

December 2005

Differentiating Local and Global Systems Requirements Gathering Processes in IS Software Development Projects

Jo Hanisch
University of South Australia

Brian Corbitt
Shinawatra University

Theerasak Thanasankit
Mahidol University

Follow this and additional works at: <http://aisel.aisnet.org/pacis2005>

Recommended Citation

Hanisch, Jo; Corbitt, Brian; and Thanasankit, Theerasak, "Differentiating Local and Global Systems Requirements Gathering Processes in IS Software Development Projects" (2005). *PACIS 2005 Proceedings*. 17.
<http://aisel.aisnet.org/pacis2005/17>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Differentiating Local and Global Systems Requirements Gathering Processes in IS Software Development Projects

Jo Hanisch
School of Computer and
Information Science
University of South
Australia
jo.hanisch@unisa.edu.au

Brian Corbitt
Shinawatra University,
Thailand
Deakin University,
Australia
bcorbitt@shinawatra.ac.th

Theerasak Thanasankit
Mahidol University,
Thailand
jua@tezza.com

Abstract

The paper discusses difference in the requirements gather processes between a local team in software development and a global software development team. The paper highlights the cultural differences between a uniform cultural team and a multicultural team and argues that the communication issues that arise are inevitably associated with culture and geography. The second major issue raised in this paper relates to differences in the relationships between the teams and the clients. In the local exemplar, committees were more formal and affective rather than the informal one used in global software development. Finally the paper shows that the process of requirements gathering was different based possibly on the impact of culture in the one instance where that culture predetermined the actual process to be used.

Keywords: Requirements gathering (engineering), global/local software development, cultural differences, communication

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, 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) among others, have focused on RE during global software development. Yet there remains little understanding of the differences in the RE processes during traditional and global software development.

This paper explores the ways employees in two domains (local and global) understand the processes of RE during systems development, with a view to understanding the commonalties and divergences between the two domains. In order to address this aim, two similar-sized case studies (one from a traditional software development environment and the other from a global software development environment) were conducted. This paper explores the processes of RE during software development in the two domains from a social and cultural perspective.

This paper first reviews RE in traditional software development, followed by the RE processes during global software development. A description of the two cases includes the processes used by software developers during RE in both local and global domains. A

discussion follows, which highlights the commonalities and divergences of the RE processes.

2. Requirements Engineering During Traditional Software Development

RE has been defined as “the disciplined application of scientific principles and techniques for developing, communicating, and managing requirements” (Christel and Kang 1992:3). This definition of RE is supported by Loucopoulos and Karakostas (1995), who define RE as:

“...the systematic process of developing requirements through an iterative co-operative process of analysing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained.”
 Loucopoulos and Karakostas (1995:13)

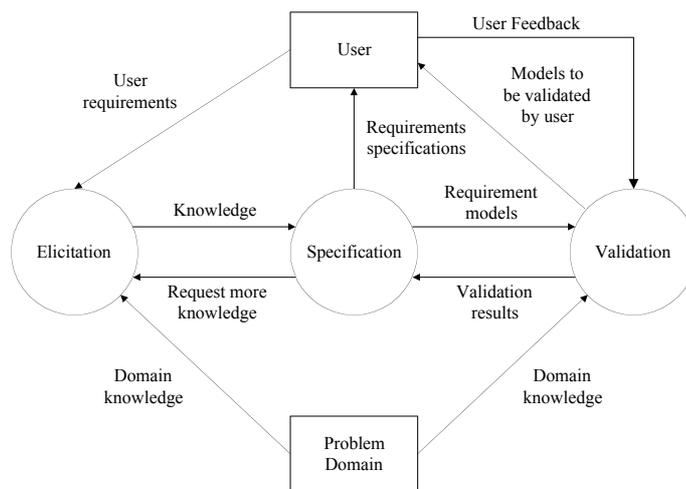


Figure 1: Requirements engineering process (Loucopoulos and Karakostas 1995: 21)

As indicated in Figure 1, RE covers all the activities of discovering, documenting, and maintaining a set of requirements for building a computer-based information system”(Thanasankit and Corbitt 1999; Sommerville and Sawyer 1997). During RE, the requirements engineer (or systems analyst) needs to be mindful of the objectives/outcomes of the RE phase and implement appropriate processes or techniques which will help to avoid failure of the IS (Macauley 1996). But the failure of many IS development projects is due not just to 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 requirements engineers to understand communication and cooperation, as well as social complexity during RE (Thanasankit 1999). RE research has traditionally been positivist in its approach, largely focusing on the methods and tools used in the gathering elicitation and validation of requirements. Many researchers conclude that the more formal the techniques used, the more likely that the requirements will be clearly defined and understood (Carmel 1999). However, organisations need to consider the emotions and culture of users and IS specialists. It has been shown that different cultures will perform tasks not only because they are responsible for the task, but because they wish to maintain surface harmony and trust between the group (Thanasankit 1999) and this often inhibits formal sign offs of requirements specifications, which in turn causes delays and potential failure of the project. In these cases, the imposition of formal, often western philosophies and methods, have a negative impact on the requirements gathering process (Thanasankit 1999).

3. Requirements Engineering During Global Software Development

Within the past 6 years, Karolak (1998) and Carmel (1999) describe the concept of global software development, where the need to solve complex tasks, such as RE, occurs in the global domain. 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.”

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 (Lipnack and Stamps, 1997) 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.

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.

Zack (1993) indicates the main distinction between local 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 global 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 global domain, however these methods may also inhibit the social aspects of virtual communications (Hanisch and Corbitt 2004), 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 global domain.

4. Research Methodology

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. The principal method for collecting data was by taped in-depth interviews, which lasted from between one and a half to four hours. The interviews were informal and semi-structured with non-directive, open-ended questioning to stimulate the subjects' thoughts. The interviewees had freedom to describe their experiences and problems beyond the questions' boundaries.

In both cases, questions were constructed to encourage interviewees to compose stories in useful constructs to analyse processes of reality construction (Lindloff, 1995). Stories provide a rich entrée into the nuances of meaning and enable the interviewee to be more conscious of the meanings which they attribute to events and concepts (Louis and Sutton, 1991). 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).

Through an analysis of the interviewees' stories and symbols, shared themes emerged which highlighted the constructions of the traditional software developers (in case 1) and the global software developers (in case 2) including the influences on the RE process on these constructions. 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.

5. Modelling the Requirements Process during Traditional Software Development – CASE STUDY 1

Thantawat Software House is a Thai software development company with about 200 employees. They specialise in outsourced software systems development. They are located in Bangkok in Thailand and operate exclusively in the Thai marketplace. Thai systems analysts use formal and informal methods of collecting requirements but the key issues involved related to business structures which impose as much influence as the actual RE collection methods.

In this case study the Thai systems analysts reported that one of the major effects on the RE process in Thailand was the existence of a 'tall' organisational structures (Hofstede 1991; Trompenaars 1993; Holmes and Tangtongtavy, 1996, Thanasankit 1999). This structure resulted in the creation of two committees to oversee the development of new or changed RE for information systems. The first was a development committee, with which the systems analysts worked closely and which was the first point of contact and reference during systems development. This committee was made up mostly of middle level managers, heads of departments or units and, sometimes, operational level users. It was usually quite large and representative of where the projects had impact in the organisation.

The development committee assisted with:

- providing the systems analysts with requirements;
- arranging users for systems analysts for elicitation;

- providing the systems analysts with feedback about the requirements specification;
- assisting the systems analysts with administrative tasks; and
- providing assistance for the systems analysts when problems arose.

The second committee, the steering committee, had minimum involvement during systems development. It comprised senior level management and was always small. The steering committee became involved when the development of the system reached a stage where approval was needed to increase costs or to include additional features or functions, which were not covered previously. Involvement also happened when the development committee had become immersed in numerous conflicts about issues upon which the members could not agree.

These committees mirrored the expected and accepted hierarchical social structure of Thai organisations. There was place and rank for everyone working on the systems. Committees and individuals had rank and the nature of human agency within Thai organisations was based on their rank and position within the organisation. In RE, where the process was always outsourced, the link between the client and the consultant was formalized by creating new structures, which were representative of all other structures and which existed in the client organisation. Their establishment was implied in any contract. This notion of rank is informed and constructed within Thai culture. As a result delegation of responsibility moves upwards to points of highest rank, rather than being delegated down to lower ranks.

The result of these structures was a long decision-making processes which began when the requirements specifications were presented to the development committees for evaluation. Formal communication was only through this committee. The feedback provided by the committee members was used by the systems analysts to improve the requirements specifications. This process was repeated until the development committee members were satisfied with the requirements specification. It reflected the iterative, complex and differentiated nature of the RE processes in this case study (Figure 2). The process described is substantially different from the ‘ideal’ model presented in Figure 1.

These iterative processes in making decisions about requirements for projects were a reconstruction of the ways Thais worked and communicated. There was an expectation of new requirements and expectation of constant iterations, as that was ‘normal practice’ in all Thai contexts where bureaucratic, hierarchical decision-making was an integral part.

After the development committee members were satisfied with the requirements specification, they then passed the requirements specification up to the steering committee for comment and/or final approval. Typically, additional requirements were identified and the iterations began again and continued until both sets of committee members were satisfied. This process resulted from fears within Thai organisations about making mistakes. Thais are not risk takers and they do not want to create any uncertainty. One way to avoid risk and uncertainty was to begin a circuitous, iterative process until all risk and all uncertainty is eliminated.

The approval of the requirements specification then was a lengthy process. In most Thai organisations, where highly developed organisational structures were in place, the Thai systems analysts were presented with long waiting periods to get approval for the requirements specification or the prototype system. This slowed down the systems development processes and hindered the systems analysts ability to move onto the next stage within systems development. Sometimes the systems analysts needed to move onto the next process without approval for the requirements specification or the prototype systems, just to maintain continuity and speed up development and implementation.

members of the development committee often did not feel comfortable with approving the requirements because by being involved in decision-making process may bring them unwanted responsibility. Therefore, by adding more requirements, the members could prevent others in the development committee from approving a system that might not meet the organisation's requirements. Each committee kept reiterating requirements and adding new ones, establishing an evolving pattern of requirements gathering and avoidance of decisions.

The social construction of authority and acceptance of responsibility in RE in organisations in Thailand reflects the hierarchical nature of society and more especially the construction of responsibility at the top. Since responsibility was always upwards, requirements approval was always delayed and sometimes inhibited. Because of this elongated decision-making process, it was difficult to set a termination period for the RE processes in Thailand. Therefore, the timeframe for RE processes in Thailand depended on each individual systems analyst's ability to use their own intuition and judgement to progress on to the next stage of their systems development methodology.

The need to conduct RE in parallel with the systems software development processes was a response to Thai expectations of flexibility to maintain close relationships. The two processes needed to run in parallel in the Thai context to allow the systems analysts to progress onto other stages of systems development and/or RE to overcome time lags in decision-making which seriously affected the set time frames of the various projects. Even though decisions had been made, there was another problem in that it often took a long time to implement those decisions. The hierarchical nature of Thai organisations slowed the process down. Therefore, the Thai systems analysts needed to carry on to the next stage while waiting for direction from the development committee on how the decision was to be implemented or incorporated within the information system.

The long decision-making processes also contributed to the attitude expressed by the Thai systems analysts about the insignificance of the specification process. Since in the Thai context the requirements specification documents could not be static, and because the systems analysts could not freeze the requirements, the construction of the requirements specification was an on-going activity throughout systems software development.

In summary, it has been argued here that the long decision-making process for RE, endemic in Thai organisations, had a significant impact on the RE processes practiced by the Thai systems analysts. Their processes needed to be flexible to enable them to progress on with systems development whilst waiting for decisions to be made and approved by their clients. The hierarchical nature of Thai society and its impact on the formal relationships within social strata, even within organisations, contributed to the acceptance of delays in decision-making. The search for certainty and the need by Thais to avoid risk ensured that any decision-making was slow and remained an iterative process until high levels of certainty emerged. To meet these demands of clients, the Thai systems analysts built prototypes as early as they could in the process. However, these delays and the existence and acceptance of uncertainty added to misconceptions within the problem domain for any project.

6. Modelling the Requirements Process during Global Software Development – CASE STUDY 2

Secure Traders* is part of a large company with 5,800 employees. The main company, which has been listed on the London Stock Exchange since 1989, has divisions in the health,

* pseudonym

insurance and banking industries. During the past six years the corporation has been developing and enhancing a complex software product between its IT group, which was located in Melbourne, Australia (headquarters of company), and several sites across Asia and the UK. To achieve the development of the product, employees needed to make sense of their clients' requirements and interpret their needs using electronic communications. This was achieved within the constraints of both corporate policies and government legislation.

To model the requirements processes during global software development, the interviewees were asked to describe both the processes from the viewpoint of dealing with a new client, and then from the viewpoint of a current client who had identified an issue or requested changes to the product. These two perspectives are now described.

6.1 Communication Flows for New Clients

Once a new client decided to purchase the software product, Secure Traders' development team became involved. The business analyst (BA) identified any potential issues regarding the client's local market, such as legislation for their stock exchange and settlement requirements. In order to develop a project plan, the BAs determined the client's target dates and high-level requirements. Then to familiarise the client with the product and its functionality, the BAs conducted a series of face-to-face business process reengineering (BPR) workshops with the clients. The workshops ran over three to four weeks, where the business processes were reviewed and any new requirements were detailed and documented by the BAs. Then the software developers and BAs discussed the client's requirements, and reported back the collective recommendations concerning the required system and implementation to the client.

The aim of the workshops and the follow up discussions was to define the requirements to a lower (more detailed) level and this assisted with costing the system and the timeframe required for implementation. Following the face-to-face workshops, most of the communication between the client and the BA was through electronic means, and according to one BA, they were then perceived by the clients to have a lesser degree of "presence". He believes that the lack of physical presence negatively influenced the speed in which the clients both made decisions, and reviewed and returned the documents to Secure Traders. All follow up discussions were made using teleconference or email and this could be problematic. One BA recounted that emails caused delays in the decision making process. Although information was conveyed between the parties using email, the issues were not brought to completion and that was the cause of the delays. Pliskin and Romm's (1996) research support this issue of delayed decision making using email.

The other issue identified was the speed in which the client responded to the information or documents Secure Traders provided. It was much more difficult to ensure action from the client when they were not located in the same place. In the traditional domain, the BA was able to physically visit the client and this helped to keep them working on the requirements documentation. Sproull and Keisler (1986) support the notion that electronic communication influences the speed and return of response. Romm, Pliskin and Rifkin (1996) consider that email is less personal, and therefore more likely to be ignored (Markus 1994).

Figure 3 indicates the requirements and development processes that occurred when a new client purchased the software product. The figure demonstrates a cycle of formal agreements that must be signed-off by the clients, before the final product was developed. The agreements were presented to the Product Review Committee (PRC) which fulfilled a number of duties, including verification that the requirements could be realistically developed and included in the product; the costing of the project; and estimated time lines for delivery.

The product was not developed unless the clients signed-off the requirements documentation and contract. The PRC was the central focus for all the various roles on the team, including the development people, the implementation people, the BAs and the testing people, and therefore is located centrally in Figure 3.

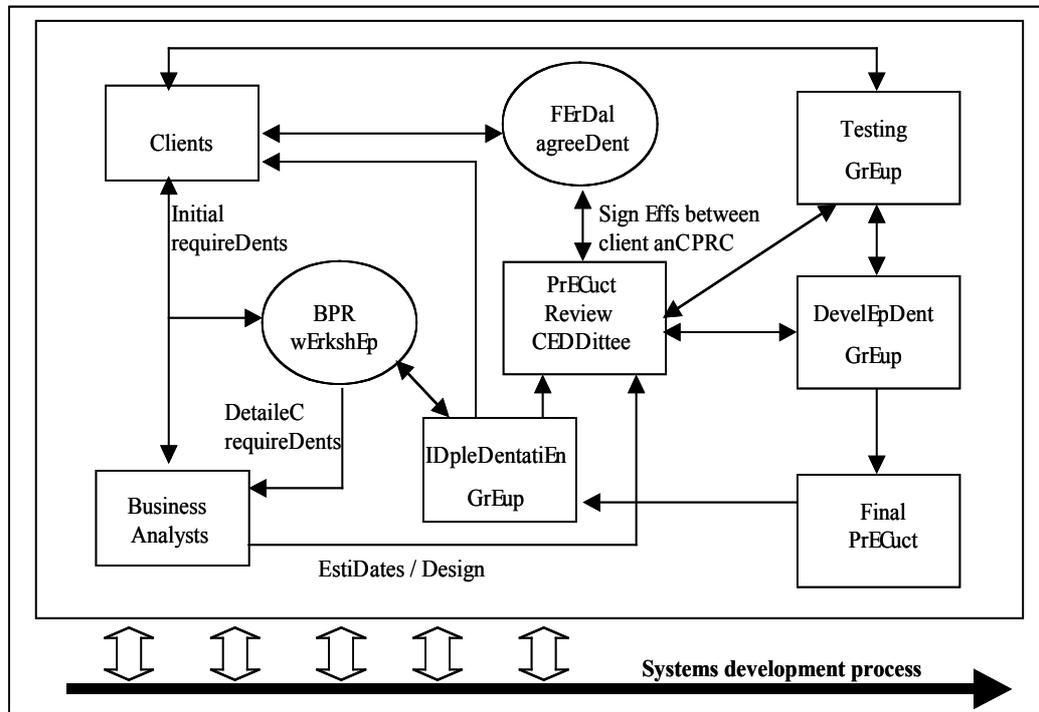


Figure 3: Requirements engineering processes of Secure Traders for new clients

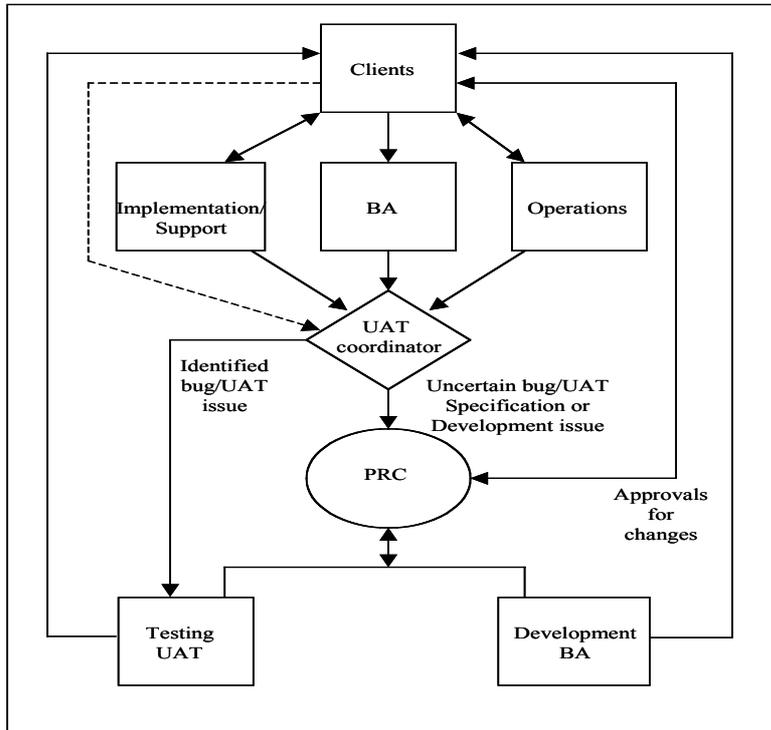
The development people were removed from the clients and they only interacted with the clients indirectly through the PRC. Until the product was implemented, there could be a number of requirements changes brought to the attention of the team. New requirements work, which caused changes to the specification, came through the BAs. Any bug/User Acceptance Testing (UAT) issues as a result of testing were brought to the attention of the team through the testing people. In most cases the issues would go through the PRC. However, straight bug issues may have been solved directly by the testing people.

6.2 Communication Flows for Existing Clients

For existing clients, the UAT coordinator handled any exception testing and bugs that the clients reported to the team. New requirements work was sent to the UAT coordinator through four different paths, including the implementation group; the operations or support group; the BA; or directly from the clients, as indicated in Figure 4. Often when the BA visited their client on site, new requirements issues would arise. When this occurred, the BA reported the requirement to the UAT coordinator on the client's behalf.

On other occasions the operations people asked a BA to visit the client directly, to circumvent the involvement of the implementers or the support people. The size and location of the client company often determined the way that requirements work was highlighted to the UAT coordinator. However, regardless of the type of client, all requests for changes or bug notifications went to the UAT coordinator who then decided whether the requirements work was a clearly identifiable bug/UAT issue or whether the issue should be reviewed by the PRC as development work.

If the work were identified as a bug/UAT, the issue was assigned to the testing/UAT group. Once the bug or UAT issue was resolved the testing group would liaise directly with the client. If the UAT coordinator was uncertain whether the issue was a bug/UAT, specification



or development issue, then the PRC reviewed the request and liaised with the client concerning any approvals or cost variations that needed to be signed-off.

Figure 4: Requirements engineering processes of Secure Traders for existing clients

Overlaying what the System Architect Manager regards as the “chaotic picture on the top” is the simple view that the requirements have been divided into one of two categories: either it was a bug/UAT issue, or it was a new requirement and was therefore a business analyst/development issue. The process involved all relevant people in their appropriate roles, who were in contact with the existing clients and who were approached with an issue to feed their requirements into a similar standard development pathway. In this way the developers were not directly involved in issues that did not concern straight development work.

Once the PRC had the signed approvals from the client for the new requirements work to proceed, the developers were informed of the request, and together with the implementers, the product was modified and installed for the clients. In this way Secure Traders were able to maintain control over their business processes; they controlled any changes to the current product, and ensured that the most appropriate people addressed any issues raised by the clients. Secure Traders also used this process to ensure the client had agreed to pay for any enhancements or changes before the requirement work commenced.

7. Themes of Commonality and Divergence

Through an analysis of the data, there are common issues recounted by interviewees in both case studies. However, the reasons for these issues, the ways they develop, and the ways they are managed differ between the cases, and this provides the interest for this paper.

7.1 Communication, Culture and the RE Process

All interviewees recount communication as their most demanding issue during RE. Communication is strongly linked with culture for these organisations, as in both cases there

is evidence that cultural differences influence both communication and the RE processes. Because the local project in case 1 includes software developers from a collectivist society, the cultural and communication issues are strongly evident. Similar cultural and communication issues occur in case 2 as expected when software development occurs across two or more countries. The authors recognise that not all local software development teams would necessarily share the same cultural issues as highlighted in this paper, because not all local projects draw from a collectivist society. Having similar cultural groups in both case studies for this research provides an excellent basis to compare communication and cultural issues between the two domains.

Interviewees from both case studies experienced difficulties in the interpretation of requirements during RE, yet they exhibited different ways of resolving those misinterpretations. During software development in the local domain, it is clear that Thais prefer to communicate by using interviews and face-to-face interaction as a first attempt to resolve any misinterpretations of requirements. The necessity for face-to-face interactions occurs for a number of reasons and influences the RE processes greatly. In case 1, communication is not only about passing requirements from clients to software developers, but reflects the behaviour of software developers. Software developers in Thailand are perceived by their clients as their subordinates, and working closely together with day-to-day interaction is important to achieve trust and strengthen close relationships. In Thailand, trust can be achieved only by face-to-face and rich communication. Talking with the systems developers, visiting the software house and general conversations are part of system development interaction between clients and systems developers in Thailand. Showing friendliness was important during RE process. The systems developers reported that Thais work to create and preserve relationships, therefore the work was more relationship-oriented rather than work-oriented. Achieving “belongingness” to the development group is an important cultural element for these developers. In this case, achieving belongingness for the development team promoted increased cooperation from the clients during RE.

In case 2, where face-to-face interactions were limited and electronic communication technologies were the main mode of communication during RE, there was minimal social interaction among the software developers and between the developers and their clients. Interviewees recount that there were many misinterpretations of requirements due to both the complexity of the messages being sent and received, and the differences in language/terminology used between the cultural groups. Explaining complex system architecture diagrams and gathering clients’ needs were more difficult when using email. Electronic communication media used by global teams has been shown to decrease rich communication (Jarvenpaa and Leidner 1996). Trust is a major element for successful global virtual teams (Carmel 1999), but the use of electronic communications limits social interactions which are important for building trust. In case 2, when communication and cultural differences occurred, team members made use of teleconferencing in an attempt to resolve issues and misunderstandings. Teleconferencing is considered a richer medium during communication than email. Even so, according to one BA, when using electronic communication media rather than visiting the client’s site, the developers were perceived by the clients to have a lesser degree of “presence”, influencing the speed in which the clients both made decisions, and reviewed and returned the documents to Secure Traders. The software developers consider that it was difficult to convey a sense of urgency using electronic communications, they felt less important or that their work had less priority in the eyes of the client, and they had problems achieving cooperation from the clients.

7.2 Committees

Both organisations used formal committees during RE in an effort to control changes, agree on costs and improve decision making. Case 1 used the development and steering committees, and case 2 used the product review committee. However, there were some differences between the two domains.

In case 1 formal approval was rarely signed-off and agreed prior to the commencement of development work. Even through to build stage the requirements were being reviewed and changed by the steering committee. As mentioned, the search for certainty and the need by Thais to avoid risk by seeking approvals through both committees ensured that any decision-making was slow and remained an iterative process until high levels of certainty emerged. However, as the Thai analysts were accustomed to the delays in decision making, they developed prototypes which were flexible and enabled them to progress with systems development whilst waiting for decisions to be made and approved by their clients.

In case 2, regardless of any cultural differences between the software developers and/or the client, the product review committee had exclusive power to determine whether or when changes were incorporated into the software. No development work commenced unless or until approvals with the clients were signed-off with the committee. Hence, even though the developers in case 2 made use of prototypes, systems development did not occur in parallel with the approval process and RE was delayed while approvals from the clients were being sought. The product review committee was at times described as a “bottleneck” by software developers, but they admitted that in the global domain this process was essential to provide clarification of requirements and prevent further misinterpretations prior to development.

The committees in the two case studies were used in the approval process. However in the local case they were not used to ensure clarification of requirements as the analysts were in daily face-to-face contact with their clients, but they were used for the cultural need to maintain certainty. In the global case the committee was used to ensure messages were clearly conveyed and agreed. The team members consider this essential as meanings were often misinterpreted using electronic communication.

8. Conclusion

The two case studies in this paper highlight the differences in software development processes between local and globalised development. For the local case, where face-to-face interaction was possible, formal processes were avoided or ignored while there were delays in decision making from management. Prototyping assisted the analysts to continue working on the system development. However, in the globalised case, where electronic communication was necessary, formal approvals were enforced, even though prototyping was also used.

Further, this paper also highlights the influences of cultural differences on communication during RE. In the local case, the cultures of a collectivist society played a major role in the processes of RE. While culture influenced communication, it affected the behaviour of analysts and influenced the client-analyst relationship. In the globalised case, cultural differences influenced the understanding and interpretation of message content rather than behaviour of clients and software developers. A key implication of these differences relates to management of resources, especially personnel and time in software projects. Businesses engaged in software development projects must consider the extra costs of cultural practice and weigh that against the complexities of global teams. Perhaps it can be suggested that the use of global teams breaks down the cultural barriers, as members of those teams are aware of cultural differences and accept them. In these teams the impact of their own cultural practice is somewhat diminished by the engagement across multiple cultures. However, in local teams cultural practices are entrenched and will always prevail. In certain instances this

creates opportunities for significant advantage, whilst in others extra resources might need to be deployed as time and resources costs increase.

References

- Boehm, B. (1981). *Software engineering economics*, Prentice-Hall: NJ.
- Burn, J.M., Davison, R., and Jordan, E. (1998). 'The information society – a cultural fallacy?', *Failure & Lessons Learned in Information Technology Management*, 1, pp. 219-232.
- Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*, Prentice Hall: NJ.
- Creswell, J. W. (1998). *Qualitative inquiry and research design: choosing among five traditions*, Sage Publications: UK.
- Christel, M. G. and Kang, K. C. (1992). 'Issues in requirements elicitation', *Technical Report CMU/SEI-92-TR-12*, SEI: PA.
- 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*, (3-5 July), Vienna, 2: 937-944.
- Damian, D. E. H. and Zowghi, D. (2003). 'An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations', In *Proceedings of the 36th Hawaii International Conference on Systems Sciences*, (6-9 January), IEEE Computer Press: USA: 19-28.
- Goguen, J. A. and Linde, C. (1993). 'Techniques for requirements elicitation', In *Proceedings of the IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press: USA: 152-164.
- 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.
- Hanisch, J. and Corbitt, B. J. (2004). 'Requirements engineering during global software development: some impediments to the requirements engineering process – a case study', In *Proceedings of the 12th European Conference on Information Systems*, (14-16 June), Turku, Finland.
- Hofstede, G. (1997). *Cultures and organisations – software of the mind*, UK: McGraw Hill.
- Holmes, H., and Tangtongtavy, S. (1995): *Working with the Thais*, White Lotus, Thailand.
- Jarvenpaa, S. L. and Leidner, D. E. (1999). 'Communication and trust in global virtual teams', *Journal of Computer-Mediated Communication*, 3 (4): <http://www.ascusc.org/jcmc/vol3/issue4/jarvenpaa.html> [online accessed 20 September 1999].
- 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*, (January), 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.

- Lipnack, J. and Stamps, J. (1997). *Virtual teams: reaching across space, time, and organizations with technology*, John Wiley & Sons: NY.
- Lindloff, T (1995). *Qualitative Communication research methods*, Sage: Thousand Oaks.
- Loucopoulos, P. and Karakostas, V. (1995). *System requirements engineering*, McGraw-Hill Book Company Europe: UK.
- Louis, M. R. and Sutton, R. (1991). 'Switching cognitive gears: From habits of mind to active thinking', *Human Relations*, 44, pp. 55-76.
- Macaulay, L. A. (1996). *Requirements engineering*, Springer-Verlag: UK.
- Markus, M. L. (1994). 'Finding a happy medium: explaining the negative effects of electronic communication on social life at work', *ACM Transactions on Information Systems*, 12(2): 119-149.
- Mockus, A. and Herbsleb, J. (2001). 'Challenges of global software development', In *Proceedings of 7th International Software Metrics Symposium*, (4-6 April): 182-184.
- Pliskin, N. and Romm, C. T. (1996). 'The role of e-mail in an industrial dispute', *Australian Journal of Information Systems*, 3(2): 47-56.
- Romm, C. T., Pliskin, N. and Rifkin, W. D. (1996). 'Diffusion of e-mail: an organisational learning perspective', *Information & Management*, 31: 37-46.
- Sommerville, I. and Sawyer, P. (1997). *Requirements engineering – a good practice guide*, John Wiley & Sons: UK.
- Sproull, L. and Keisler, S. (1986). 'Reducing social context cues: electronic mail in organizational communication', *Management Science*, 32(11): 1492-1513.
- Thanasankit, T. (1999). *Towards understanding the impact of culture on requirement engineering – ethnographies of Thai systems analysts*, Unpublished PhD Thesis, University of Melbourne: Australia.
- Thanasankit, T. and Corbitt, B. J. (1999). 'Toward an understanding of the impact of Thai culture on requirements elicitation', In Harris W H (Ed) *Proceedings of the Conference on Information Technology in Asia*, (16-17 September), Malaysia: 420-440.
- Thanasankit, T. and Corbitt, B. J. (2002). 'Understanding Thai culture and its impact on requirements engineering process management during information systems development', *Asian Academy of Management Journal*, 7(1): 103-126.
- Trompenaars, F. (1993). *Riding the Waves of Culture*, London, UK: Nicholas Brealey Publishing.,
- 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*, (27-19 March), IEEE Computer Society Press: USA.