2004

Requirements Engineering During Global Software Development: Some Impediments to the Requirements Engineering Process - A Case Study

Jo Hanisch
University of South Australia, jo.hanisch@unisa.edu.au

Brian J. Corbitt
Deakin University, bcorbitt@deakin.edu.au

REQUIREMENTS ENGINEERING DURING GLOBAL SOFTWARE DEVELOPMENT: SOME IMPEDIMENTS TO THE REQUIREMENTS ENGINEERING PROCESS – A CASE STUDY

Hanisch, Jo, School of Accounting and Information Systems, University of South Australia, City West Campus, 70 North Terrace, Adelaide, SA, jo.hanisch@unisa.edu.au

Corbitt, Brian, Deakin University, 221 Burwood Highway, Melbourne, Victoria 3215, bcorbitt@deakin.edu.au

Abstract

Requirements engineering is not straightforward for any software development team. Developing software when team members are located in widely distributed geographic locations poses many challenges for developers, particularly during the requirements engineering phase. This paper reports on a case study concerning a large software development project that was completed in just seven months between users located in the UK and software developers from an international software house based in New Zealand. The case indicates that while "true" global requirements engineering may be desirable in achieving economy of resources, a "hybrid" structure of requirements engineering processes is more realistic so that lasting relationships with clients may be formed, and requirements engineering activities achieved. The main impediment to the process of requirements engineering during global software development, as recounted by the team members in this case, is communication. Communication issues may be further described in terms of four categories: distribution of the clients and the development team, distribution of the development team, cultural differences between the clients and the development team and cultural differences among the development team

Keywords: global software development, requirements engineering, hybrid global virtual teams

1 INTRODUCTION

Requirements engineering (RE) is an important phase in the development of information systems. RE has enjoyed many years of research attention, but mainly centring on the technical aspects of the phase, with the development of methods which capture and process users' requirements (Zave 1995). However, Thanasankit (1999a) and Hanisch, Thanasankit and Corbitt (2001) have considered the cultural and social aspects of RE, and have found some challenges including: developing trust between team members and their client; accounting for communication preferences; and sensitivity to the ways various cultures work. More recently, Damian and Zowghi (2003) have focused on RE during global software development.

Issues that appear most frequently in the emerging literature concerning global software development, include loss of communication richness (Jarvenpaa and Leidner 1998; Carmel 1999; Battin, Crocker and Kreidler 2001), cultural differences (Carmel 1999; Herbsleb and Moitra 2001), loss of identity with the team (Karolak 1998; Carmel 1999) and lack of management support (Karolak 1998). When RE is undertaken as part of global software development these issues may be exacerbated by the dynamic nature of the process of RE as well as the social and cultural aspects associated with gathering and managing requirements.

The main purpose of this paper is to explore communication issues associated with the RE phase of the software development process as it occurs in the virtual domain. This paper focuses on one case study concerning a large software development project that was completed in just seven months between users located in the UK and software developers from an international software house based in New Zealand. The global software development team in this research experienced some similar communication and coordination issues to those in previous literature. However, this paper contributes to the literature by focusing on and describing in-depth the communication issues associated with RE during global software development. The impediments to the processes of RE in this case highlight the need for understanding of the software development project environment and the ways in which business can address the complex activities of RE.

This paper first reviews global software development and RE during global software development, followed by the discussion, which is centred on a case study involving the RE processes in a dispersed team as part of a New Zealand Software House. The paper does not deal with problems and solutions for distributed teams, but rather considers the RE phase with a focus on communication issues.

2 GLOBAL SOFTWARE DEVELOPMENT

As recently as 10 years ago, virtual work consisted mainly of simple administrative tasks for teleworkers who worked from home and who mainly used email and word-processed documents (Staples 1996). These type of tasks required little collaboration and co-operation and there were few projects requiring on-going teamwork. Within the past 5 years, Karolak (1998) and Carmel (1999) describe the concept of global software development, where the need to solve more complex tasks occurs in the virtual domain.

Ramesh and Dennis (2002:1) consider "global virtual teams are teams whose members share a common purpose and are located in at least two different countries," and global software development teams are just one type of global virtual team. Global software development requires teamwork and the co-operation and collaboration of team members who use electronic communication media to explain and communicate complex concepts. According to Damian and Zowghi (2003:319), "software engineering is witnessing a transition from traditional co-located form of development to a form in which global software teams collaborate across national borders." The use of global software development teams has increased rapidly due to the limited pool of trained workforce; the necessity to locate specific expertise close to the customer; the differences in development costs and the promise of round-the-clock development (Carmel 1999; Mockus and Herbsleb 2001).

Most studies concerning global software development have addressed technical dimensions of meeting systems or tools such as CASE (Damian, Shaw and Gaines 2000). While certain formalised communication and collaborative technologies exist for global software development, their uptake has been slow and sporadic, and we consider that this may be due to the influences in the software development project environment. One key factor is the influence of electronic communication tools on the social and cultural processes of software development. Carmel (1999) suggests that success in global software development occurs when rigour is imposed on the team. This requires greater discipline but compensates for the loss of informal communication which allows developers "to get the job done" as structure is imposed on the team. Structure imposed on the global software development team may in fact decrease the social and cultural exchange between the team members. Carmel (1999) argues this is necessary to ensure strict change management, sign-off procedures and reporting mechanisms.

3 RE DURING GLOBAL SOFTWARE DEVELOPMENT

The development of an "effective IS requires thorough analyses of user information needs prior to IS design" (Byrd et al 1992:117). RE, which is concerned with understanding the needs of the client (user) and determining the systems requirements which satisfy these needs, given any identified constraints and exclusions (Carroll and Swatman 1997), is a crucial phase in the software development process (Greenspan et al 1994). Failure of many IS development projects is due not just to the inadequate requirements (Boehm 1981) in general, but more specifically to the social, political and cultural factors associated with the project (Goguen and Linde 1993). Within the social context, it is necessary for IS specialists to understand communication and cooperation, as well as social complexity during RE (Thanasankit 1999b). RE research has traditionally been positivist in its approach, largely focusing on the methods used for elicitation, specification and validation of requirements. Many researchers conclude that the more structured the methods used, the more likely that the requirements will be clearly defined and understood. However, as Thanasankit (1999a) argues, organisations need to go further and consider the emotions and culture of users and IS specialists.

When considering RE during global software development, Damian and Zowghi (2003:320) state "inadequate communication in global structures creates most challenges" and contributes to many underlying problems. Further they suggest that unless the four major aspects of cultural diversity, inadequate communication, knowledge management and time differences, are addressed in global software development, the stakeholders will face difficulties in RE practice. According to Land and Somogyi (1986) there is an interaction between formal systems and their environment, and this may explain why different management approaches, tools and techniques are needed for different system types and environmental conditions. As Hanisch et al (2001) suggest the appropriate choices of tools, techniques and approaches may help to improve the elicitation of requirements and the chances of success of global software development.

3.1 Collaboration and Communication

Zack (1993) indicates the main distinction between collocated and global teams is their mode of communication. Communication is an important managerial issue for global teams. Not only are the team members required to make use of computer-based communication technologies for everyday project tasks, managers are also required to exhibit leadership, track performance and solve complex software development problems through the use of computer-based communication. RE, which requires a higher degree of communication than the other phases in systems development, involves communication and collaboration that is more complex in global teams. As Mockus and Herbsleb (2001:182) consider, problems occur with requirements changes in global software development because "it is hard for the formal mechanisms of communication, such as specification documents, to react quickly enough."

Choosing the appropriate communication tools and techniques for RE in the virtual domain is problematic, because the activities of RE often require communication-rich media to analyse the requirements and present creative solutions. Most electronic communication media, such as email, is recognised as the antithesis of

communication-rich. There may occur a dilemma for project managers who are responsible for the RE phase during global software development. Structured software development methods may assist requirements engineers in gathering requirements in the virtual domain, however these methods may also inhibit the social aspects of virtual communications (Hanisch 2001), and this may cause misinterpretation and miscommunication of the requirements. The most appropriate choices of management approaches, tools and techniques as recommended by Land (1998) may not be available in the virtual domain.

3.2 Communication and Separation of Clients/Developers

Communication issues for RE during global software development may be considered from two perspectives, including: the separation of the clients from the development team members, and the separation of developers from each other. Within these separations there are further factors influencing the RE processes, such as cultural differences. Communication issues relating to the separations of the team members from each other and their clients are discussed in this paper, adding further to the understanding of the influences and processes of RE during global software development.

4 RESEARCH METHODOLOGY

The aim of this research is to explore the processes of RE during global software development with emphasis on the communication issues that arose during the project and some solutions used by the company to address those issues. This paper explores the issues and impediments to RE during global software development from a social and cultural perspective. The research question framed for this research is: How are the processes of and communications during RE influenced by working in global virtual teams? In order to address this question, a global project team consisting of the project manager, three system architects and two system developers from a New Zealand Software House were interviewed in New Zealand during December 2001 – January 2002.

The interviewees were selected based on their responsibilities in gathering requirements from clients, who were engaged in interviewing users, and who were observing users' activities, and gathering documents to construct requirements for development of the system. This comprised the entire project team, excluding the users. It was not considered appropriate to interview the users for two reasons: they were not under line control of the development company's project manager, and hence not considered part of the development project team; and there were on going contractual arrangements between the user's company and the Software House which could not be jeopardised.

The principal method for collecting data was by taped in-depth interviews, which lasted from between 1.5 to 4 hours. The questions were open-ended and team members had freedom to describe their experiences and problems beyond the questions' boundaries. Further, the researchers were provided access to all documentation concerning the project, as well as a "live" demonstration of the final software product. The interviews were transcribed and data from all sources were analysed using typical case study techniques of themes, descriptions and assertions as detailed in Creswell (1998). Follow-up email and telephone calls after the interviews provided clarification when necessary. The conclusions reported in the following analysis of the case study data represent the authors' interpretations of the evidence. The case study is applied to the conceptualisations outlined above accepting that it is only one case study, albeit in some detail, and that it represents only one sphere of influence.

5 SAPPHIRE SOFTWARE HOUSE* CASE STUDY

Sapphire Software House was chosen for this research because the company recently used a global project team to develop and implement an international information system. This case study covers the development and implementation of an object–oriented system between Sapphire Software House, (an international software company based in New Zealand), the users from a large rail company located in the UK and the users' IT group which was also situated in the UK. Hence, the project team for a significant part of the project lifecycle were working in a virtual domain.

Sapphire Software House, founded in 1978, is a privately owned international software company with approximately 400 employees that has developed a suite of enterprise application development solutions. Sapphire offers application service provider capability where an application is rented, or enterprise service provider facilities where the entire business application is managed for a client. Sapphire's main product combines the robust nature of mainframe technology with the flexibility and low cost advantages of commodity hardware and the Internet.

Sapphire was contracted by the users of this project to develop an on-line ordering system for the rail industry in order to streamline the ordering, planning and monitoring of relevant trains. Sapphire had not been involved in the development of applications for the rail industry previously, but the company was chosen for the contract based on the type of development solutions they produce and the speed in which they agreed to deliver the project. The existing system did not provide an efficient method of tracking orders in progress due to the manual nature of the planning process and as the information was split across many disparate spreadsheets, querying and reporting on information was time-consuming.

In summary, the project was instigated to address the following issues:

- The train order management process was complex and time consuming;
- Ordering of trains occurred through many different approaches, (for example email, fax and phone), which required coordination and manual intervention;
- Large numbers of different technology components existed;
- The inflexibility of the system made no allowances for last minute changes; and
- Information management was difficult due to lack of an integrated database for queries and reports.

The system that Sapphire undertook to build consisted of four modules including ordering, planning/scheduling, schedule confirmation and performance monitoring, and management reporting. One of the key development issues was the necessity to provide a central repository of information. This was achieved by the use of a central database, which is owned and maintained by Sapphire, but where multiple users are able to query, update and plan using a web interface.

6 SAPPHIRE'S RE PROCESSES DURING GLOBAL SOFTWARE DEVELOPMENT

Prior to the final contractual arrangements, one of the system architects travelled to the UK to gather and develop a business requirements document (BRD) through several face-to-face interviews. The interviewing and documentation was achieved in three weeks, and was used as a basis to inform the development team in NZ on the high level requirements for the system. The BRD does not represent the systems requirements but was mainly used for contract negotiation. The system architects in NZ commenced communication with the users in the UK once the contract negotiation between the two companies was complete. The users' Senior Management assigned two users to the project and they visited the site in NZ during initial stages of the project. Once the two users returned to the UK, communication between Sapphire and the two users occurred

* pseudonym

using email and the telephone. Sapphire decided to relocate the three system architects to the UK, as they were concerned that progress in the project was being hampered by communication constraints with all of the users located the UK, and all of the development project team located in NZ. Sapphire's project manager and the two junior developers remained in NZ. Hence the development team was divided between the two continents for the majority of the project.

Sapphire developed the system architecture using object-oriented tools and techniques in the framework of the Agile development methodology, as described by Cockburn (2002). The difference for this project compared with the previous literature concerning global software development, is that the requirements were never "nailed down". Hence change was continuous throughout the project and the RE processes essentially did not end until the point of implementation. One of the principles of Agile methodology, as described by Cockburn (2002:213) and valued by the team, was to "welcome changing requirements, even late in development" and that customer collaboration is valued over contract negotiation.

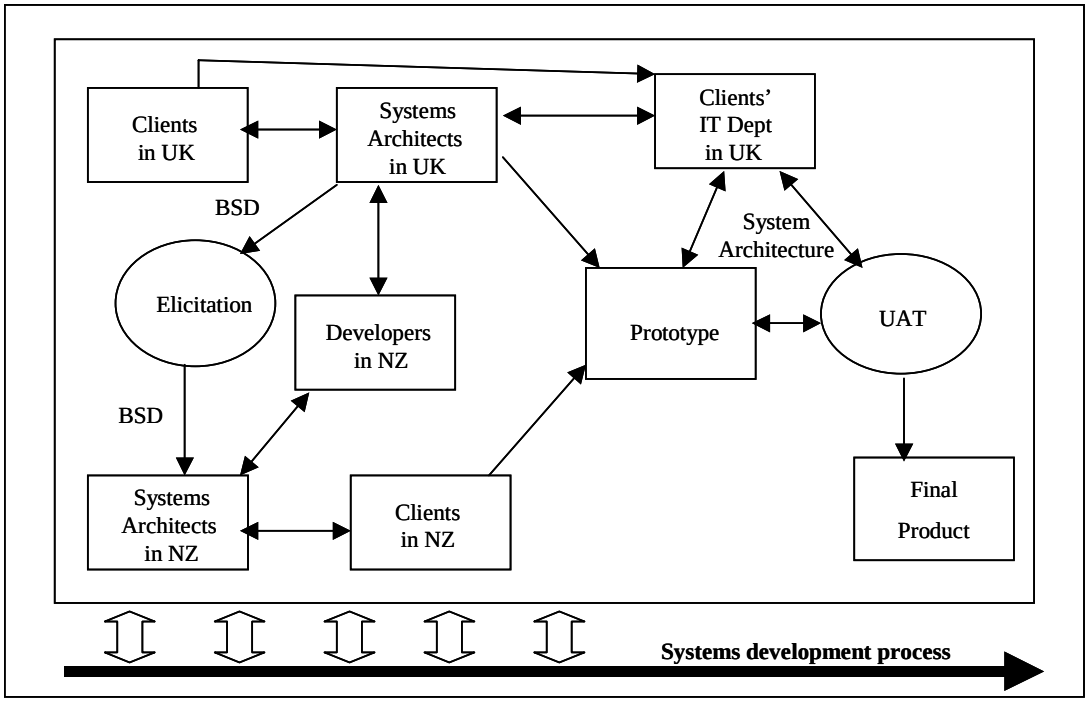

Figure 1: RE processes by Sapphire Software House during the project

Figure 1 indicates the processes used by Sapphire during the project to elicit the users' needs, build the prototype and then move to user acceptance testing and the final product. However, figure 1 presents the processes in a reductionist manner and fails to encapsulate the complex web of interactions that occurred. The process of RE in this case is complex because it entailed an iterative and ongoing set of communication challenges and modes within the development team, between the users and the developers, within the users' department and between the users' department, their IT group and some of the developers. The level and intensity of the communication rose throughout the development lifecycle, and the process was non-trivial due to the nature of the tasks and the necessity to work across the two continents using electronic communication media that were not always appropriate to the task.

Sapphire's system architects were frustrated by the pace of progress of the project once the users' IT department were in control, and they did not understand the bureaucracy of a large government department, compared with their flat, flexible company structure. They felt thwarted in their attempts to move the project forward and concerned that the deadline for the implementation would be affected by the users' IT department. They had underestimated the communication issues and need for good relations with the user's IT department for the success of the project. In Figure 1 there is a triangle of arrows labelled "systems

architecture" which indicates a communication bottleneck that caused the project to slow considerably and caused stress and frustration for Sapphire's team members.

Figure 2 represents the relative intensity of communication problems and requirements changes that occurred during the project. Because changes were embraced throughout the project, the RE process was iterative and ongoing. The curved line indicates that as the requirements were constantly changing and communication was more challenging, the relative intensity of communication challenges increased. The drivers for these challenges were associated with changes in the dynamics of the team, as various members were relocated between the two sites. At each interval where the challenge became too intense, (mainly due to miscommunication /misinterpretation and invalid requirements), new ways of communicating, or improved use of the current communication mode, were introduced to meet the communication challenge.

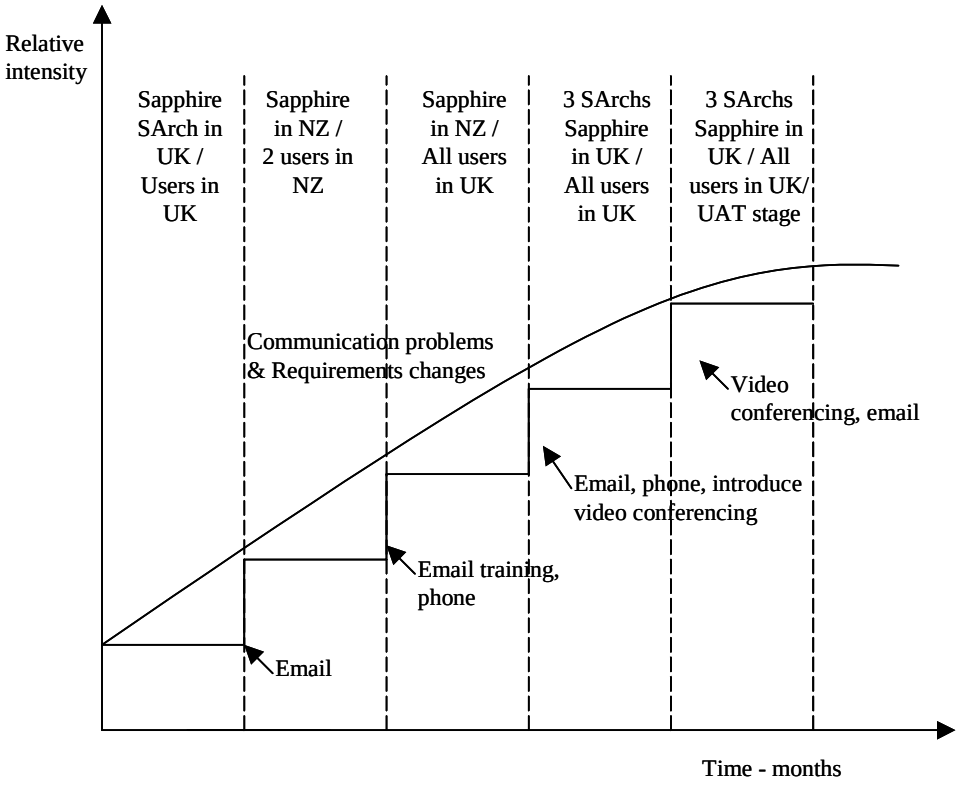

Figure 2: Intensity of communication during the project

According to one system architect, once the system design was "firm", Sapphire's project team lost some control over the project as the users' IT department led the project through the User Acceptance Testing (UAT) phase. While UAT would not normally be considered part of the RE process, the communication flows during this phase are interesting and relevant to the research, as there were considerable changes in requirements late in the project. This may be partly due to the philosophy of the Agile and iterative development methodology and partly due to the communication issues during the project.

This was contrasted by the communication between Sapphire and the two assigned users who had been located in NZ. The system architects placed emphasis on their rapport with the users, with whom they had been working. They considered that good communication channels with the users were imperative and worth much more to the success of the project than contractual agreements and formal signing off. The impediments to the RE process are of significant interest in this case because they highlight the influences of communication on the RE phase of the project, and from these impediments some solutions to the issues may be determined.

7 IMPEDIMENTS TO RE DURING GLOBAL SOFTWARE DEVELOPMENT

Informants in Sapphire highlighted some major issues concerning global virtual teams in this project:

- the majority of Sapphire's experienced skill sets were located at one site, hence miscommunication and misinterpretation of requirements occurred at the location where team members were less experienced;
- communication overheads were exacerbated by the short development time frame over such a large time zone difference;
- Sapphire were not able to identify and did not have appropriate access to the key users, hence the requirements gathered were tailored towards the users who were available;
- the difference between the culture of the working environments in the UK and New Zealand once the users' IT department gained control over the project; and
- the hidden meanings of cultural differences within the development team.

Sapphire considered that communication was the most difficult issue overall for the project. Communication issues, which the team members recounted as their main cause of impediments to the RE processes, may be further described in terms of the following four categories: distribution of the development team; distribution of the clients and the development team; cultural differences between the clients and the development team; and cultural differences among the development team.

7.1 Communication and Distribution of the Development Team

The system architects' main communication issues arose because Sapphire moved all of their experienced system architects to the client's site in the UK and the remaining developers in NZ were junior team members. Sapphire's project manager was not directly involved in the technical issues of the project. Hence, the junior developers in NZ lacked understanding of both the system requirements and the direction of the project. The three system architects agreed that problems arose because the content of the communication was complex (object-oriented design documents) and therefore required explanation and understanding. According to one system architect, the required tasks were difficult to explain using email. Hence email, as the communication mode for the system requirements, was not appropriate (Zack 1993).

The communication problems appear to add complexity to the RE process. The time zone difference between the two sites impacted on how quickly the requirements could be relayed from the UK to NZ. This in turn caused pressures within the development team, because when delays occurred due to the time difference, there was an added burden on the system architects in the UK to work long hours to address the communication issues as well as the requirements issues. When there were time delays as well as delays in understanding and interpretation by the developers in NZ, the system architects again felt under time pressure to address the communication issues and requirements problems. The system architects agreed that email was useful for sending documents, but to elicit and explain the requirements, it was necessary to provide a medium where further conversation, immediate responses, and validation of requirements were possible. In order to improve the communication flows, understanding of the system requirements and the speed of communication between the system architects in the UK and the developers in NZ, video conferencing was established. Sapphire felt that video conferencing was the only way to achieve a similar level of contact as face-to-face communication. By using both email and video conferencing, the development team were able to establish an "around the clock" work environment. However, this was only possible due to the heroic efforts on the part of the system architects who worked excessive hours. They did not consider this arrangement viable and sustainable in the long term.

7.2 Communication and Distribution of the Clients and the Development Team

According to the system architects the project was unique due to the time pressures placed on the team, and the separation of the clients and the development team added to the communication issues. Identifying and gaining access to appropriate users in the beginning of the project caused communication issues between the

clients and the development team. Those users who were located initially in NZ were able to devote the majority of their time to the requirements of the project. While this appeared an ideal situation, there were problems because they were not key users, and were uncertain of many facets of the system required by the users' department. Without Sapphire realising, the two users were able to direct the project after establishing face-to-face rapport in NZ, through filtering or ignoring the requirements of the other users in the UK. As one system architect recounts,

> *SA1: We started narrowing them [the requirements] down after [User 1] started galloping off on her horse at the great the rate of knots going in all sorts of strange and wonderful directions and wanting all sorts of things when she was here [in NZ].*

Another system architect confirmed that User 1 was powerful and determined to direct the project towards her goals to the exclusion of the other users who were located in the UK.

> *SA2: She [User 1] was the strongest user representation that we had…and there was one guy we met when we got over there [UK] and we didn't know about him at the start…we just didn't see him enough at all – we didn't get time from him until far too late in the project. You know [User 1] looked at this as an opportunity to – finally get **my** system…*

Had the system architects known who were the key users and been provided access to them in the UK throughout the entire project, the project outcomes and/or the project delivery time may have been different. While this is potentially no different to any development project (traditional or global) where access to key users is problematic, the issues were exacerbated in the virtual domain for this project. The distance between the system architects and the users, the time lag in gaining answers to questions about their requirements, and the problem that the responses were filtered through or ignored by the assigned users in NZ, provided no opportunity for the system architects to clarify any issues with the users in the UK; gain a different perspective concerning an issue; or ensure that the project was going to suit the majority of users. When the system architects relocated to the UK and obtained access to other users who were higher ranking stakeholders, they found many requirements were overlooked and many changes were needed. The system architects conceded that various key users were not as well consulted as those who were assigned and worked in NZ. While the underlying philosophy of the development team valued embracing change; and customer collaboration over contract negotiation (Cockburn 2002), the communication issues were evident when requirements changes were requested well beyond the UAT.

Overall, identifying and gaining access to key users is important to reduce the number of changes in requirements as the system is being developed. In the virtual domain, it is difficult for the development team to determine who are the key users, as the development team has no local or informal knowledge when they are not located on site. Further, the differences in time and distance of the virtual domain adds complexity and difficulty in gaining access to key users. In this case, once the system architects were located on the users' site in the UK, access to the users was still demanding, but with persistence, opportunities occurred which were not possible when the systems architects were located in NZ. This highlights clearly an impediment to RE as software is developed in the virtual domain.

7.3 Communication and Cultural Differences Between Clients and Development Team

Communication and cultural differences between the clients and the development team in this case was due to the difference in working culture between the two organisations. Sapphire highlighted the differences in working culture between government workers in the UK and a dynamic business in NZ. These differences caused conflict, frustration and power plays during the progress of the development. Sapphire was initially in control of the project, and according to one system architect, the users' IT department felt "de-powered" and "sidelined". However, once the users' IT department gained control of the project during UAT, the power base changed and then Sapphire was delayed and frustrated in their efforts to move the project forward.

The system architects lacked understanding of the culture of a traditional public service institution, which was evident during the interviews. Sapphire found the working pace of the client company could be an

impediment to the RE phase of the system development. They considered that culture of the government employees was a problem that influenced the progress of the project. Again while working cultures also vary between large traditional public service institutions and smaller flexible businesses in traditional software development, the communication issues are exacerbated when these companies are from different countries.

7.4 Communication and Cultural Differences Among the Development Team

The final communication aspect is the cultural differences among the development team members. This occurred in two ways in this research. When the system architects were located in the UK, they were able to devote extended working hours to the project because they had no other outside commitments. However the developers in NZ remained in their home environment with everyday commitments, and were not able to devote endless overtime to their work. There was pressure from their more senior colleagues in the UK to work longer hours and this caused conflict in the team.

The other aspect relating to communication and cultural differences in this case concerns the cultural (ethnicity) differences within the development team. The system architects did not explicitly admit to any communication issues due to cultural differences within the team, but the junior developers did attribute some differences in understanding of the design requirements to cultural differences. While all of the development team were long-term residents of NZ, the system architects were born in NZ, but the two junior developers were from Nicaragua and Malaysia. As Thanasankit (1999b) has indicated, there exists hidden meanings and interpretations of requirements information between developers from different cultural backgrounds, and this may be an impediment to the requirements engineering process. Overall the miscommunication and misinterpretation of the system requirements between the system architects and the junior developers may be attributed to: inexperience, the difficulties in interpreting complex designs using less rich communication media, as well as lack of understanding between the two sites of the hidden meanings relating to cultural differences.

8 SAPPHIRE'S SOLUTIONS TO RE ISSUES

Overall, the four main influences on communication as recounted by Sapphire's project team included the distribution of the development team; the distribution of the clients and the development team; cultural differences between the clients and the development team; and cultural differences among the development team members.

Sapphire's main solutions to these problems were two fold:

- the project team elected a project leader (from the three system architects) who became responsible for verifying and validating communication and information rather than designing the system; and
- the team members moved into the same room at their restive sites, and then at both sites they set up video conferencing facilities with printable whiteboards. They conducted two videoconferences per day, in the morning and the evening. This enabled the UK part of the team to communicate information to the NZ part of the team before they left for the evening, and vice versa. They were therefore able to work on the project continuously.

The three system architects who relocated to the UK were effectively working 20-hour days in order to coordinate the information, requirements and design. Overall Sapphire Software House was prepared to support their development team to work long hours on this project for a short time in order to win the contract, and the project was successful and is continuing to provide an on-going business relationship. However, when some members of the team were asked their lessons learned, they admitted that they did not enjoy working on this type of project and would not undertake a similar project in the future.

9 DISCUSSION AND CONCLUSIONS

This paper has described a case study where an information system is developed and implemented in the virtual domain. The paper is of significance because it extends our knowledge and understanding of the ways in which software development teams work in the virtual domain. The paper particularly highlights some communication issues that may arise during the RE phase, and the impediments to RE during global software development. This paper supports the work of Mockus and Herbsleb (2001), Ramesh and Dennis (2002) and Damian and Zowghi (2003), and adds to knowledge in the area of global software development, by demonstrating that communication involves a complex web of interactions and there are many layers of influences on the RE processes, so that determining the impediments, and therefore solutions to communication issues, is problematic.

The literature commonly argues that the nature of "true" virtual teams is that members of the team rarely or never meet, and the team uses electronic communication as their main mode of communication (Jarvenpaa and Leidner 1998). As reported by Vogel, Davison and Shroff (2001), the tasks undertaken by their teams are not as complex as software development, yet according to Vogel et al (2001) the teams still experienced communication difficulties. We consider that the development team in this case experienced a great deal of pressure, as they were required to fulfil a contractual agreement and ensure a strong relationship with their client. As evidenced by this case, Sapphire Software House was prepared to do their utmost to win an ongoing contract, and the development team members placed extraordinary effort into their work in order to make the project successful. Sapphire did not hesitate to fund travel to remote sites in order to improve their chances of success in the project.

Ramesh and Dennis (2002) propose a new type of team, namely the "object-oriented" team as they note that some global teams act differently to others reported in the current literature. They consider "object-oriented teams used routine, mature work processes with well-defined task deliverables" (Ramesh and Dennis 2002:8). This case describes a team that does not use either of their "object-oriented" or "integrated virtual team" classifications, but makes use of a different model to ensure the project's success.

Hence, as evidenced by this case and for some complex software development projects, we consider that a "hybrid" model of virtual teams is more likely. The hybrid model asserts that some members will meet face-to-face while others will never meet, but the business must weigh the benefits of being "truly virtual" against the need to create a successful project outcome, and that this will increase the development cost and/or time for the project. This concurs with recent research by Griffith, Sawyer and Neale (2003: 268) that "most of today's organizational teams are likely to fall into the large hybrid category of teams." Further, it supports the work of Ebert, Parro, Suttels and Kolarczyk (2001) that working in the global context does not necessarily reduce the costs of software development or reduce development time.

Ongoing research is required to describe the processes of other development teams undertaking RE during global software development, and to improve our understanding of the ways businesses solve the communication issues in global virtual teams. During global software development, the task of developing an information system or product goes beyond simple integration or cooperative alliance (Roy 2001), because the task (system development) to be accomplished is a "given" (the team know that their combined effort will accomplish the task) but, the collaboration of the team requires shifts in perception and a broadening of their world views (Roy 2001) so that the goals can be achieved. This complex process requires communication competence, which Gudykunst (1993:22) defines as "a minimisation of misunderstandings". Communication competence may be achieved through effective and appropriate communication (Roy 2001). However, what constitutes both effective and appropriate communication is complex and open to conjecture. Hence, to understand a team's ability to collaborate in RE during global software development, it is necessary to understand the team's background and propensity for effective and appropriate communication. Overall the RE processes will be influenced by the communication and the various distributions of the development team and their clients.

References

Battin R D, Crocker R and Kreidler J (2001) Leveraging resources in global software development, *IEEE Software*, March/April: 70-77.

Boehm B (1981) *Software Engineering Economics*, Prentice-Hall: NJ.

Byrd T A, Cossick K L and Zmud R W (1992) A synthesis of research on requirements analysis and knowledge acquisition techniques *MIS Quarterly* March, 117-138.

Carmel E (1999) *Global software teams: collaborating across borders and time zones*, Prentice Hall: NJ

Carroll J M and Swatman P A (1997) How can the requirements engineering process be improved?, in *Proceedings of the 8th Australasian Conference on Information Systems*, University of South Australia.

Cockburn A (2002) *Agile software development*, Addison-Wesley Pearson Education: USA

Creswell J W (1998) *Qualitative inquiry and research design: choosing among five traditions*, Sage Publications: UK

Damian D E H, Shaw M L G and Gaines B R (2000) A study in requirements negotiations in virtual project teams, In *Proceedings of the 8th European Conference on Information Systems 2000*, Austria.

Damian D E H and Zowghi D (2003) The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization, *In Proceedings of the IEEE joint International Requirements Engineering* 9-13 September 2002: 319-328

Ebert C, Parro C H, Suttels R and Kolarczyk H (2001) Improving validation activities in a global software development, In Proceedings *IEEE Software:* 545-554

Goguen J A and Linde C (1993) Techniques for requirements elicitation, *Proceedings of the IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press, USA, 152-164.

Greenspan S, Mylopoulos J and Borgida A (1994) On formal requirements modeling languages: RML revisited, In *Proceedings 16th International Conference on Software Engineering*, Sorrento, May

Griffith T L, Sawyer J E and Neale M A (2003) Virtualness and knowledge in teams: managing the love triangle of organizations, individuals, and information technology, *MIS Quarterly*, 27(2): 265-287

Gudykunst W B (1993) Toward a theory of effective interpersonal and intergroup communication, In Wiseman and Koester (Eds), *Intercultural communication competence*, Sage Publications: CA.

Hanisch J (2001) Requirements engineering in virtual software development: achieving balance, In *Proceedings IRMA 2001*, Toronto Canada, May

Hanisch J, Thanasankit T and Corbitt B (2001) Exploring the cultural and social impacts on the requirements engineering processes – highlighting some problems challenging virtual team relationships with clients, *Journal of Systems and Information Technology*, 5(2): 1-19.

Herbsleb J D and Moitra D (2001) Global software development, *IEEE Software*, March/April, 16-20

Jarvenpaa S L and Leidner D E (1998) Communication and Trust in Global Virtual Teams, *Journal of Computer-Mediated Communication*, 3 (4), http://www.ascusc.org/jcmc/vol3/issue4/jarvenpaa.html. [date accessed 24/10/2001]

Karolak D W (1998) *Global software development: managing virtual teams and environments*, IEEE Computer Society Press USA

Land F F (1998) A contingency based approach to requirements elicitation and systems development, *Journal of Systems and Software*, Jan, 40 (1): 3-6

Land F F and Somogyi E K (1986) Software engineering: the relationship between a formal system and its environment, *Journal of Information Technology* 1(1): 14-21

Mockus A and Hersleb J (2001) Challenges of global software development, *In Proceedings of 7th International Software Metrics Symposium* 4-6 April 2001: 182-184

Ramesh and Dennis (2002) The object-oriented team: lessons for virtual teams from global software development, In Proceedings of *the 35th Hawaii International Conference on System Sciences*, January.

Roy M H (2001) Small group communication and performance: do cognitive flexibility and context matter?, *Management Decision*, 39(4): 323-330.

Staples S (1996) An Investigation of Some Key Information Technology -Enabled Remote Management and Remote Work Issues, in *Proceedings of 7th Australasian Conference on Information Systems*, 2: 665-676.

Thanasankit T (1999a) Social interpretation of evolving requirements – the influence of Thai culture, in Zowghi, D. (ed) (1999), *Proceedings of The Fourth Australian Conference on Requirements Engineering*, Macquarie University, Australia

Thanasankit T (1999b) *Towards understanding the impact of culture on requirement engineering – ethnographies of Thai systems analysts*, Unpublished PhD Thesis, University of Melbourne.

Vogel D R, Davison R M and Shroff R H (2001) Sociocultural learning: a perspective on GSS-enabled global education, *Communications of the AIS*, 7(9): 1-40.

Zack M H (1993) Interactivity and Communication Mode Choice in Ongoing Management Groups, *Information Systems Research*, 4(3): 207-239.

Zave P (1995) Classification of research efforts in requirements engineering, in *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, March 27-29, IEEE Computer Society Press: USA.