEEE Transactions on Software Engineering 38(3), 707–735 (2012)
19. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A roadmap. In: Conference on The Future of Software Engineering, ICSE (2000)
20. Coulin, C., Sahraoui, A.E.K., Zowghi, D.: Towards a Collaborative and Combinational Approach to Requirements Elicitation within a Systems Engineering Framework. In: International Conference on Systems Engineering (2005)
21. Damian, D., Chisan, J.: An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management. IEEE Transactions Software Engineering 32, 433–453 (2006)
22. Damian, D., Kwan, I., Marczak, S.: Requirements-Driven Collaboration: Leveraging the Invisible Relationships between Requirements and People. In: Collaborative Software Engineering, pp. 57–76. Springer, Berlin (2010)
23. Panjer, L., Damian, D., Storey, M.-A.: Cooperation and Coordination Concerns in A Distributed Software Development Project. In: International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE (2008)
24. Hannemann, A., Hocken, C., Klamma, R.: Community Driven Elicitation of Requirements with Entertaining Social Software. In: International Workshop on Software Engineering within Social Software Environment (2009)
25. Storey, M., Kwan, C., Van Deursen, A., Cheng, L.: The Impact of Social MEdia on Software Engineering Practices and Tools. In: Workshop on Future of Software Engineering Research (2010)
26. Duarte, D., Farinha, C., da Silva, M., da Silva, A.: Collaborative Requirements Elicitation with Visualization. In: IEEE 21st International Workshop onEnabling Technologies: Infrastructure for Collaborative Enterprises, WETICE (2012)
27. Bougie, G., Starke, J., Storey, M., German, D.: Towards Understanding Twitter Use in Software Engineering: Preliminary Findings, Ongoing Challenges and Future Questions. In: 2nd International Workshop on Web 2.0 for Software Engineering (2011)
28. Sutcliffe, A.: Collaborative Requirements Engineering: Bridging the Gulf between the Worlds. In: International Perspectives on Information Systems Engineering (2010)
29. Lohmann, S., Dietzold, S., Heim, P., Heino, N.: A Web Platform for Social Requirements Engineering. In: International Workshop on Software Engneering within Social Software Environment (2009)
30. Johansen, R.: GroupWare Computer Support for Business Teams. The Free Press, New York (1988)
31. Richter, A., Riemer, K.: Corporate Social Networking Sites – Modes of Use and Appropriation through Co-Evolution. In: 20th Australasian Conference on Information Systems, ACIS (2009)
32. Richter, A., Koch, M.: Functions of Social Networking Services. In: 8th International Conference on the Design of Cooperative Systems, Coop (2008)