

Risk Management Strategies for Intercultural Factors in Global Software Development

by

Eve MacGregor

B.A.Sc., University of British Columbia, 2004

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

The University of British Columbia

October, 2007

© Eve MacGregor, 2007

ABSTRACT

More and more development effort is spread across the globe in a variety of project configurations. In order to effectively manage these projects software project risk analysis must be expanded to include factors that are unique to projects that span distances, time zones and national and organizational cultures. This thesis describes a qualitative exploratory case study within a company that was initially outsourcing to a software house in India and later moved that development to an offshore office in China. This case is one of several that were part of a multi-pronged research effort exploring the effects of culture in a Global Software Development (GSD) environment. The interview questions explored the day-to-day work process of project personnel through a detailed conversation about their daily work life and their opinions about how the project went or was going. The analysis of participant interviews took a Grounded Theory approach. This thesis explores the issue of risks related to culture from two perspectives; a top-down approach wherein the literature in sociology and anthropology give insight into the concept of culture and into socio-cultural models and a bottom-up approach whereby the case study results are synthesized into practical recommendations. The results are two-fold. The first is a set of propositions that are useful for the identification and planning phases of risk management. The second is a summary of the risks encountered in the case under study along with associated strategies and the socio-cultural model concepts and indices that are related.

TABLE OF CONTENTS

| | |
|---|------|
| ABSTRACT | ii |
| LIST OF TABLES | vi |
| LIST OF FIGURES | vii |
| ACKNOWLEDGEMENTS | viii |
| INTRODUCTION | 1 |
| 1.1 PROBLEM STATEMENT | 2 |
| 1.2 RESEARCH QUESTIONS..... | 3 |
| 1.3 ORGANIZATION OF THIS THESIS | 5 |
| CHAPTER 2 BACKGROUND AND PREVIOUS WORK | 6 |
| 2.1 THE EMERGENCE OF GLOBAL SOFTWARE DEVELOPMENT | 8 |
| 2.2 WHAT IS CULTURE? | 9 |
| 2.3 SOCIO-CULTURAL MODELS | 11 |
| 2.3.1 Kluckhohn & Strodtbeck..... | 11 |
| 2.3.2 Hofstede..... | 12 |
| 2.3.3 Hall..... | 14 |
| 2.3.4 Trompenaars and Hampden-Turner..... | 15 |
| 2.4 GLOBAL TEAMS DEFINED | 16 |
| 2.5 THE IMPORTANCE OF CULTURE IN SOFTWARE ENGINEERING | 17 |
| 2.6 BEYOND CULTURAL MODELS | 18 |
| 2.7 COMPUTER SUPPORTED COOPERATIVE WORK (CSCW) | 19 |
| 2.7.1 Computer Mediated Communication..... | 19 |
| 2.7.2 Trust | 22 |
| 2.8 RISK IDENTIFICATION OF INTERCULTURAL FACTORS..... | 23 |
| 2.9 QUALITATIVE RESEARCH METHODS | 24 |
| 2.10 SUMMARY OF THE LITERATURE | 25 |
| CHAPTER 3 METHODOLOGY | 28 |
| 3.1 RESEARCH METHODS..... | 29 |

| | |
|--|-----------|
| 3.2 OTHER CASES | 31 |
| 3.3 DATA COLLECTION | 32 |
| 3.4 CASE STUDY SETTING | 33 |
| 3.5 DATA ANALYSIS | 35 |
| 3.5.1 Codes and coding..... | 36 |
| 3.6 VALIDITY | 38 |
| 3.6.1 Construct Validity..... | 38 |
| 3.6.2 Internal Validity | 38 |
| 3.6.3 External Validity..... | 38 |
| 3.6.4 Reliability | 39 |
| 3.6.5 Data Accuracy..... | 39 |
| CHAPTER 4 FINDINGS | 40 |
| 4.1 IMPORTANCE OF SHARED CODE REPOSITORY & BUG TRACKING..... | 41 |
| 4.1.1 When Quality Becomes an Issue..... | 42 |
| 4.2 ALIGNMENT OF ARCHITECTURAL & TEAM BOUNDARIES..... | 45 |
| 4.3 THE ROLE OF PUSHBACK | 47 |
| 4.3.1 Pushback and Retention | 51 |
| 4.4 TURNOVER | 51 |
| 4.5 INDIVIDUAL PERFORMANCE | 53 |
| 4.6 OTHER FACTORS | 54 |
| 4.6.1 Cultural Approach to Self-Assessment | 54 |
| 4.6.2 Culture-Specific Domain Knowledge | 55 |
| 4.7 SAMPLE CODE NETWORK FOR THE CODE “PROBLEM” | 56 |
| CHAPTER 5 DISCUSSION..... | 57 |
| 5.1 PROPOSITIONS | 57 |
| 5.2 RISKS, STRATEGIES AND ASSOCIATED CULTURAL ELEMENTS | 57 |
| 5.3 RISK AREAS FROM THE LITERATURE..... | 64 |
| 5.4 THE RESEARCH QUESTIONS REVISITED | 64 |

| | |
|--|-----------|
| 5.5 LIMITATIONS OF THE STUDY | 65 |
| CHAPTER 6 CONCLUSIONS AND FUTURE WORK..... | 67 |
| REFERENCES..... | 70 |
| APPENDIX A – DATA COLLECTION..... | 73 |
| A.1 INTERVIEW PROTOCOL | 73 |
| A.2 CODES WITH FREQUENCY..... | 77 |
| A.3 CONCEPTUAL MAPS OF TOPICS AND THEMES FROM EACH INTERVIEW | 79 |
| A.4 BREB CERTIFICATE | 85 |

LIST OF TABLES

| | |
|--|----|
| TABLE 2.1: POWER DISTANCE (PDI) AND INDIVIDUALISM (IDV) INDICES FOR CANADA, CHINA AND INDIA..... | 14 |
| TABLE 2.2: THE EFFECTS OF DIFFERENT AFFORDANCES ON COMMUNICATION BEHAVIOURS AND PROCESSES. SOURCE: (WHITTAKER, 2003)..... | 20 |
| TABLE 3.1: SUMMARY OF 5 CASES FOR WHICH DATA WAS GATHERED. | 32 |
| TABLE 3.2: PARTICIPANT ROLE DESCRIPTIONS..... | 34 |
| TABLE 3.3: RISK MANAGEMENT CODES THAT WERE DEVELOPED BEFORE ANALYSIS BEGAN. FEW OF THESE WERE ULTIMATELY ASSIGNED TO PORTIONS OF THE INTERVIEW TEXTS..... | 37 |
| TABLE 5.1: SUMMARY OF STUDY PROPOSITIONS..... | 57 |
| TABLE 5.2: SUMMARY OF RISKS WITH ASSOCIATED STRATEGY AND RELEVANT CULTURAL ELEMENT | 59 |

LIST OF FIGURES

| | |
|---|----|
| FIGURE 1: ICEBERG MODEL. SOURCE: (BRAKE <i>ET AL.</i> , 1995), P. 33 | 11 |
| FIGURE 3: CONTINUOUS RISK MANAGEMENT. SOURCE: (CARR <i>ET AL.</i> , 1993), P. 4 | 24 |
| FIGURE 4: VISUAL REPRESENTATION OF THE LEVELS OF ANALYSIS IN THIS STUDY | 31 |
| FIGURE 5: SAMPLE CODE NETWORK WITH "PROBLEM" AS THE CENTRAL CODE | 56 |
| FIGURE 6: SOME RISK AREAS PREVIOUSLY IDENTIFIED IN THE CSCW, GSD, SE AND IS LITERATURE..... | 64 |

ACKNOWLEDGEMENTS

A heartfelt thank-you to my supervisor, Philippe Kruchten, whose caring and supportive mentorship made this research project wonderful learning experience. Financial support was provided by an NSERC grant awarded to Dr. Kruchten.

Thanks also to Yvonne Hsieh, for keeping me company on this journey, for teaching me more about culture and for always being willing to discuss anything.

Gratitude also goes to all the research participants for their willingness to share their experiences with us.

This research was conducted with a Certificate of Approval numbered B05-0596, issued by the UBC Behavioural Research Ethics Board.

CO-AUTHORSHIP STATEMENT

The study described in this thesis was conducted in collaboration with Yvonne Ying-Fan Hsieh of the UBC Software Engineering and Architecture Laboratory during the period 2004-2007. The collaboration is mostly during the data collection phase of this thesis study (as described in Chapter 3). The author of this thesis is responsible for all the data analysis presented herein.

Parts of Chapter 2 (Background and Previous Work) are from the papers listed below that were co-authored with Yvonne Ying-Fan Hsieh and Philippe Kruchten. The majority of the re-used text was from the 2005 papers and the author made a significant contribution to their writing, with the exception of the text in Section 2.3 Socio-Cultural Models the majority of which was written by Yvonne Ying-Fan Hsieh.

- Hsieh, Y., Kruchten, P., & MacGregor, E. (2006). *Matching expectations - when culture wreaks havoc with global software development*. Paper presented at the Work Beyond Boundaries: An International Conference on Tele-mediated Employment and its Implications for Urban Communities, Vancouver, BC.
- MacGregor, E., Hsieh, Y., & Kruchten, P. (2005a, May 16, 2005). *Cultural patterns in software process mishaps: Incidents in global projects*. Paper presented at the Human and Social Factors of Software Engineering (HSSE), St. Louis, Missouri, USA.
- MacGregor, E., Hsieh, Y., & Kruchten, P. (2005b, May 1-4). *Intercultural factors in global software development*. Paper presented at the Canadian Conference on Electrical and Computer Engineering, Saskatoon, Sask.

INTRODUCTION

In recent years, many companies in the software development industry have begun outsourcing and/or off-shoring all or part of their software development work. It has become the industry norm rather than an exception. The development of any piece of software carries with it a set of risks. These risks are multiplied when that development extends into a context that has temporal, geographic and cultural boundaries. There are risks inherent to any development effort due to the degree of uncertainty and of change with regard to requirements, design aspects, management practices and organizational processes. A critical aspect of the successful execution of any software development is the team's ability to clearly communicate and coordinate expectations in a way that those expectations can be successfully met. One key element of communication that is limited within distributed projects is informal communication. Without this informal communication, there is little opportunity to establish relationship, determine the skill sets and backgrounds of co-workers, discover conflicting assumptions, incorrect interpretations or unanticipated dependencies and establish social networks and shared understanding.

To date the research on the communication and coordination of global software development (GSD) has not adequately addressed the role that intercultural factors play in globally distributed projects. Cultural factors are present in a variety of behaviours and thought processes such as; how we conceptualize, how much information we include (or do not include) in any communication, our sense of time and our sense of hierarchy.

These cultural factors affect both formal and informal communication as well as our approach to the coordination and execution of work.

The focus of this thesis is the *risks* inherent in a global software development environment that have a *significant intercultural aspect*. Because cultural elements cannot be separated from the context within which they exist, the appropriate approach to investigating these factors is to examine actual cases wherein the factors are encountered within their context. The study presented in this thesis is *exploratory* and the case is *representative* of GSD projects. The analysis takes a grounded theory approach, so that the theories developed stay close to the context, and use the language of the participants.

1.1 Problem Statement

Engineers and computer scientists are all technically trained, with differing foci depending on a variety of factors including, where they were formally educated. An investigation of the socio-cultural literature, and the growing body of literature related to technology and globalization shows that despite the pragmatic focus of technical training people issues, and by extension intercultural factors, remain significant (Carmel, 1999; Laroche, 2002; Nicholson & Sahay, 2001; Olson & Olson, 2004). In the literature on outsourcing, a number of writers and researchers note the presence of cultural factors, but treat the issue in a cursory fashion (Carmel, 1999; Karolak, 1998; Yourdon, 2004). General recommendations about the importance of the development process or the need for good communication and coordination strategies are not enough for today's practitioner. In order to gain greater understanding of the role that intercultural factors play in GSD, this exploratory study begins by looking at the specific problems that

occurred in specific projects and the *ad hoc* problem solving strategies that were undertaken *on the fly* by project personnel.

The analysis discussed in this thesis is one part of a multi-pronged effort. Yvonne Ying Fan Hsieh and I interviewed 15 participants, some of them twice for a total of 21 interviews. The 15 participants represented 5 different GSD projects spanning 7 countries including; Canada, USA, Nepal, China, India, Russia and Italy. Of the 5 cases Case A and Case E comprised the majority of the interviews. Case A was a multi-million dollar Canadian-Italian subcontracting arrangement and was analyzed by Hsieh (2007). Case E, the study discussed in this thesis, was a Canadian company that first outsourced in India and subsequently opened an offshore office in China.

1.2 Research Questions

Current research on intercultural factors in software development teams has not developed enough to make it possible or meaningful to test previously developed theories or hypotheses. It is therefore necessary to ask exploratory questions in order to lay the groundwork for further research. As such, this study of Case E asks:

1. *What are the risks related to intercultural factors in globally distributed software development?*
2. *What risk management strategies can be undertaken to mitigate problems related to cultural factors in the GSD environment?*

This study recognizes that culture is inherently dynamic and multi-layered and that in any given situation a person may or may not respond with any specific “tool” they have available within their cultural “toolbox” (Swidler, 1986). This study does not aim to make any causal or predictive claims related to culture and specific behaviour.

This study first developed *a priori* constructs from the risk management literature and then attempted to utilize them to identify the presence or absence of typical risk management stages in the case described herein. It was found that the study participants spoke mostly in the language of emergent problems and the *ad hoc* strategies they employed to address these problems rather than specific risk management planning stages. As this analysis evolved, cultural aspects drawn from socio-cultural literature were identified in the incidents the participants discussed or within the themes that emerged during the axial coding stage.

The main findings of this study are encapsulated in 6 propositions:

| Number | Proposition |
|--------|---|
| P1 | Not having a shared code repository with individual visibility and equal access on all sides, where there is a mismatch in power distance, can negatively affect trust when quality issues arise. |
| P2 | Team and architectural boundaries along geographical lines can have negative consequences. |
| P3 | Personnel from a high power distance culture are less likely to naturally “pushback.” |
| P4 | Personnel from a low power distance culture assume that “pushback” is a natural and universal skill in the software industry. |
| P5 | The high turnover characteristic of the software development industry in India and China can make it difficult to build domain knowledge. |

| Number | Proposition |
|--------|--|
| P6 | Explicit individual productivity rankings in a culture with a collectivist orientation may undermine the team's sense of job security. |

1.3 Organization of this Thesis

The organization of the remainder of this thesis is as follows: Chapter 2 discusses the background and previous work in relevant fields; Chapter 3 describes the methodology used in this study; Chapter 4 more fully describes the study findings; Chapter 5 makes some recommendations for the practitioner and Chapter 6 concludes and makes recommendations for future work.

CHAPTER 2

BACKGROUND AND PREVIOUS WORK

In order to fully understand the subject of this thesis, this chapter will provide some background information and appropriate literature in relevant though highly diverse fields. The literature in Anthropology and Sociology provided guidance on how to approach and conduct an exploratory qualitative case study. This guidance included interviewing techniques, my role as a researcher in relationship to the research setting and discussion of methodological concerns regarding the study design, data collection and analysis. Intercultural studies are also found in both Anthropology and Sociology, with a more specific focus on conducting business in intercultural settings found in some of the Organizational Behaviour literature. These studies provided an understanding of the concept of “culture” as well as current socio-cultural models. This understanding of culture enhanced my awareness of my particular cultural bias, as a Canadian and a Software Engineer. I utilized these socio-cultural models as “conceptual anchors” (Brannen & Salk), rather than prescriptive frameworks, during the data analysis phase.

The fields of Human Computer Interaction (HCI) and Computer Supported Collaborative Work (CSCW) provided background on the challenges associated with communicating and coordinating across distances in computer mediated frameworks. The literature in Software Engineering (SE) and Informations Systems (IS) provided background on how qualitative methods have been utilized and/or adapted for studies

related to questions having a significant “human” component in the Information and Communication Technology (ICT) sector.

Many of the difficulties confronting today’s GSD environment have little to do with technical issues; rather, they are “human” issues that occur when extensive collaboration and communication among developers with distinct cultural backgrounds is required. Although project managers are anecdotally reporting that intercultural factors are affecting software practices and artifacts, and that they deserve more detailed study, little analytical research has been conducted in this area.

As the number of businesses engaging in GSD increases, attention has been directed to studying the problems challenging personnel in GSD efforts. Software researchers and developers have identified issues such as physical dispersion (Carmel & Agarwal, 2002), collaboration across distances (Olson & Olson, 2003), extra telecommunication requirements, and they have developed some tactical approaches to ameliorate these problems. One crucial problem has been overlooked in the examination to date of global software projects: the misunderstandings related to intercultural factors that are present when developers, managers, and executives from different cultural backgrounds work together.

The intercultural factors discussed here are not merely differences in the ways of eating, dressing, or speaking, but rather, they are the results of each culture’s fundamental value system that, explicitly or implicitly, governs its people’s thinking and behaviour. Most

people recognize that differences exist among cultures. What is often less obvious is the degree to which these differences affect the day-to-day working life of technical professionals. The variety of cultural backgrounds found among developers in global projects means that they are often faced with problems that arise from intercultural issues. These problems can lead to miscommunication, misunderstanding, frustration, and underutilization of talents, presenting significant risks to software development.

This research is based on the notion that there are, and will continue to be, intercultural factors that affect both co-located and distributed software development efforts. It should not be assumed that these factors are all negative. In fact, at least two project managers have reported that a good mix of cultural approaches was beneficial to their projects (Borchers, 2003; Kruchten, 2004).

The goal of this thesis is to provide project personnel with up-front knowledge of potential cultural differences, their influence on projects as well strategies for identifying risks associated with these differences. These strategies will assist in both palliating and mitigating existing and potential issues, leading to higher quality up-front risk assessment and on-going risk management.

2.1 The Emergence of Global Software Development

In the early 1990's software development on a global scale really started to take off (Krishna *et al.*, 2004). For reasons such as cost reduction, access to wider talent pools and a desire to focus on core competencies (Desai, 2002; Dibbern *et al.*, 2004), some companies traditionally based in North America and Europe began sending all or part of

their development efforts abroad. They opened their own offshore development facilities, outsourced offshore (Corbett, 2004; Karolak, 1998; Yourdon, 2004) or sub-contracted with international companies.

In 2001, Carmel and Agarwal (2002) noted that 203 of the U.S. Fortune 500 companies engaged in GSD activities. Since 1995, India, the poster child for software outsourcing destinations, has seen a 40% growth in its software production, and the country's revenue from IT-based and IT-enabled services is projected to jump from US\$12 billion in 2003 to \$US62 billion in 2009 (PricewaterhouseCoopers, April, 2004). Other countries, including the Philippines, Israel, Russia, and Chile, have all become active outsourcing destinations in the GSD phenomenon; China is emerging as a fierce competitor with India for the largely Western outsourcing/off-shoring market (Carmel & Agarwal, 2002).

2.2 What is Culture?

In order to fully understand our discussion of culture in the GSD environment and, specifically, to know how to manage the risks related to these factors, we must first understand the term *culture*. Cultural Anthropology and Sociology developed properly as fields in the early 20th century though there have been "travellers tales" since antiquity (Brewer, 2000). Both of these have provided a body of knowledge that serves as background to understand the topic at hand.

But first, a definition of the term *intercultural* used in the title of this thesis and its distinction from the term *cross-cultural*. *Cross-cultural* refers to something that covers more than one culture. An example might be "A cross-cultural study of software

developers in India and Canada” which would study developers in India and then developers in Canada and do a comparison. Whereas, “An intercultural study of software developers in India and Canada” could be a study of the experience of doing development in both places. The key part of “intercultural” is that it involves interaction and therefore is the appropriate choice for the discussion of this research.

All human beings carry within themselves patterns of thinking, feeling, and acting that are inherited or socially conditioned. These patterns are based on the values people hold true and, in turn, will affect their behaviour. Hofstede (1980, 1997) draws an analogy between humans and computers, and calls such patterns mental programs.

For the goal of this thesis, an appropriate definition of culture is provided by Psychologist Spencer-Oatey who says, culture is "a fuzzy set of attitudes, beliefs, behavioural norms, and basic assumptions and values that are shared by a group of people, and that influence each member's behaviour and her/his interpretations of the 'meaning' of other people's behaviour (2000).”

Culture is sometimes over simplified to “superficial” matters, such as the ways of dining, clothing, religious rituals, architectures, or sports. Such “visible” attributes are certainly constituents of culture; however, they only make up a small portion as the largest portion of culture is “invisible” and must be inferred. To illustrate the visible and invisible properties of culture, an iceberg representation (as shown in Figure 1) is often used.




Image removed for copyright reasons.

Figure 1: Iceberg Model. Source: (Brake *et al.*, 1995), p. 33

2.3 Socio-Cultural Models

There are a number of prominent culture researchers within the fields of Anthropology and Sociology including Kluckhohn & Strodtbeck, Hofstede, Hall, and Trompenaars & Hampden-Turner. A brief summary of some of their relevant work and pointers to further reading are presented in the following sections.

2.3.1 Kluckhohn & Strodtbeck

Kluckhohn & Strodtbeck (1961) study culture from the perspective of “value orientations.” They identify five areas in which all cultural groups have fundamental though differing beliefs. These value orientations represent how a culture views human nature, the relationship of its people with nature, time, individual or collective focus, and whether space is public or private. For each of the orientations, Kluckhohn & Strodtbeck identify three relative positions where a culture may stand.

- **Human Nature** People are born good, evil, or a mixture of both.
- **Person vs. Nature** People value their subjugation to nature, mastery over nature, or harmony with nature.

- **Time Sense** Priority is given to traditional customs, future plans, or present events.
- **Social Relations** Society is organized around a lineal hierarchy of authority, collateral interests, or individual goals.
- **Space** Business and life is conducted publicly or privately, or a mix of the two

2.3.2 Hofstede

One of the most widely cited cultural researchers is Geert Hofstede. Based on a large scale study of a variety of types of IBM employees located in over 40 countries, Hofstede developed a set of cultural indices. The indices are relativistic scales (continuums) for a culture's approach to power distance, individualism/collectivism, masculinity/femininity, uncertainty avoidance and long-term/short-term orientation (1980, 1997).

Power Distance: The power distance index measures the extent to which a culture embraces social inequality. In a culture with high power distance, there exists an established hierarchy of power, based on status, wealth, intellectual capacity, or other factors. A culture with low power distance, on the other hand, considers every individual as equal, despite difference in power, status or wealth.

Individualism/Collectivism: The individualism/collectivism index is based on how an individual perceives and is perceived in a culture: either as an independent entity, or as part of a tightly knit group. A highly individualistic culture is one where individual interests take precedence over collective ones and everyone is expected to look after himself/herself. A highly collectivist culture is one in which people are integrated into strong, cohesive groups. People are expected to give allegiance to the groups to which they belong.

Masculinity/Femininity: A more masculine culture has more distinct social gender roles (i.e., men are supposed to be assertive, strong, and focused on material success while women are gentle, caring, and concerned with quality of life). Gender roles in a feminine culture are more fluid (i.e., both men and women can be concerned with relationship, modest, tender, and focused on improving quality of life). This index is reflective of culture on a national, rather than personal, level.

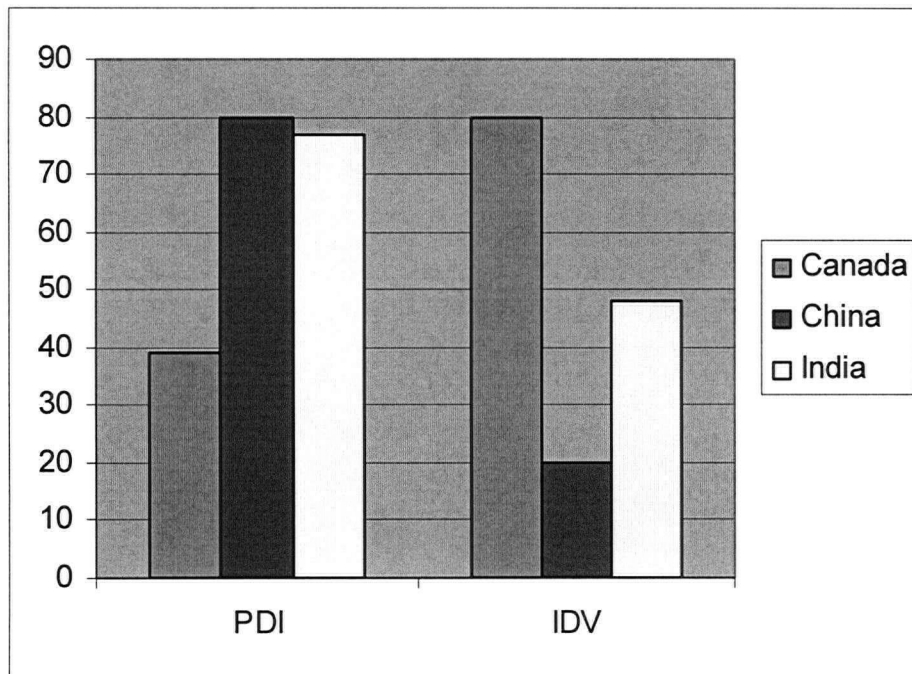
Uncertainty Avoidance: The uncertainty avoidance index indicates the tolerance a culture exhibits towards unfamiliar or ambiguous situations. A culture with a high ranking in this area may rely upon strict, detailed rules and procedures in order to mitigate uncertainty. A culture with low uncertainty avoidance is more comfortable handling unknown events and thus relies less upon rules.

Long-Term/Short-Term Orientation: This dimension did not appear in Hofstede's early work. It was added after a similar study that was carried out by researchers from Asian and Pacific countries. A culture with long-term orientation prescribes to long-term commitments and perseverance toward slow results. A culture with short-term orientation is more pragmatic welcoming to changes, and looking for rapid compensations, possibly to the detriment of the final outcome and the survival of the organization.

As an example of these indices, Table 2.1 below compares the Power Distance Index (PDI) and Individualism (IDV) Index for Canada, China and India. As can be seen in the table, India and China have a considerably higher PDI than Canada and for the IDV index India falls somewhere between China and Canada. A higher PDI means that there is more acceptance in China and India for hierarchical social structures whereas, a higher IDV

index correlates to a more individualistic society while a lower IDV index correlates to a more collectivist society.

Table 2.1: Power Distance (PDI) and Individualism (IDV) indices for Canada, China and India



2.3.3 Hall

Hall's research (1976) results in a dimensional model that examines culture from a more anthropological standpoint. The two dimensions we will discuss here are time (polychronic vs. monochronic) and communication patterns (high-context vs. low-context).

Time: In a monochronic culture time is managed in a linear manner: one event takes place at a time; a task is completed before another can be started. Activities such as meetings have definite start and end points and scheduling mechanisms are used to ensure that interruptions are avoided. A polychronic culture, on the other hand, considers

time to be much more flexible. Tasks can be handled simultaneously and interruptions are common.

Communication Patterns: According to Hall, a culture's communication patterns fall somewhere in the continua between high context and low context. In a low-context culture the speaker assumes that the message must contain all relevant information, including the context. The intention of the speaker is directly and unambiguously stated. In a high-context culture the speaker assumes that every participant in the conversation understands the context and thus complexity may be expressed in fewer words.

2.3.4 Trompenaars and Hampden-Turner

Trompenaars and Hampden-Turner's (1997) research builds upon that of Hofstede with a focus on the impact of intercultural variances on business and management processes. They conducted an empirical study with managers in multinational corporations and developed a set of seven value dimensions.

Universalism vs. Particularism: The universalism/particularism value orientation describes a culture's preference for rules or relationships. A universalist culture believes that established rules should be applied under all circumstances. A particularist culture focuses on the nature of the present circumstance and is more inclined to give special consideration based on the uniqueness of the situation or relationship.

Neutral vs. Emotional Expressions: The neutral/emotional dimension addresses the extent to which feelings can be openly expressed in a culture. Members of a neutral

culture are reluctant to reveal their feelings and thoughts in public while people in a more emotional culture may more plainly display feelings and thoughts.

Specificity vs. Diffuseness: This dimension measures the degree of involvement. In a specific-oriented culture, each (business) relationship is precisely defined around its limited context. Co-workers are less likely to establish a relationship outside of the work context. In a diffuse culture, however, more areas of life are interconnected, with arenas like work and family life intermingling.

Achievement vs. Ascription: An achievement-oriented culture accords status by performance while an ascription-oriented culture judges status based on a variety of factors including birth, kinship, gender, age, and seniority.

2.4 Global teams defined

Our traditional notion of “team” is people who work together in a collocated environment on a common task. However, the “global” component of GSD means that software teams are usually composed of members located around the world.

In this distributed context, teams are referred to as “virtual” and/or “global.” Leonard et al. describe virtual teams as interacting, having bounded social systems whose members are interdependent, being collectively responsible for the group’s product and sharing responsibility for managing their own work (1998). Because virtual teams can exist

within a single country or even within a building, Gibbs goes further in order to distinguish global teams:

...global teams are defined as crossing boundaries - cultural, geographical, temporal, and organizational. They are both virtual and multicultural. They also rely predominantly on the use of electronic communication rather than face-to-face interaction, due to their geographical dispersion. Team members are at least somewhat interdependent, to the extent to which they work collaboratively to complete common tasks. Finally, the team's strategy and work itself is global in scope (Gibbs, 2002).

2.5 The Importance of Culture in Software Engineering

Gibbs described some of the findings from a long-term case study of a global software team that was spread across locations in North America, Europe, Asia and South America. Her ethnographic study, which consisted of participant observation, in-depth interviews, focus groups and document and email analysis, found that “The global team was loosely coupled due to team members’ multiple cultural identifications, geographical dispersion, time differences and electronic rather than face-to-face communication.” (2002) This “loose coupling” had some positive effects but also created process and structure insufficiencies.

Greg Borchers analyzed two projects involving teams with members in Japan, India and the United States. He concluded that “The impact of cultural factors on our software engineering efforts caught us off-guard.” (2003)

Despite the fact that software engineering is a highly analytic process, culture and intercultural factors are playing an often unrecognized role in projects. While there are intercultural factors in teams that are co-located, the potential impact of intercultural factors in teams that are not co-located is much greater. “The largest factor in Global teams that we have seen is that they come from different cultures” (Olson & Olson, 2003).

Grinter et al. suggest organizing the project’s structure around the most difficult coordination problem (1999). Other researchers have suggested that global project structures should be designed to minimize intercultural communication as a risk management strategy (Krishna et al., 2004).

2.6 Beyond Cultural Models

In the context of culture and global software development research, there are a number of critics of nationalistic cultural models, particularly of Hofstede’s work. In IS research Myers & Tan argue that when multinational organizations are considering cultural differences in the development and use of information and communications technology, models of national culture are “overly simplistic” (2002). They further argue that nationalistic cultural models have “little explanatory power” and that IS researchers should see culture as contested, temporal and emergent. Walsham has also said that we need to move beyond these models toward theoretical frameworks which take into account the dynamic nature of culture (2002).

In their study of negotiated culture in a German-Japanese joint venture, Brannen & Salk said “Results show that aggregate models of cultural difference are useful only to the extent that they serve as latent conceptual anchors guiding individuals’ cultural responses to events (2000).”

2.7 Computer Supported Cooperative Work (CSCW)

CSCW is defined as combining “the understanding of the way people work in groups with the enabling technologies of computer networking” and also issues related to the design of such technologies (Wilson). Software development is a highly creative and collaborative effort and, as such, there are a number of issues involved in successful computer supported collaboration across distances and through mediated communication channels. These issues include, but are not limited to, establishing and/or maintaining trust, finding common ground, supporting informal communication, knowledge management and organizational memory.

Technology mediated communication is a fact of life. The human communication apparatus is constrained in several ways. There are limits to the distance at which speech is audible, and visible behaviours such as gesture, gaze or facial expressions are perceivable. Furthermore, these natural communication behaviours are ephemeral and do not persist over time. Given these limitations, we must rely on some form of mediation, if we are to communicate at distance and across time (Whittaker, 2003).

2.7.1 Computer Mediated Communication

Many of the problems identified with computer-mediated communication (CMC) are useful for identifying potential problem areas for distributed teams.

Cultures interpret communication behaviours such as facial expression, gaze and gesture in different ways, but there are core communicative phenomena that can be looked at on a human level. Table 2.2, below, shows communication behaviours affected by mode of communication available (i.e., visual or interactive modes which afford different communicative behaviours).

Table 2.2: The effects of different affordances on communication behaviours and processes. Source: (Whittaker, 2003)

Table removed for copyright reasons.

Cognitive cueing theories address such issues as turn-taking, conversation initiation and the role of a shared environment. Each of these three issues has distinct properties that are

likely to affect most cultures, though perhaps in different ways. Gaze and gestures support general turn-taking mechanisms, but how gaze and gesture are interpreted is culture-specific. For example, eye-widening is interpreted as surprise in western countries and anger in some African countries (Johnstone, 2002). In co-located teams there is visible information regarding availability for impromptu meetings or casual conversation, which is missing in a distributed environment. Research has found that 15% of co-located developer time is spent in informal communication (Perry, Staudenmayer and Votta, as reported in (Grinter et al., 1999)). This informal communication, a.k.a. “water cooler talk,” has proven to be important for decision-making, maintaining an understanding of the system design and social relationship building. Shared environments also permit conversational inference by “allowing certain types of reference such as deixis¹” (Whittaker, 2003). As reported in Whittaker’s literature review of computer-mediated communication, lack of persistent feedback and/or disrupted speech communication leads to over-elaborate messages. When combined with differences in communication styles (high/low context), mediated mediums that have delayed or diminished feedback capability may magnify intercultural communication issues.

Because communication is not only for the exchange of explicit knowledge, social cueing theories add another dimension to the research on mediated communication by looking at the role of interpersonal information. In team situations, the exchange of tacit knowledge and socio-emotional feedback are both important for “norming” type information

¹ According to the Oxford English Dictionary *deixis* is “Indication, pointing out”. As an example, when I turn my gaze toward the object about which I am speaking but could also consist of other physical and verbal references.

(Whittaker, 2003). The presence or absence of this type of feedback in communication channels, particularly culture specific “norming” information, must be kept in mind when evaluating intercultural communication in a computer mediated context. “Socialization and communication norms that occur naturally in co-located groups have to be consciously created when team members are physically dispersed.” (Leonard et al., 1998)

Research has suggested that companies should attempt to deal with distance by organizing the project structure around the most difficult coordination problem (Grinter et al., 1999). In distributed projects the constraints of the system architecture often mean assigning modules to teams that are co-located. The integration phase is then occurring across distances and cultural lines. According to Borchers “the amount of communication required between remote developers working on an integration area will quickly exceed their ability to communicate with each other (Borchers, 2003).”

2.7.2 Trust

The issue of trust arises in all teams and affects how teams function. If teams were co-located previously, they must rely on already established trust to carry them through the project. If the teams were not previously co-located, then there is the issue of establishing trust. There is evidence that trust is more “fragile” in situations where people are not interacting face-to-face (Olson & Olson). Establishing trust, or even the nature of trust, will differ from culture to culture.

2.8 Risk Identification of Intercultural Factors

The risks identified in this study can be utilized in the risk assessment phase of any GSD project, as will be discussed further in Chapter 5. To aid that discussion I will first give a description of the risk management process.

Risk management ranges from maintaining a simple “top 10” risk list to more involved processes such as the IEEE standard 1540-2001 (IEEE, 2001). Risk is defined as the possibility of some loss and can be thought of in terms of some form of *risk exposure* (potential loss times the probability of loss) (Boehm, 1991). Risk management is particularly necessary for GSD projects because the degree of uncertainty and change, with regard to requirements, design aspects and organizational processes, makes coordination more difficult (Sangwan *et al.*, 2007).

The type of risks investigated in this thesis are considered *operational risks* in that they are “Risk of suboptimal output that results from a variety of cases, including complexity of operations, geographic separation between client and vendor, and the limitations of the communications and transmission systems between the two.” (Aron *et al.*, 2005)

At its simplest, risk management consists of identifying, analyzing, planning, tracking, controlling and communicating risk. An initial baseline risk assessment can be followed by an ongoing continuous process as shown in Figure 2.

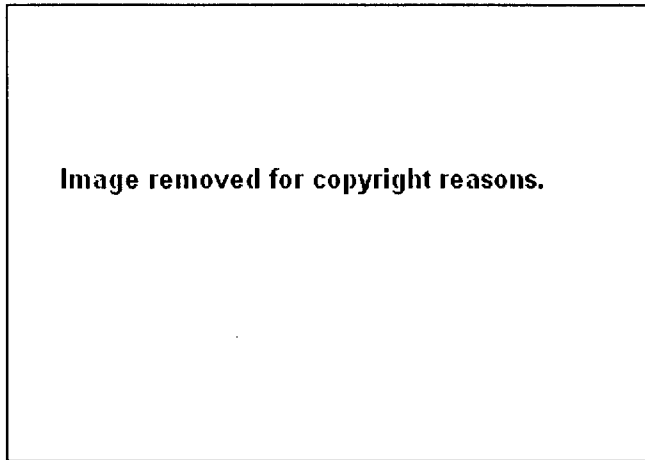


Figure 2: Continuous risk management. Source: (Carr *et al.*, 1993), p. 4

Once a risk has been identified, utilizing methods such as the one developed by Carr *et al.*, risk information can be turned into decisions and actions. These decisions and actions may take the form of mitigation, avoidance, acceptance or further study (Carr *et al.*, 1993).

Risks related to intercultural factors in GSD are, to date, not well defined. The risks enumerated in Chapter 5 (Recommendations for the Practitioner) broaden the currently identified risk areas in GSD.

2.9 Qualitative Research Methods

Qualitative research methodologies were first developed in the fields of Sociology & Anthropology (Denzin & Lincoln, 1994). These methodologies have proved useful, in their pure form or in appropriately adapted forms, for a number of fields relevant to this study, including; CSCW, software engineering (Sim *et al.*, 2000) and information systems research (IS) (Klein & Myers, 1999).

Traditional engineering research has been largely composed of quantitative methods. Practitioners in fields such as human computer interaction, CSCW and IS are increasingly turning to qualitative methods, such as ethnography, to answer questions about whether and how people are *actually* using systems. For a good overview of qualitative research methods and their role in the type of research described in this thesis see McGrath (1995). Qualitative methods are an appropriate choice when researching complex and dynamic human environments such as GSD projects. The study design and research methodology are discussed further in Chapter 3.

2.10 Summary of the Literature

Walsham (2004) comments that Hofstede's work assumes "cultural homogeneity," does not analyze work patterns and does not pay enough attention to the dynamic nature of culture. There are other critics of Hofstede (McSweeney, 2002). Nonetheless, his work, as well as that of other culture researchers, provides a good framework for beginning to understand intercultural issues.

There are a number of researchers who have recognized that culture in global teams is a phenomenon to be taken seriously (Krishna et al., 2004; Maznevski & Chudoba, 2000; Nicholson & Sahay, 2001; Olson & Olson, 2003, 2004; Walsham, 2001). In global software development there are several elements around which misunderstandings may occur, including: intercultural insensitivity (Maznevski & Chudoba, 2000), challenges in mediated communication (Whittaker, 2003), planning and management (Banerjee, 2003),

work-style differences (Walsham, 2001) and power, hierarchy and agency (Nicholson & Sahay, 2001).

Researchers have suggested the following strategies for ameliorating the effects of these problematic elements:

1. Strategically choose projects to be outsourced. Outsource offshore only the development of those components that are not culturally sensitive, such as middleware or a component to be embedded in an operating system. Such projects may reduce the risk of intercultural problems as they can be specified in a culturally neutral way (Krishna et al., 2004).
2. Accept projects only in which the benefit of the knowledge gained outweighs the cultural risks (Krishna et al., 2004).
3. Try to outsource projects where an “effective in-depth working relationship” can be established, such as when there is cultural closeness. “This match relates not merely to linguistic closeness but also to compatible ways of working and understanding user attitudes” (Krishna et al., 2004)
4. Create a “negotiated culture” within the project to reduce the effects of intercultural mismatches (Olson & Olson, 2003).

While all these suggestions have merit, they are not always practicable. In reference to the third and fourth suggestions, some cultural mismatches are inevitable because the real world is less than ideal; the suggested solution of creating a negotiated culture is weakened by physical dispersion, mediated communication and human factors. GSD

projects function in a distributed, multi-cultural environment and creating a negotiated culture in such settings is difficult. For risk assessment purposes it is therefore necessary to identify, as is done in this study, communication and coordination risk factors associated with software engineering in a distributed and intercultural setting.

CHAPTER 3

METHODOLOGY

The study described in this thesis explores the issue of cultural factors and their effects on Global Software Development efforts. The research question is examined through asking study participants to describe their experience in a specific GSD project, including both the “good” and the “bad”. The analysis in this study aims to produce risk identification and mitigation recommendations for the practitioner.

The study examines a particular case and fits Yin’s definition of an appropriate empirical inquiry because it “investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.” (1994), p. 13. In this case, the phenomenon is cultural factors and the real-life context is the GSD environment.

In general the research and analysis had the following flow:

1. Ask about how things went
2. Identify risks and problems
3. The risks go in the “risk list”
4. Work backward from the problems (i.e., figure out what would have been the risks before they became “problems”)
5. Add those risks to the “risk list”
6. Identify the strategies that were adopted once the problem emerged

7. Work backward from the strategies to find palliation and mitigation strategies

3.1 Research Methods

The goal of this case study is to develop a better understanding of the risks related to working cross-culturally in this particular context and what strategies have been employed to ameliorate those risks. This goal presupposes a theoretical approach; that there are cultural factors that will present risks for GSD projects and that there are appropriate risk management strategies for those factors. This theoretical approach necessitates the risk management and socio-cultural literature informing the analysis of this case.

The research questions ask *what* risks exist related to cultural factors and *what* risk management strategies can be taken. These “*what*” questions, combined with a lack of previous studies mean that an exploratory case study is an appropriate approach (Yin, 1994).

Gaining access to global software project personnel is difficult for two reasons. Firstly, there is a general unwillingness in software companies to talk to outsiders about projects where there are problems and secondly, they are reticent for intellectual property reasons. For these reasons and because of the relatively young nature of GSD Yin says a single-case, where multiple project personnel are interviewed can be considered *revelatory*. The structure of the relationships within the case (it contains both an outsourcing relationship with a company in India and later an offshore office in China) means that it can also be considered *representative*.

A grounded theory approach is used for the analysis of the interviews which allows for the emergence of constructs and theory (Charmaz, 2006).

The study can be defined as an exploratory and representative, single-case embedded design (Yin, 1994). The embedded nature of the study refers to the analysis taking place at multiple levels. There are three levels of analysis in this study; the individual level, the incident level and the case level. At the *individual level* each study participant is asked for their recollections and perceptions of the project through a semi-structured interview. At an *incident level*, the interview of each participant is examined and issues or incidents that arose and were spoken about by at least two participants are identified and examined. The Critical Interview Technique, as described by Chell (1998) in her adaptation of Flanagan's Critical Incident Technique (1954), contributed to the study questions and was applied cyclically as incidents identified in one interview were asked about in subsequent interviews. At the *case level* the study examines the entire "story" that emerges. These multiple levels of analysis are represented visually below in Figure 3

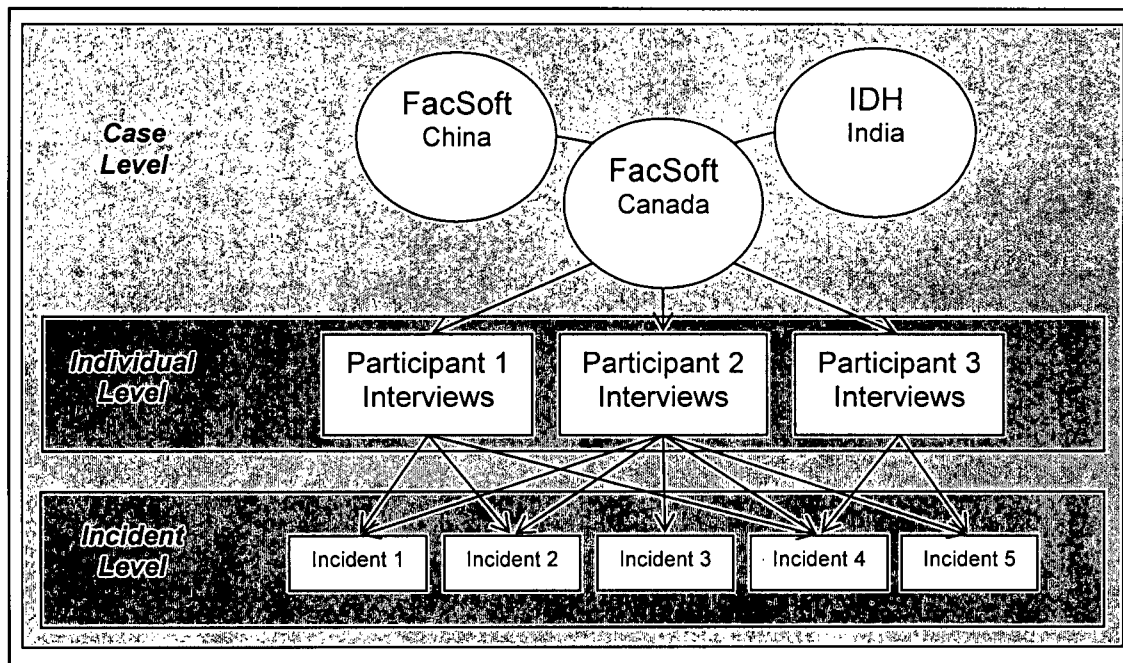


Figure 3: Visual representation of the levels of analysis in this study

The study is presented as a descriptive case study because the story, as a whole, is illuminative and the issues (or “incidents”) that arise are best illustrated by a description of the issue along with relevant quotes from study participants to highlight the description. In an effort to steer away from unnecessary “methodological acrobatics” (Sandelowski, 2000), the study can best be described as an exploratory and descriptive case study that takes a grounded theory approach to coding and analysis.

3.2 Other Cases

This study is one part of a multi-pronged research effort that investigates the role of culture in global software development efforts. Along with the case that is the subject of analysis in this thesis, another researcher, Yvonne Ying-Fan Hsieh, and I collected data for four other cases. Case A, described below, was analyzed by Hsieh (2007).

Table 3.1, shown below briefly describes each case, the number of study participants and the number of interviews conducted.

Table 3.1: Summary of 5 cases for which data was gathered.

| Case | Description | # of Subjects | # of Interviews |
|------|---|---------------|-----------------|
| A | Case A examines the relationship between a Canadian company that has a sub-contracting arrangement with two firms in Italy | 6 | 9 |
| B | Case B focuses on a contract software development company located in Nepal. The company has clients located in the USA and Europe. | 1 | 1 |
| C | Case C focuses on a outsourcing relationship between a USA-based company specializing in security technologies and a Russian software consulting company. | 1 | 1 |
| D | Case D examines a software company with headquarters in the US and a major development centre in Canada. The Canadian site further outsources to an Indian consulting company. The case focuses on the interactions between the Canadian and the Indian site. | 1 | 1 |
| E | Case E is the case analyzed in this thesis and is described in Section 3.3 | 6 | 9 |

I chose to analyze Case E because of the richness of the data, which was mostly due to the number of study participants we were able to interview.

3.3 Data Collection

Data collection took place between October 2005 and September 2006. During that time Yvonne Ying-Fan Hsieh and I conducted a total of 21 interviews, with 15 participants, from 6 companies about 5 different GSD projects. The audio for each interview was digitally recorded and later transcribed by one of us utilizing the transcription program *f4*

(Kollan *et al.*, 2004-2005). Some interviews we conducted on our own. For all of the interviews that were analyzed in both cases (Case A and Case E) we conducted the interviews together with a single study participant. During each interview we shared the task of asking questions that were prepared by us beforehand and each took notes which we later compared.

3.4 Case Study Setting

The study that is analyzed in this thesis was conducted in a company that has headquarters in California and offices around the world including Canada, Australia and China. In order to preserve confidentiality, I have created pseudonyms for the companies and the study participants and have changed the specific city locations of the company offices. FacSoft develops facilities management software for clients, a more detailed description of the software's functionality is not possible as, to do so, may violate confidentiality. The interviews took place at the company offices in the Vancouver area. At the time of the first interview, FacSoft had been in a 5-year outsourcing arrangement with a development house in India, called IDH, which has a local office in Vancouver. At that time FacSoft was considering opening an offshore development centre in Chengdu, China. By the second round of interviews they had ended their outsourcing arrangement with the company in India, were working with their own offshore personnel and were in the process of hiring 20 new developers and testers; bringing the total number of employees in Chengdu to 26. The developers and testers in China, who speak Mandarin, were being managed by a native Mandarin speaker, based in the company's Vancouver office who alternated two weeks in Chengdu and two weeks in Vancouver.

Another researcher, Yvonne Yin-Fan Hsieh (2007), and I interviewed three people, twice for a total of six interviews. In the preliminary stages of this research I conducted three other interviews in this case study. These three were considered part of the learning phase and were not transcribed, nor included in any subsequent analysis. This initial stage allowed us to refine the list of questions and the interview protocol. As examples, subsequent to those three interviews we decided to interview only one person at a time and only those people who were somewhat close to the day-to-day communication and coordination of the project. See Table 3.2 for a description of the study participants whose interviews were analyzed.

Table 3.2: Participant role descriptions

| Name | Role Description |
|------|---|
| Ajay | Development Lead for both the team in Vancouver and the team in India. At the time of his first interview he had been the main communication conduit between the two teams for four or five months. Prior to his role as the sole development lead he was one of two leads for two smaller teams working with IDH. At the time of his first interview he had been employed at FacSoft for 5.5 years. Prior to that Ajay had worked at a company that was acquired by FacSoft. Later in the case study when the off-shore centre was opened in China his role shifted to a Development Lead for a collocated team in Vancouver. |
| Jena | She was a lead developer in Vancouver until the company moved to open the offshore office, at which time she was given the role of Operations Manager . At the time of her first interview Jena had been with FacSoft for 6 years. As OM, she was responsible for hiring, training and mentoring personnel in Chengdu and also project handling, which includes day-to-day operations. She split her time evenly between the two development sites in two week increments. At the time of her first interview there were 6 developers and testers, but the company was soon to hire 20 more. Jena is originally from Taiwan and speaks Mandarin. |

| Name | Role Description |
|------|--|
| Alex | As Director of Product Development , Alex was one step removed from the daily management of the development teams. He was responsible at a high level for managing the outsourcing relationship with IDH. He was also largely responsible for the process of opening the off-shore centre in Chengdu and spent time in Chengdu establishing the HR process, conducting interviews and hiring personnel. |

3.5 Data Analysis

Using a grounded theory approach I coded each interview in several stages utilizing the qualitative analysis program *Qualrus* (Idea Works). “Coding means naming segments of data with a label that simultaneously categorizes, summarizes, and accounts for each piece of data.” (Charmaz, 2006)

Utilizing the methodology first described by Glaser & Strauss (1967) I did a first pass on each interview employing open coding. Open coding is done quickly, almost instinctively, and involves the identification and labelling of concepts and categories (Charmaz, 2006). Approximately 50% of the codes that I applied at this stage were *in vivo*, codes that simultaneously keep the analysis in the language of the study participants (Coffey & Atkinson, 1996) and allow for new ideas to emerge from the analysis without over reliance on previous theories, ideas or concepts (Charmaz, 2006). Following the first pass I wrote a text-based memo summarizing each interview that included themes and relationships. These memos began the next stage of analysis.

The next stage of analysis involved interleaved and iterative “chunks” of analysis. During this stage I employed axial (Strauss, 1987) and selective coding. Axial coding is the

process of developing categories and making connections between those categories and sub-categories. Selective coding involves the integration of the categories developed during axial coding (Charmaz, 2006). Both of these coding processes involve stepping away from the data and finding relationships between the codes while simultaneously categorizing and relating those categories to sub-categories. Once the analysis of all six interviews was concluded I had a total of 544 instances of 142 codes. Partly in response to the sheer number of codes and code instances and in order to “step away” from the data I decided to represent visually the partial results from the axial and selective code stages. This visual representation took the form of conceptual maps and can be found in Appendix A.3.

3.5.1 Codes and coding

Prior to the analysis, I designed a set of 16 category codes specifically related to the risk management literature. There are six continuous activities performed during the risk management lifecycle (as discussed in Section 2.7, on page 23); identify, analyze, plan, track, control and communicate. These activities inform the codes defined below. The idea was that these codes would be utilized during the axial and selective stages of coding. Not all of these pre-defined categories proved useful, as none of the study participants spoke specifically in risk management terms, nor could the stages of risk management be identified in the interviews. During the axial coding stage only a limited number of the codes listed in Table 3.3 were assigned. The codes should also have been developed with more parallelism, though ultimately this would not have changed whether or not they were assigned. The codes that were assigned related to identifying a risk as a

problem after it had already occurred or identifying the *ad hoc* strategy they implemented as something they could have planned beforehand.

Table 3.3: Risk management codes that were developed before analysis began. Few of these were ultimately assigned to portions of the interview texts.

| Construct | Explanation |
|-------------------------------|--|
| Unidentified | A problem that came up that went unidentified in the risk management lifecycle and even post-mortem they did not identify it as a risk/problem |
| Identify - <i>a priori</i> | A potential risk they identified up front |
| Identify - <i>posteriori</i> | A risk they identified after the problem |
| Magnitude | Anything about the magnitude of the exposure |
| Analyze – higher exposure | A risk they identified up front but had greater effect than they expected |
| Analyze – lesser exposure | A risk they identified up front but had lesser effect that they expected |
| Analyze – <i>a posteriori</i> | Exposure they would have assigned to a risk they identified after the problem |
| Planning – up front | This is the planning they did to mitigate risks. “Up front” here is used to mean before the problem, rather than at the beginning of the project. This means that “up front” could occur at any stage of the development lifecycle as long as identification of the risk precedes occurrence of the problem. |
| Planning – wish | This is the planning they wish they had done |
| Tracking | This is the risk tracking they did |
| Tracking – hindsight | This is the tracking they wish they had done. Things they would have tracked but did not, or things they would have tracked differently |
| Correction – <i>a priori</i> | These are the corrections they applied to risks that were identified beforehand |
| Correction – hindsight | These are the corrections they would have applied given what they know now |
| Correction – <i>ad hoc</i> | These are the corrections they applied on-the-fly for problems that emerged (i.e., for risks that went unidentified until they turned into problems) |
| Communication | What they did to communicate about risk through the project |
| Communication – hindsight | What they would have done to communicate about risk given what they know now |

3.6 Validity

Validity concerns, as discussed by Yin, that may affect the soundness of the exploratory case described in this thesis, are addressed below (1994).

3.6.1 Construct Validity

In a case that investigates the effect of some change, care must be taken to establish the soundness of the measures of that change (Yin, 1994). As this case does not develop any pre-defined measures with regard to intercultural risk in GSD, no effort is needed to justify the validity of such constructs.

3.6.2 Internal Validity

Usually, internal validity is only of concern for explanatory case studies as the results are intended to be causal explanations; either direct or inferred (Yin, 1994). However, the theory-building techniques of grounded theory are a form of explanation building and as such let the data hang together in such a way that the “truth value” of the results can be considered sound (Charmaz, 2006; Yin, 1994).

3.6.3 External Validity

According to Yin’s definition of analytic generalizability, the theories, or propositions, developed in this study can be tested through replication and, on their own, the propositions point toward further research (1994). The theories developed in this study may not be generalizable (i.e., you may not find exactly the same phenomena in other

GSD projects) but the recommendations in Chapter 5 should be generally useful for project personnel.

3.6.4 Reliability

In order to address the issue of reliability, the study protocol is documented in this thesis. This protocol includes the interview procedures, the initial set of questions for the semi-structured interviews and a description of the analysis process.

3.6.5 Data Accuracy

Data accuracy is related to all of the above mentioned validity concerns. Due to access, resource and time constraints, interviews were the only source of data gathered and therefore data convergence was not possible. Some degree of triangulation was achieved through interviewing multiple subjects using the same study questions and utilizing the first round of interviews to ask participants to expand on incidents of interest. This included asking participants during the second round of interviews about incidents that were identified by other participants in the first round of interviews.

CHAPTER 4

FINDINGS

There were numerous incidents in this case study where there was a mismatch in expectation regarding quality, individual performance and push-back. Many of the findings from this case study resolve around a particular issue such as quality, team dynamics, the interplay of hierarchy and communication, team and architectural boundaries, individual performance, turnover and culture specific domain knowledge. The dynamics around these particular issues often include a mismatch in expectation whether it is an expectation of a certain level of quality or a particular style of self-assessment. There were aspects of these issues that were easily identifiable, what was not so obvious was that both the sources and ramifications of these mismatches had farther reach than was immediately evident.

First, the issues that each participant spoke about and were found to affect the project significantly are described here along with relevant quotes from the study participants. These issues are then formulated into a set of propositions. Secondly, there were some issues that were mentioned by only one or two of the study participants. These issues are also described. Finally, a sample code network that resulted from the final selective coding stage is shown, the results are summarized and limitations of the study are noted. To assist in reading the quotes of the study participants, readers should know that the symbols [.] mean an approximate 5 second pause with additional periods indicating longer pauses. Also the symbol / indicates an incomplete sentence.

4.1 Importance of Shared Code Repository & Bug Tracking

There are a variety of configurations for the organization of and access to the content of essential project tools like the code repository and bug tracking system. Some software development projects keep their code in separate sub-repositories according to project location (with modules and or parts of the code they are working on) and then upload code to a larger repository on some pre-defined schedule. Others have a single repository which all project personnel work from, checking out and checking in code as needed. The checkout and check-in frequency can vary and has a pattern that is dictated by the workflow and/or organizational processes specific to each project and/or organization. FacSoft prefers to have a single global code repository. The developers are expected to work closely with the code repository with check-out/check-in repeated frequently during a single day. For both IDH and the offshore office in Chengdu, bandwidth and network latency issues meant that the remote teams work a little disconnected from the code repository, often with a single check-out/check-in and the beginning and end of their workday. Similarly, there are a variety of configurations for bug tracking systems. FacSoft also prefers to have a single, global bug tracking system.

With IDH there was no individual visibility in either the code repository or bug tracking system, meaning that unless the person doing the commit or entering a defect specifically put a note in the commit message, no one at FacSoft necessarily knew who had authored any piece of code.

I talk about them being part of our team. It's a little different in that we don't always know even individual names in every case /

-Alex

Part of the reason that there was no individual visibility in the bug tracking system was that IDH tended to funnel the updates to the bug tracking system through one person. This lack of individual visibility was exacerbated by the sense of power distance in India:

And they'll be even monitoring their communication and even, in some cases, for the newest people there they'll even filter for umm you know for advice or you know, preview before they'll fire off an email.

-Alex

Because we do have this kind of interesting observation before with our Indian outsourcing [firm]. They have everything very structured. They recorded every single thing: their MSN conversations reported to their team leads. And I even hear in China / they say India outsourcing / they never report to the North American office unless the manager is in the CC list. It's just their culture. They don't reply to their customers directly without their managers in the list.

-Jena

The pattern of reporting every communication to their team lead is significantly different than the communication pattern at FacSoft. There it is expected that communication goes unmonitored and that everyone can ask anyone anything.

4.1.1 When Quality Becomes an Issue

There was a significant issue with the team in Vancouver attributing poor quality code to the whole team at IDH. Alex, Jena and Ajay all talked quite a bit about how the visibility of the individual affected the dynamic around code quality. The premise is that if a team member can see who made the change, fixed the bug or wrote the code then that person is more likely to associate the quality of that section to the individual responsible rather than to the whole team. The thought process around this issue became, colloquially, 'if you can not see who did it, then they all did it; and if it is bad, then they are all bad':

I think it's the team as a whole. I don't think our developers here really knew exactly who made the change or / to that detail.

-Ajay

and then it becomes engraved in everyone's mind that their quality is bad. And once that sets in it's difficult to / like every little thing / not little thing, everything / that you notice is / [the same].
– Ajay

The lack of individual visibility combined with a high level of power distance on the IDH side, which constrained free communication between team members at IDH with team members at FacSoft, meant that feedback mechanisms within the global team were inhibited. The visibility in the code repository as well as the channelled communication meant that when quality issues came up the developers and testers in Vancouver did not know who to attribute the work to and also had no channel for direct feedback to individuals. This resulted in attributions of poor quality being made to the whole team in India. What started out as localized judgements such as "I don't understand why they have done it this way" or "there is a mistake here" snowballed into a general belief by the Vancouver team that the team in India produced poor quality code. The same constrained communication and lack of individual visibility also meant that when clarification was needed regarding a specific piece of code or bug the team in Vancouver did not know to whom they should direct their questions. In an effort to see what was really going on, the management looked at bounce rate (bug reported fixed, but it actually is not fixed) statistics and found that there was no difference between the two teams.

/ you show them the data then you can say, you know it's not statistically any different than our team here so what's / what else is going on in your mind. What else is causing you to uhh you know, paint them all with the same paint brush or something like that.
-Alex

Some of the quality issue was just the difference in experience level. Parts of the team at IDH stayed with the project for years and were considered by Ajay to be just as good as

the local team, but this was offset by the turnover at IDH (the issue of turnover is explored further in Section 4.4)

It's just the nature of the beast. It doesn't mean they're less capable or not and these are the kinds of conversations we have.

-Alex

The problem with them is, you know, we have maybe 3 or 4 developers that's been there [in India] and who has worked with us for a while, but the rest they re-assign, you know ... it's difficult, you know, to get the right answers from them. So you have to know 'I should ask certain developers down there, and they'll know,' that type of thing.

-Ajay

It's just one of those things you've got to manage right, from a perspective of making sure the team still stays engaged ... a thing I watch for really clearly is watching for when people are / are picking up on those types of things and then translating them into uhh other umm [...] other umm, I don't know, opinions or whatever / ... and actually saying you know / turning it into assessments of quality of work or something like that right? When the reality of it is, it's other things and so you try to watch, and that sort of thing.

-Alex

FacSoft utilized the lessons they had learned through working with IDH when setting up workflow and communication patterns between the personnel in Vancouver and China. A specific strategy they undertook for the work dynamic between the team in China and the team in Vancouver was to give everyone full access and visibility and to use exactly the same tools. This helped the dynamic and ease of work flow but there were still some issues:

They know exactly who has done what, individual check-ins and that sort of a thing umm, bug tracking systems the same things. They seem to have the same issues [as India], they tend to funnel the updates of the bug tracking system through one person.

-Alex

None of the people at FacSoft offered a theory for why the remote teams funnel bug tracking updates through one person.

The issue of not having a shared code repository and bug tracking system, combined with constrained communication and a power distance mismatch, affected project progress and the level of trust that FacSoft team members had for IDH team members. Thus, I propose that:

| | |
|------------|--|
| P1: | Not having a shared code repository with individual visibility and equal access on all sides, where there is a mismatch in power distance, can negatively affect trust when quality issues arise. |
|------------|--|

4.2 Alignment of Architectural & Team Boundaries

Conventional wisdom in the GSD literature says that the architecture of the system should allow modules to be assigned along geographic boundaries. The idea is that if there is loose coupling between the modules and each module is assigned to a team that is co-located then this minimizes the need to communicate and coordinate across time zones and cultures (Herbsleb & Grinter, 1999). In the case of FacSoft there were other ramifications that should be considered before following this advice explicitly.

For the team located in Vancouver the most problematic phase was stabilization and integration. As would be expected, during that part of the lifecycle the amount of communication back and forth increased significantly because of the need to resolve problems or fix bugs. Because the modules had been assigned along geographic boundaries that meant that there was no one local who could answer critical questions in a timely way:

But their way would be / they would have questions coming back and going back and forth. So I almost had to / "OK someone in-house has to take care of this." So I would reassign it to someone here.

-Ajay

The team member's sense of where the team's boundary is also had consequences for the sense of inclusiveness including, their willingness to accept the consequences of the quality of other team member's work:

The only thing was when the quality was bad, that was a huge issue for us here, because our developers would feel like "OK, I am cleaning up someone else's mistakes.

-Ajay

Both the team boundary and differences in the sense of power distance meant the communication channels between the teams were limited. This led to the Vancouver team feeling like there was a disconnect between them and the people at IDH because the two locations did not have the same sense of the time pressure the project was under:

They're not in our email trails, they don't get all the communication that's coming back and forth here. We only tell them what they need to know. So they don't really feel what we feel here as far as "OK, we have this deadline.

-Ajay

/ where I don't think they really understood how critical it is within something small like that and the urgency / that was a big thing for us because they didn't feel the urgency like coming down to a release where we would have 20 defects and we need to get it done because it has to be in by this Friday.

-Ajay

The conventional wisdom regarding the architecture of the system may be sound, but project managers should keep in mind the scenario above. Careful consideration must be given to the consequences of having no "local experts" if the stabilization and integration is to happen at one site. In this case Conway's Law (1968) held true in the sense that the

effectiveness of the team was constrained by the absence of good communication paths to the necessary experts. Given the foregoing, I propose that:

P2: Team and architectural boundaries along geographical lines can have negative consequences.

4.3 The Role of Pushback

“Pushback” is a term that Alex, Ajay and Jena used during the course of our interviews. Its meaning seems to be understood in the software industry but does not appear to be formally defined in this context. In their terms, it is an interactive process whereby project personnel ask questions until they fully understand, question decisions about the assignment of tasks, requirements, designs and coding strategies and make suggestions about the how, when and why of requirements, design and code, until everyone has reached understanding and/or agreement about how to proceed. Pushback goes farther than “feedback” in that it is a bi-directional negotiation and could possibly be seen as a form of consensus building. Pushback, at least by a North American definition, goes at least one step farther than “feedback” in that its central purpose is to establish common ground, context and understanding thereby contributing to forward movement in the development lifecycle.

Alex, Ajay and Jena all talked about pushback, or lack thereof, and its effect on the successful execution of the project. Alex said that had they not managed the lack of pushback from IDH it could have severely negatively affected the project:

We've actually had to explicitly coach them and their team, their senior team members there to not do that, you know. 'This bugs us when you do that' and 'what we expect of you is to actually understand all of this and push-back when it

doesn't make sense and ask lots of questions'. So that whole 'I don't want to say no to somebody' or 'I don't want to question what somebody's asking me' is umm, is something we've had to manage around and I think that had we not done anything about it / it could have been actually pretty bad.

-Alex

The issue of pushback is intricately connected to culture; specifically the notion of Power Distance:

Because they don't / they won't tell you 'okay we have a problem', right. I'm spending two days doing something when in actual fact we could do it in like half a day here, right

-Ajay

Because in China we find the culture part, like the shy to talk / to ask question to say "I don't understand" right and then the people repeat the / their sentence again is not part of the China culture is more, we found, than here. Even from university education here, we say "why" right, but in China everyone just nod, nod, nod (nodding her head) right. So you don't know that / these people that they understand fully.

-Jena

One of the effects of a lack of pushback is the costly implementation of requirements that have not been fully understood, but, in one of the instances described below, the spill over effect was that it also undermined the Vancouver team's sense of confidence in the remote team:

We've definitely had some cases where umm you know the interpretation of the written requirement was taken extremely literally and not questioned. Even though when you took the literal interpretation it came out to be something quite silly, right? 'Blah, blah, blah' text on a page or form or something like that, which was just a sample and ... actually got delivered in a couple of cases. And that's an example of the type of thing that a tester or developer would go 'what are they thinking' right?

-Alex

For example, we were doing the requirements for penny rounding. So for New Zealand they don't have the 2 cents. They don't have the 2 cents so they have to round up to 5 cents. So that's a requirement for rounding and then we have 6

testers there [in China], each of them write down the use cases for rounding and all of the numbers are different.
-Jena

When the development cycle gets close to a release if the remote team has not learned to pushback in the way that the North American team expects, stress and frustration can be exacerbated. This is partly because it is at this stabilization and integration stage that any requirements issues will be highlighted. If there was a “miss” or mistake in the requirements that went unnoticed by everyone, then it is accepted more readily, but if it is that the remote team has made a literal interpretation of the requirements or had questions that they did not clarify this is the stage where people will start saying “okay, this should have been discussed during the requirements review stage (Ajay)”. The magnitude of the risk in this scenario is quite high:

But the ones that get you / like big ones where you have to spend days fixing is going to be a miss in the SRS [System Requirements Specification] or a misunderstanding, that kind of thing.
-Ajay

Another aspect of pushback is the clarification of expectations. The team in Vancouver found that some requests produced results they had not expected:

We have had where their discretion seems to be at odds with what we would / you know they'll have come back having spent four days on something we would have expected somebody to spend 2 hours in the morning sometime. And then coming back to us, you know and they'll come back with a five page report, 'whoa where did this come from?'... our response has been to actually have them do less of that ... But the instinct becomes well 'let's just give it to somebody I know' umm 'I'll get what I expect.'
-Alex

These unexpected results lead to project personnel channelling requests to local people when they were not sure how the content of the request would be interpreted. As a manager, Alex coached people in how to be clearer about their expectations.

FacSoft expected, as a check and balance mechanism, that project personnel would pushback. It was therefore critical to project success that they manage the risk inherent in working with a culture that may have other mechanisms to achieve the same end. Overall, FacSoft found that they had to be more explicit in their requirements specifications and to be clearer regarding expectations when assigning tasks. They considered this a learning process and at the time of the second interview thought that their effort to come to an understanding with the IDH team had proven successful:

I think at the beginning it was like that, right. They were / whatever, they read the SRS and they thought "okay this is what they mean" and they would go ahead and do it and we've run into problems, in the past. But now if they have a question, they are really good at bringing up "how do we do it"
- Ajay

Based on the foregoing discussion, I propose that:

P3: Personnel from a high power distance culture are less likely to naturally “pushback.”

P4: Personnel from a low power distance culture assume that “pushback” is a natural and universal skill in the software industry.

In this case the implications of Propositions 3 and 4 for FacSoft were that the lack of pushback, and the assumption that pushback is a universal skill, created significant issues with regard to the execution of tasks during all phases of the project. The lack of pushback was particularly costly during the requirements review stage. Also, the

assumptions of the universality of pushback lead to unmet expectations, which in turn spiralled into dissatisfaction, frustration and mistrust.

4.3.1 Pushback and Retention

One of the practices that FacSoft has in place for the training of new team members in North America is a peer mentoring system. Each new developer or tester is assigned to someone more senior who is their “go-to” person if they have questions, problems or concerns. They did not have this mentoring process with IDH, but decided that it would be a good way to integrate the offshore personnel in Chengdu with the people in Vancouver. New team members in Chengdu have a local mentor and one in Vancouver with whom they have a weekly one-on-one meeting for 15 to 30 minutes.

We hope through these weekly one-on-one we can know more about these China folks, and then have a better retention. To understand their needs earlier rather than saying "they don't talk" and after a while they just say "I'm not going to do this, I'm leaving," right, so we have a more regular check.

-Jena

FacSoft undertook the mentoring process partly because they knew that they would need to make an effort to integrate their remote personnel. By giving them an avenue to talk about opinions or concerns related to work FacSoft hopes to have better overall retention.

4.4 Turnover

FacSoft found that there was a high turnover in project personnel on the IDH side. This turnover was both internal and external. Internally, IDH moved people between projects:

What I've noticed is that they're very / they want to get into new technology. So they [individuals] don't want to stay working on the same project. ... So the problem with our contract is they have to keep their, you know, employees happy so they don't want to keep the same people on the same project for a long time.

-Ajay

This high internal turnover combined with an external turnover² meant that it was difficult to build necessary domain knowledge among the developers and testers:

With IDH we had quite a few issues where knowledge / especially building the, what I would call subject matter experts there. They would bring in new developers all the time so they were not even familiar with our products, right. So it takes quite a long time to get them up to speed and really be productive.

-Ajay

A lot of that was again, kind of cultural. That was how they felt they would move up in their career / they get different types of jobs.

-Alex

You know IDH managed that pretty well umm but you're going to have more issues with a more junior team.

-Alex

While starting the offshore office in Chengdu Alex noticed that there were a number of applicant's resumes that showed they had moved companies frequently by his standards:

We're seeing some of that in people applying for jobs in China as well, a lot of different companies on their résumé. The ones that have had a few / you know 5 years experience or something like that. It's interesting because we'll look at that resume and see it as a red flag. Why? You know. They can't get along with people or they're not good enough or? You know, whatever, and they're not looking at it that way at all, it doesn't seem to be anyways and umm so we don't quite have enough experience at hiring people with that type of background yet to know whether or not it's an issue or not, whether we're interpreting it correctly or not but uh it's noticeable, the difference.

-Alex

I think culturally people are sorry / I think the team here is aware of some of the cultural differences around things like retention and attitudes around engagement in the job are a little or they're perceived to be anyways, a little bit different. ... it's like retention, like look at retention and say [the perception is] 'okay, they must not care very much about their, you know, about their work, this job, you now, these projects' and that sort of a thing because they're willing to go off to a different company and / for another 10% raise or something.

-Alex

² Alex thought the IDH external turnover was about 15-18% during the course of their project with IDH. NASSCOM (National Association of Software and Service Companies) cited the turnover rate in India for 1999-2000 as being ~35%, but said that number had fallen to 25% in 2000-01 and to 12% in 2001-2002. Source: <http://www.nasscom.in/Nasscom/templates/NormalPage.aspx?id=1292>

Based on the foregoing discussion on turnover and its effects, I propose that:

P5: The high turnover currently characteristic of the software development industry in India and China can make it difficult to build domain knowledge.

One consequence of Proposition 5 for FacSoft was that over time, senior developers and testers in North American began to question the ability of remote personnel and the quality of the code they produced.

4.5 Individual Performance

One of the metrics that FacSoft uses is the productivity of each employee. In North America they produce aggregate rankings that tell employees which range they are in; for example, they might be in the top or bottom 10%. For the new employees in China they were concerned that a collectivist approach to team, and the individual's role in that team, would not encourage the competitive approach to productivity that has been fostered in North America. During the hiring process, they vetted for a competitive sensibility and did not hire people whose personal goals were to be in the "middle of the group."

Jena said that culturally people did not want to stand out:

Because some people I interview in China "oh I just want to be in the middle of the group, I don't want to be the star". Because the star can be, from their point of view, can be reached, if there is something wrong then the star can be pulled out of first, right.

One of their strategies to address this “grouping” was to post explicit individual productivity rankings, despite the fact that this is not something they would do in North America. The focus was not on the ranks, but rather on the stars in an effort to show “the benefits of them showing their personal performance (Jena)” and have the bottom person improve.

These explicit rankings apparently had the desired effect, but some of the fallout from this system seemed to be more concern about job security.

For here people will feel embarrassed but then it will not be the kind of like thinking about job security, right. So I tell the China team "when you make a mistake as long as you recognize the mistake and don't make the same mistake again / it's not that when you make the mistake we are not going to use you any more.

-Jena

Given the foregoing discussion, I propose that:

P6: Explicit individual productivity rankings in a culture with a collectivist orientation may undermine the team’s sense of job security.

4.6 Other Factors

The issues identified in this section were not discussed with all study participants, but are included because they highlight relevant risk areas.

4.6.1 Cultural Approach to Self-Assessment

Jena noticed that there was a different approach to self-assessment in China as compared to North America. She said generally, if you ask a developer in Vancouver to assess their progress they will give you a list of stuff they have done in a very matter of fact way.

In China it's the opposite. When they say "what I have done", they will say "what I haven't done well" right but in their mind they still want me to encourage them, right, so I just stimulate them in a way that "okay, you need to tell me what you think you are doing well" right and then we'll start from there to have the constructive feedback because if they go for the bad part usually the meeting kind of sink down more and more and just go for the bad part and totally forget about the good part they have done.

-Jena

4.6.2 Culture-Specific Domain Knowledge

Domain knowledge is important when developing software. Some systems are designed for a context that is outside the cultural purview of employees in an offshore office:

So for them it's kind of difficult because in China they don't know 'oh, I can go to [a facility], book something.' So we have to just give them more and more business knowledge from here to there. So it's the business knowledge part that we are still working on it.

-Jena

Depending on the role project personnel are playing, culturally sensitive domain knowledge can be more or less important. In this case study Jena said that the developer role was less affected by this because the business rules are in the code. The testers were more affected because they were required to "think more from the customer's perspective":

Right, so the developer can look at all the logic saying "if case, else case" so they can look from the code to understand what the business is like. But for the testers it's not only to look at the documents but also need to put themselves into customer's shoes, right.

-Jena

To address this issue, the company had to find a mechanism by which the testers could learn this culturally rooted knowledge. In their case, they provided audio files of support calls for the testers in China.

4.7 Sample Code Network for the Code “Problem”

The open coding process pulled apart the data into concepts that were mostly expressed in the language of the study participants. The codes were then categorized and summarized which lead to the descriptions and propositions in the preceding sections. In the selective coding phase, code networks were built, using tools provided in the qualitative analysis software Qualrus' (Idea Works), showing relationships between the codes. A sample code network for the code “problem” is shown in Figure 4.

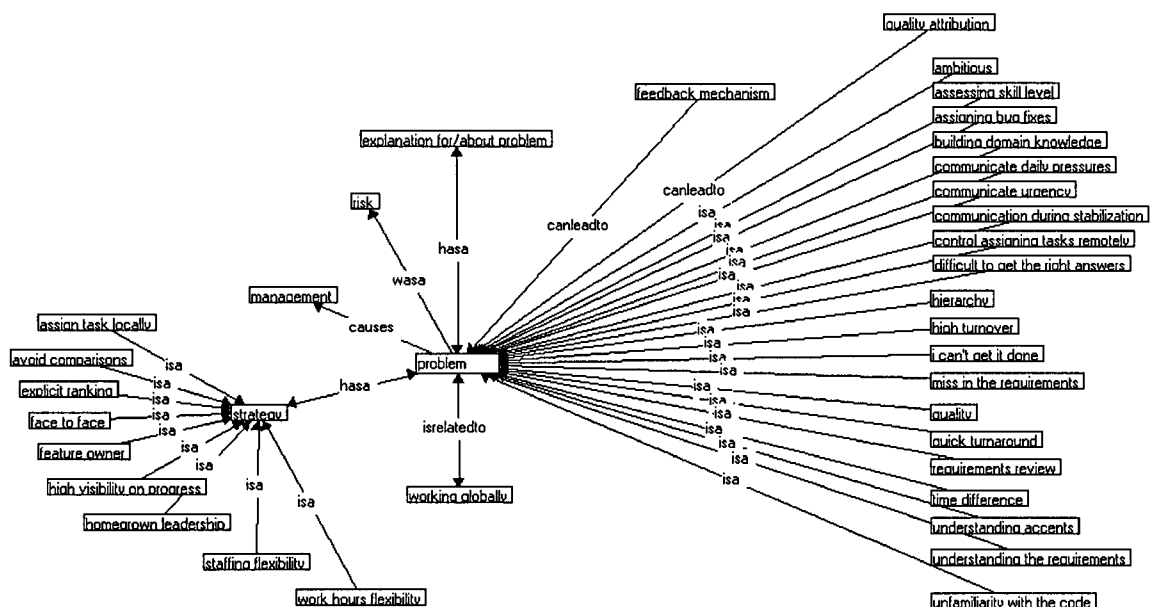


Figure 4: Sample code network with "problem" as the central code

CHAPTER 5

DISCUSSION

The contributions resulting from the foregoing study are presented in this chapter in a set of propositions, and in an enumeration of risks that have a significant cultural component. The propositions are summarized below in Table 5.1, and the risks identified in this study are presented along with mitigation strategies in Table 5.2.

5.1 Propositions

Table 5.1: Summary of study propositions

| Number | Proposition |
|--------|---|
| P1 | Not having a shared code repository with individual visibility and equal access on all sides, where there is a mismatch in power distance, can negatively affect trust when quality issues arise. |
| P2 | Team and architectural boundaries along geographical lines can have negative consequences. |
| P3 | Personnel from a high power distance culture are less likely to naturally “pushback.” |
| P4 | Personnel from a low power distance culture assume that “pushback” is a natural and universal skill in the software industry. |
| P5 | The high turnover characteristic of the software development industry in India and China can make it difficult to build domain knowledge. |
| P6 | Explicit individual productivity rankings in a culture with a collectivist orientation may undermine the team’s sense of job security. |

5.2 Risks, strategies and associated cultural elements

Found on the following six pages, Table 5.2 is a summary of the findings described in the previous chapter with an associated strategy and cultural elements that may affect, or be affected by, that particular risk/strategy combination. In the “Strategy Description”

column, the regular text refers to strategies FacSoft undertook to manage or mitigate the associated risk and the *italicized text* refers to recommendations that I would make based on the literature related to working in intercultural settings. Practitioners should keep in mind the limitations of the study that are presented at the end of this chapter.

Table 5.2: Summary of risks with associated strategy and relevant cultural element

| Risk & Description | Strategy Description | Cultural Elements |
|---|--|--|
| Lack of Pushback: Lack of pushback from some team members, and other team members assumption that this is a universal skill in software engineering | <ul style="list-style-type: none"> • Coach high power distance personnel to pushback • Train low power distance personnel to understand lack of pushback | <ul style="list-style-type: none"> • Power Distance • Individualism/Collectivism • High/Low context |
| Miss in the SRS or misunderstanding of the requirements: Anything that was missed in the initial SRS. Can also be aspects that were not clarified, or acceptance of SRS at face value | <ul style="list-style-type: none"> • Coach high power distance personnel to push-back • Train low power distance personnel to understand lack of pushback | <ul style="list-style-type: none"> • Power Distance • Individualism/Collectivism • High/Low context |
| Turnover: The high level of turnover in India and China; this turnover is both internal (within company, between projects) and external. This can have the spillover effect of NA personnel questioning commitment of remote personnel | <ul style="list-style-type: none"> • Instil company culture of engagement and commitment to long-term careers • Provide avenues to make suggestions or voice concerns (Create peer mentoring system) • Conversation with North American staff about the culture / context in India and China | <ul style="list-style-type: none"> • Power Distance |

| Risk & Description | Strategy Description | Cultural Elements |
|--|---|--|
| <p>Sense of Team: There are distinct differences in the sense of what makes a team, each member's role as an individual and their responsibility to the team</p> | <ul style="list-style-type: none"> • Carefully consider any measures to align the approaches • Explicit ranking to "unglue" the group • <i>May be counter-productive to try and change this to entirely NA approach</i> • <i>Understand and accept the different approaches and how they manifest in personnel's behaviour</i> | <ul style="list-style-type: none"> • Individualism/Collectivism • Power Distance |
| <p>Team Boundary: Team boundaries being restricted to geographical locations. More difficult to integrate teams, create sense of common purpose and feel the same urgency, confidence and accomplishment. Exacerbates communication problems during stabilization and integration</p> | <ul style="list-style-type: none"> • Personnel working together on teams that span geographical locations • FacSoft had 50-50 ratio in Chengdu and Vancouver • Open communication structure | <ul style="list-style-type: none"> • Individualism/Collectivism • Power Distance |

| Risk & Description | Strategy Description | Cultural Elements |
|---|---|--|
| Architecture along geographic lines: Assigning architectural module boundaries along geographic lines exacerbates communication problems during stabilization and integration. Means no local experts when integration is occurring at one site. | <ul style="list-style-type: none"> • Personnel working together on teams that span geographical locations • FacSoft had 50-50 ratio in Chengdu and Vancouver • Open communication structure | <ul style="list-style-type: none"> • Individualism/Collectivism • Power Distance |
| Work hours flexibility: High Power Distance and collectivism can mean that personnel will not feel comfortable leaving before their boss. | <ul style="list-style-type: none"> • Set work hours (even if it is something the company would never do in NA) | <ul style="list-style-type: none"> • Power Distance • Individualism/Collectivism |
| Unmet expectation: Any time where the results of a task were not what was expected. This is related to the amount of instruction people accept and/or expect to be given or to give. | <ul style="list-style-type: none"> • More explicit in instructions where there is a difference in communication context and power distance • <i>Coaching on how to make communication closer to the style of the receiver</i> • <i>May make low power distance personnel uncomfortable as they may feel they are being overly instructive</i> | <ul style="list-style-type: none"> • High/Low Context • Individualism/Collectivism • Power Distance |

| Risk & Description | Strategy Description | Cultural Elements |
|--|--|--|
| Culture Specific Domain Knowledge: In the case of FacSoft, they make software for facilities that are common in NA, though not so common in India or China. Means that personnel may have a harder time putting themselves in the customer's shoes | <ul style="list-style-type: none"> • Use training tools (in this case FacSoft is using recorded customer support calls so the team in China can better understand the domain) | |
| Misunderstanding of cultural approach to self assessment: Personnel in China and NA have different approaches to self-assessment. | <ul style="list-style-type: none"> • Coach personnel to "start with the good stuff" • <i>Accept that there are differences and not let this approach set a negative tone (for Individualist listener)</i> • <i>Accept that there are differences and not assume the person is bragging (for Collectivist listener)</i> | <ul style="list-style-type: none"> • Individualism/Collectivism |
| Individualized performance rankings: In China, in an effort to not stand out, personnel may shoot for the "middle ground" in terms of performance. Can be exacerbated if there is a mismatch between the values of the team member and the organization. This is also related to a person's sense of team. | <ul style="list-style-type: none"> • Careful strategies if trying to shift toward a more individualist approach (there may be unexpected ramifications) • <i>Feedback based only on personal performance (not on status) may have negative consequences</i> | <ul style="list-style-type: none"> • Individualism/Collectivism • Ascription/Achievement |

| Risk & Description | Strategy Description | Cultural Elements |
|---|--|--|
| Assigning open ended tasks: If no deadline is given, a remote team member may incorrectly assess expectations. If poor quality work is returned because of an assumption of urgency – the relationship can be harmed. | <ul style="list-style-type: none"> • Make the expectations explicit (see Unmet Expectation) | <ul style="list-style-type: none"> • Power Distance |
| Lack of individual visibility in code repositories (CR) and bug tracking systems (BTS) can have negative consequences if questions arise regarding quality. If quality becomes an issue, and there is no obvious “culprit” then the entire remote team may be blamed. This cycle, if initiated, can be difficult to interrupt. | <ul style="list-style-type: none"> • Give everyone full visibility in CR and BTS • Give everyone full access to CR and BTS • Keep the conversation positive • Provide feedback channels • Address any issues immediately | <ul style="list-style-type: none"> • Power Distance • Individualism/Collectivism |
| Tempting to offer individualized incentives – but in China salary etc. is very public. Can have negative consequences. | <ul style="list-style-type: none"> • Keep things public • <i>People will be asked and will share</i> | <ul style="list-style-type: none"> • Individualism/Collectivism • Diffuse/Specific |

5.3 Risk Areas from the Literature

Figure 5, below, summarizes some of the risks identified in a variety of bodies of literature.

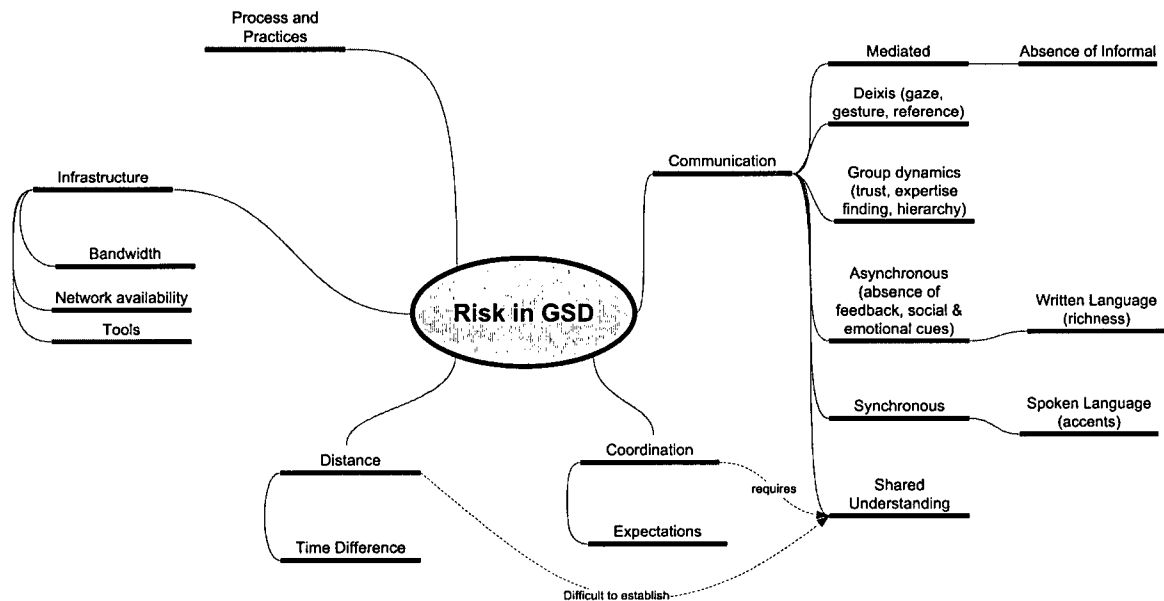


Figure 5: Some risk areas previously identified in the CSCW, GSD, SE and IS literature

5.4 The Research Questions Revisited

The research questions were first outlined in Section 1.2. The first research question addresses risks associated with cultural factors in GSD projects. The study finds that there are culturally based risks associated with globally distributed software work. Some of the risks are associated with the more obvious aspects of culture, such as language spoken or difficulty understanding accents. These risks are more apparent, and thus allow personnel to make quick adjustments and develop reasonable on-the-fly strategies for coping. The less apparent, but more important risks noted in this study have to do with Power Distance, Individualism/Collectivism, High/Low Context communication styles and Ascription/Achievement. The risks associated with these

cultural factors are often only visible through the consequences; such as, unmet expectations, misunderstood behaviour, miscommunication, and misinterpretation of requirements. The ramifications of such problems are farther reaching than can be immediately apparent and the magnitude of the risk associated with them is higher. Table 5.2 and the concept maps in Appendix A.3 enumerate the specific risks related to culture.

The second research question addresses risk management strategies that can be undertaken to compensate for problematic cultural factors in the global software development environment. The study finds that there are strategies that can be undertaken to manage the effects of cultural factors in GSD. The specifics of these strategies are listed above in Table 5.2

The propositions in Table 5.1 and the risk/strategy summary in Table 5.2 are most relevant in the identification and planning phases of risk management. The GSD environment is complex, having numerous variables that present significant challenges to software development and the associated risk management. Identifying risk in this global, multicultural context is part of the difficulty and should be considered a risk in-and-of itself.

5.5 Limitations of the Study

There are several limitations to the application of the results from this study. This is a single case within a specific context. The findings may not apply to other settings or project configurations. Though the study investigates both an outsourcing and an off-shoring relationship, the two relationships were at different stages. The outsourcing arrangement was coming to an end and the off-shoring arrangement was just beginning. As such, some of the gloss and energy from the

new arrangement or a natural loss of impetus for the “old” relationship may have affected the perspective of the study participants.

Another limitation is that, due to resource constraints of the study all the analyzed interviews were with developers at FacSoft, therefore the findings may have a Canadian or FacSoft bias. Conducting and analyzing interviews from personnel at IDH and in the off-shore office in Chengdu might have shed new light on the findings.

Lastly, I did not use any metrics to empirically evaluate the magnitude of the problems they encountered, nor the efficacy of the strategies they employed and, therefore, the study says nothing about the potential magnitude of the risks found in the study, nor about the potential benefit of applying those strategies.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This thesis presents an exploratory case study of culturally based risk in GSD that utilizes grounded theory methods for analysis. The results show that there are risks associated with cultural factors in the GSD context.

There are significant risks associated with a mismatch in the sense of Power Distance as it relates the dynamic around *pushback*. North American software engineers expect that clarification, questioning decisions, and suggesting alternative solutions are a normal part of the work dynamic. The results in this study suggest that this is *not a universal trait*. In fact, the lack of pushback from their counterparts in India and China led to stress, frustration and lessening of trust when aspects of the development cycle did not proceed as expected. In the case studied, FacSoft learned during the course of their outsourcing arrangement that they could not rely on the pushback dynamic to clarify assumptions, requirements specifications nor expectations. They took a two pronged approach by coaching North American personnel to adjust their communication style and also by successfully coaching the personnel in India about FacSoft's expectations regarding how and when to pushback.

Other areas that presented risk in the studied case were the *architecture*, *team boundaries* and *sense of team*. In particular, architectural and team boundaries along geographic lines can have unexpected downstream effects such as Conway's Law about system architecture being exacerbated by Power Distance. The higher sense of Power Distance at IDH contributed, at least

partially, to restricted communication channels between individual developers at FacSoft and IDH. This effect was particularly noticeable during the stabilization and integration phase when there were no “local experts” to answer questions in a timely fashion. The constrained communication channels that existed between FacSoft and IDH contributed (from the perspective of FacSoft personnel) to a sense of disconnectedness and had consequences for their sense of inclusiveness and their willingness to accept the consequences of the quality of remote team member’s work.

Lack of individual visibility in both the code repository and bug tracking system affected the ability of FacSoft personnel to accurately attribute the responsibility for specific pieces of code to individuals on the IDH side. This resulted in attributions of poor quality being made to the whole remote team, rather than just individuals. The lack of individual visibility in both these systems was a structural risk generated, in part, by a hierarchy mismatch and by the structure of the remote teams. The spill-over effect was that FacSoft personnel were unable to direct clarification questions to the appropriate person, which restricted feedback channels. As a strategy, FacSoft now ensures individual visibility in the code repository for the team members in Chengdu.

Future work would be particularly appropriate on the issue of pushback, as this was an ongoing issue that FacSoft had to manage through the course of their five-year outsourcing relationship with IDH. The question of how to successfully reduce the effects of the intercultural mismatch regarding pushback would be useful for all projects where there is a difference in the concept of Power Distance. This case study has a Canadian bias in that the participants work in Canada, the

researcher is Canadian and was trained at a Canadian university. Similar case studies conducted from the all the cultural perspectives represented in this study would also be interesting. This thesis provides a basis for further investigation of this issue and other risks associated with intercultural factors in global software development.

REFERENCES

- Aron, R., Clemons, E. K., & Reddi, S. (2005). Just right outsourcing: Understanding and managing risk. *Journal of Management Information Systems*, 22(2), 37-55.
- Banerjee, P. (2003). Narration, discourse and dialogue: Issues in the management of inter-cultural innovation. *AI & Society*, 17, 207-224.
- Boehm, B. W. (1991). Software risk management: Principles and practices. *IEEE Software*, 8(1), 32-41.
- Borchers, G. (2003). *The software engineering impacts of cultural factors on multi-cultural software development teams*. Paper presented at the 25th International Conference on Software Engineering (ICSE.03), Portland, OR.
- Brake, T., Walker, D. M., & Walker, T. (1995). *Doing business internationally: The guide to cross-cultural success*. New York: McGraw-Hill.
- Brannen, M. Y., & Salk, J. E. (2000). Partnering across borders: Negotiating organizational culture in a german-japanese joint venture. *Human Relations*, 53(4), 451-487.
- Brewer, J. D. (2000). *Ethnography*. Buckingham, UK: Open University Press.
- Carmel, E. (1999). *Global software teams: Collaborating across borders and time zones*. Upper Saddle River, NJ: Prentice-Hall.
- Carmel, E., & Agarwal, R. (2002). The maturation of offshore sourcing of information technology work. *MIS Quarterly Executive*, 1(2), 65-77.
- Carr, M. J., Konda, S. L., Monarch, I., & Ulrich, F. C. (1993). *Taxonomy-based risk identification* (Technical Report No. CMU/SEI-93-TR-6). Pittsburgh: Carnegie Mellon University.
- Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. London: SAGE.
- Chell, E. (1998). Critical incident technique. In G. Symon & C. Cassell (Eds.), *Qualitative methods and analysis in organizational research* (pp. 51-72). London: SAGE Publications Ltd.
- Coffey, A., & Atkinson, P. (1996). *Making sense of qualitative data: Complimentary research strategies*. London: Sage Publications.
- Conway, M. (1968, April). How do committees invent? *Datamation*.
- Corbett, M. F. (2004). *The outsourcing revolution: Why it makes sense and how to do it right*. Chicago: Dearborn Trade.
- Denzin, N. K., & Lincoln, Y. S. (1994). *Handbook of qualitative research*. Thousand Oaks, Calif.: Sage Publications.
- Desai, A. (2002). *Global software development--a cross-cultural relationship* (No. Unique ID: 291231). London, UK: London School of Economics.
- Dibbern, J., Goles, T., Hirschheim, R., & Jayatilaka, B. (2004). Information systems outsourcing: A survey and analysis of the literature. *SIGMIS Database*, 35(4), 6-102.
- Flanagan, J. C. (1954). The critical incident technique. *Psychological Bulletin*, 51(4), 327-358.
- Gibbs, J. L. (2002). *Loose coupling in global teams: Tracing the contours of cultural complexity*. Unpublished Dissertation, University of Southern California, Los Angeles.

- Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine.
- Grinter, R. E., Herbsleb, J. D., & Perry, D. E. (1999). The geography of coordination: Dealing with distance in R&D work, *Proceedings of the international ACM SIGGROUP conference on Supporting group work*. Phoenix, Arizona, United States: ACM Press.
- Hall, E. T. (1976). *Beyond culture*. New York: Anchor Books/Doubleday.
- Herbsleb, J. D., & Grinter, R. E. (1999). *Splitting the organization and integrating the code: Conway's law revisited*, Los Angeles, CA, USA.
- Hofstede, G. (1980). *Culture's consequences*. Beverly Hills, CA: Sage.
- Hofstede, G. (1997). *Culture and organizations--software of the mind*. New York, NY: McGraw-Hill.
- Hsieh, Y. Y.-F. (2007). *Shared understanding and the effects of culture in the global software development team - a case study*. University of British Columbia, Vancouver.
- Idea Works. Qualrus. from <http://www.ideaworks.com/qualrus/>
- IEEE. (2001). IEEE std 1540-2001 IEEE standard for software life cycle processes—risk management -description. from http://standards.ieee.org/reading/ieee/std_public/description/se/1540-2001_desc.html
- Johnstone, B. (2002). *Discourse analysis*. Malden, Mass.: Blackwell.
- Karolak, D. (1998). *Global software development: Managing virtual teams and environments*. Los Alamitos, CA: IEEE Computer Society Press.
- Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, 23(1), 67-93.
- Kluckhohn, F., & Strodtbeck, F. (1961). *Variations in value orientations*. Evanston, IL: Row Peterson.
- Kollan, F., Proeser, M., & Neubauer, H. (2004-2005). F4: <http://www.audiotranskription.de/f4>.
- Krishna, S., Sahay, S., & Walsham, G. (2004). Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, 47(4), 62-66.
- Kruchten, P. (2004). *Analyzing intercultural factors affecting global software development*. Paper presented at the (GSD2004) 3rd International Workshop on Global Software Development, Collocated with ICSE, Edinburgh, Scotland.
- Laroche, L. (2002). *Managing cultural diversity in technical professions*. London, UK: Butterworth-Heinemann.
- Leonard, D. A., Brands, P. A., Edmondson, A., & Fenwick, J. (1998). Virtual teams: Using information technology to manage geographically dispersed development groups. In S. P. Bradley & R. L. Nolan (Eds.), *Sense & respond: Capturing value in the network era* (pp. 285-298). Boston, Mass.: Harvard Business School Press.
- Maznevski, M. L., & Chudoba, K. M. (2000). Bridging space over time: Global virtual team dynamics and effectiveness. *Organization Science*, 11(5), 473-492.
- McGrath, J. E. (1995). Methodology matters: Doing research in the behavioral and social sciences. In R. M. Baecker, J. Grudin, W. A. S. Buxton & S. Greenberg (Eds.), *Human-computer interaction: Toward the year 2000* (pp. 152-169). San Francisco: Morgan Kaufmann Publishers Inc.
- McSweeney, B. (2002). Hofstede's model of national cultural differences and their consequences: A triumph of faith - a failure of analysis. *Human Relations*, 55(1), 89-118.
- Myers, M. D., & Tan, F. B. (2002). Beyond models of national culture in information systems research. *Journal of Global Information Management*, 10(1), 24-32.

- Nicholson, B., & Sahay, S. (2001). Some political and cultural issues in the globalisation of software development: Case experience from Britain and India. *Information and Organization*(11), 25-43.
- Olson, J. S., & Olson, G. M. (2003). Mitigating the effects of distance on collaborative intellectual work. *Econ. Innov. New Tech.*, 12(1), 27-42.
- Olson, J. S., & Olson, G. M. (2004). Culture surprises in remote software development teams. *ACM Queue*, 1(9), 52-59.
- PricewaterhouseCoopers. (April, 2004). A fine balance: The impact of offshore IT services on the Canadian IT landscape, executive summary. Retrieved November 5, [Last accessed] November 5 2004, from <http://www.pwd.com/ca/afinebalance>
- Sandelowski, M. (2000). Whatever happened to qualitative description? *Research in Nursing & Health*, 23(4), 334-340.
- Sangwan, R., Bass, M., Mullick, N., Paulish, D. J., & Kazmeier, J. (2007). *Global software development handbook*. Boca Raton: Auerbach.
- Sim, S. E., Singer, J., & Storey, M.-A. (2000, June 5). *Beg, borrow or steal: Using multidisciplinary approaches in empirical software engineering research*. Paper presented at the ICSE Workshop Report, Limerick, Ireland.
- Spencer-Oatey, H. (2000). *Culturally speaking: Managing rapport through talk across cultures*. New York: Cassel.
- Strauss, A. L. (1987). *Qualitative analysis for social scientists*. New York: Cambridge University Press.
- Swidler, A. (1986). Culture in action: Symbols and strategies. *American Sociological Review*, 51(2), 273-286.
- Trompenaars, F., & Hampden-Turner, C. (1997). *Riding the waves of culture* (2 ed.). New York: McGraw-Hill Trade.
- Walsham, G. (2001). Globalization and ICTs: Working across cultures (pp. 37). Cambridge, UK: University of Cambridge.
- Walsham, G. (2002). Cross-cultural software production and use: A structural analysis. *MIS Quarterly*, 26(4), 359-380.
- Walsham, G. (2004). Cross-cultural issues in global software outsourcing. In [Http://www.Almaden.Ibm.Com/institute/pdf/2004/geoff_walsham_1.Pdf](http://www.Almaden.Ibm.Com/institute/pdf/2004/geoff_walsham_1.Pdf).
- Whittaker, S. (2003). Theories and methods in mediated communication. In A. C. Graesser, M. A. Gernsbacher & S. R. Goldman (Eds.), *Handbook of discourse processes*. Mahwah, N.J.: Erlbaum.
- Wilson, P. (1991). *Computer supported cooperative work*: Kluwer.
- Yin, R. K. (1994). *Case study research: Design and methods* (2nd ed. Vol. 5). London: SAGE.
- Yourdon, E. (2004). *Outsource--competing in the global productivity race*. Upper Saddle River, NJ: Prentice-Hall.

APPENDIX A – DATA COLLECTION

A.1 Interview Protocol

Beginning of the interview

- Inform the subject of the content of the consent form
- If appropriate have them sign it
- Describe the semi-structured nature of the interview; questions are mainly a cue for memory
- Ask if it is okay to record the interview
- Turn on the recorder

Interview questions

Enabling Technologies

- Do you use any special ICTs to facilitate communication/collaboration?
- Does everyone involved in the project use the same tools/applications?
- Have you met with any resistance regarding usage of these tools?

Communication mode/tools

- What medium(s) do you use (e-mail, face-to-face, telephone, video conferencing, etc.)?
- Telephone
- Travel
- Email
- Direct
- Indirect

After questions about mode – maybe drill down a bit more

- During interactions who talks, what happens?
- Are there any examples of meetings that didn't work well?
- Who was "driving" the meeting?
- How formal/informal is your communication?

Communication structure

- Who do you communicate with?
- Between remote sites do you have designated proxies (and all communication must go through the proxies)?
- Do you have established, regular meeting time, or are meetings called as needed?

Information about communication style (point of contact)

Collocated

- Synchronous
 - Any mishaps using this method?
 - How effective is it?
- Asynchronous
 - What kind of communication do you use this for (passing passive information, generating new information, or reaching decisions)?
 - Any mishaps using this method?
 - How effective is it?

Not collocated

- Synchronous
 - Any mishaps using this method?
 - How effective is it?
 - The decisions made during the meetings - are they followed through?
- Asynchronous
 - What kind of communication do you use this for (passing passive information)?
 - Any mishaps using this method?
 - How effective is it?

How do you work together?

Each responsible for an area and only communicate to pass information?

Do you negotiate or collaborate on (division of tasks, design of system, schedule, more?)?

Kind of work being achieved

Frequency/regularity with which things happen

Possible scenarios to “regularity”

- Very rigid
 - Communicate with a regular pattern (e.g., every two weeks)
 - The project manager says this is how things are going to be
- Not rigid
 - Scrum
 - Happens over the phone
 - Regular and high frequency
 - Does not use email
 - Live

Problem resolution

- Can issues/problems be raised?
- When issues are raised what happens?
 - Is it written down?
 - Is anyone looking for a “culprit”?
 - Who raises issues?
 - How formal is the process?
 - What happens (is it: stored, written down, discussed, ignored)?
 - How are issues tracked? Who tracks the problems? Who solves the problems?
 - Does this happen through normal channels or is there some other mechanism?

Software practices

- Do you have a formal process (i.e., IEEE, ISO, CMM, 6-sigma)?
- Do you have a written process definition?
- Do you practice any of the Agile methods (XP, regular code reviews, change control board)?

Note: would not be practical to ask questions about the technical specifics such as, what language they are coding in, etc.)

- What software process expectations did you have going in to this project?
- What practices/processes do you use?
- What practices/processes do they use?
- Over the course of the project did anyone need to adapt their process/practices?
- If yes, did you adapt or did they?

Scheduling

- Who decides the schedule? Do the “doers” have any input to the schedule?
- How did the schedule go, or is it done entirely by the management/clients?
- Was there any correlation between the schedule and what actually occurred?
- Do you think that everyone thought the same way about the schedule in terms of its importance?

Documentation

- What is the purpose of documentation?
- Is the documentation followed through? Does documentation actually reflect the product?

Trust and miscellaneous

- What was the relationship between the source and destination?

- How would you characterize the level of trust between distributed sites? (e.g., Do you feel you're trusted? Do you trust the remote site?)
- Did that trust change over the course of the project?
- Was anything done to build up trust?
- How would you characterize the level of trust within your immediate team?
- What was the structure of the team?
- Were your site and the remote site considered one team or were they separate teams?
- Where there any mishaps or other incidents during the course of the project?
- The use of bridgeheads – are people sent to the remote site to “supervise” the projects? Bridgeheads act as managers or employees?

A.2 Codes with Frequency

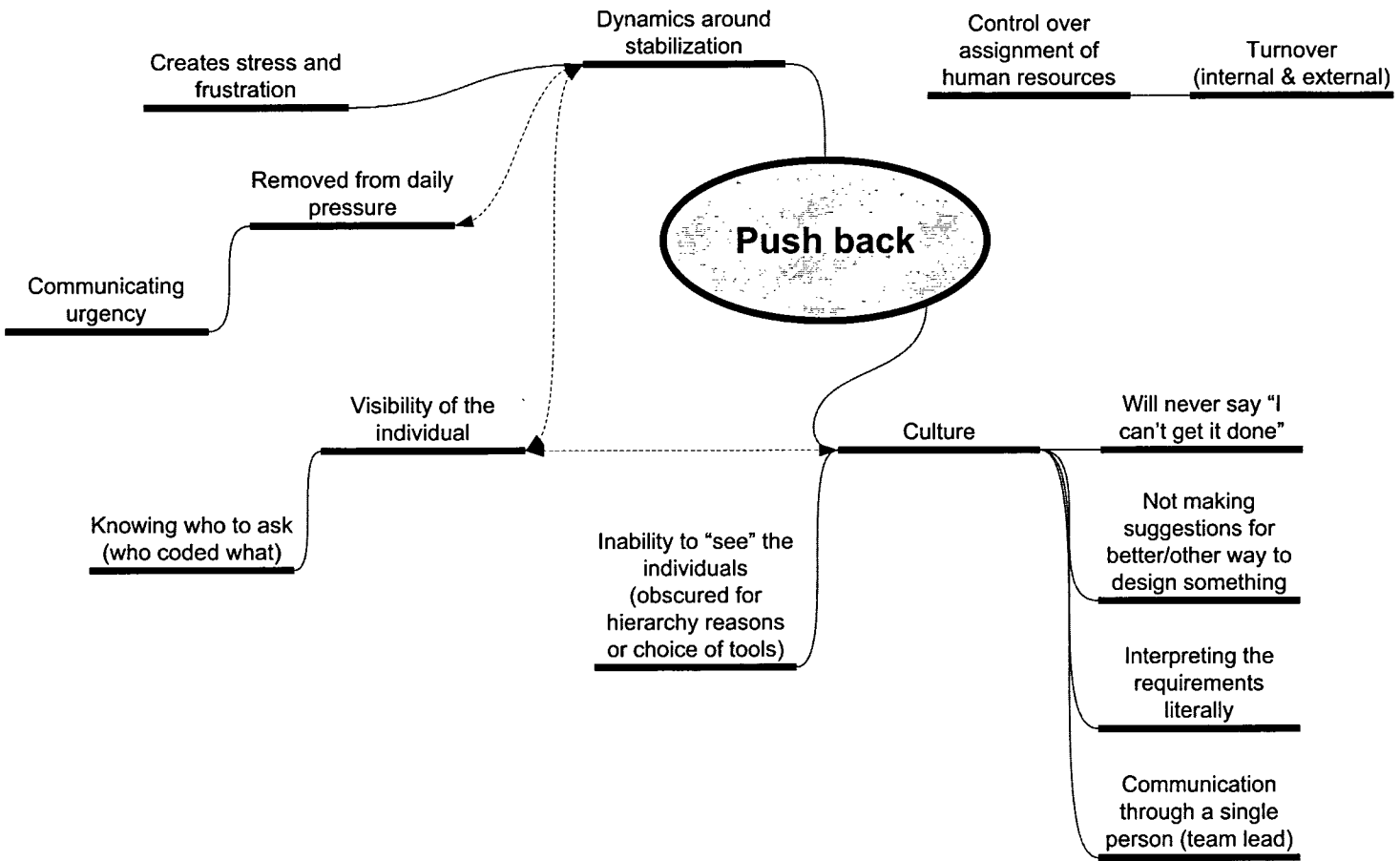
| Code | Instances |
|--------------------------------|-----------|
| Problem | 54 |
| Strategy | 46 |
| Culture | 44 |
| Team | 34 |
| Communication | 32 |
| Risk | 16 |
| Process | 14 |
| HR | 13 |
| Management | 11 |
| Quality | 11 |
| Agile | 11 |
| Language use | 9 |
| Hierarchy | 9 |
| Domain knowledge | 8 |
| Turnover | 8 |
| Tools | 7 |
| Outsourcing | 7 |
| Infrastructure | 7 |
| Magnitude | 7 |
| Time difference | 7 |
| Personal performance | 6 |
| Working globally | 6 |
| Quality attribution | 6 |
| Explicit ranking | 5 |
| Requirements review | 5 |
| Explanation for/about problem | 5 |
| Perception of situation | 4 |
| Organizational culture | 4 |
| Understanding the requirements | 4 |
| Communicate urgency | 4 |
| Expectation | 3 |
| Team structure | 3 |
| Homegrown leadership | 3 |
| Experience level | 3 |
| Peer mentoring | 3 |
| Work style | 3 |
| Push-back | 3 |
| Interpreting requirements | 3 |
| Comparison between teams | 3 |

| Code | Instances |
|---|-----------|
| Avoid comparisons | 3 |
| Responsiveness | 3 |
| Assessing skill level | 3 |
| Lessons learned | 3 |
| High turnover | 3 |
| Scheduling | 3 |
| Team boundary | 2 |
| Transplanting | 2 |
| Individualism collectivism | 2 |
| Visibility of individual | 2 |
| Face to face | 2 |
| Analyze – higher exposure | 2 |
| Contract | 2 |
| Communication filter | 2 |
| Time sensitive | 2 |
| Sense of time | 2 |
| Responding to email | 2 |
| Understanding accents | 2 |
| Opinions set in | 2 |
| Control assigning tasks remotely | 2 |
| Assign task locally | 2 |
| Quick turnaround | 2 |
| Building domain knowledge | 2 |
| Miss in the requirements | 2 |
| Asking for clarification | 2 |
| Planning – up front | 2 |
| Identify – a priori | 2 |
| Difficult to get the right answers | 2 |
| Territorial behaviour | 1 |
| System architecture along geographic boundaries | 1 |
| Revealing age | 1 |
| Salary privacy | 1 |
| No secret rewards | 1 |
| Managing relationships | 1 |
| Fired | 1 |
| Unglue grouping | 1 |
| Offshoring | 1 |

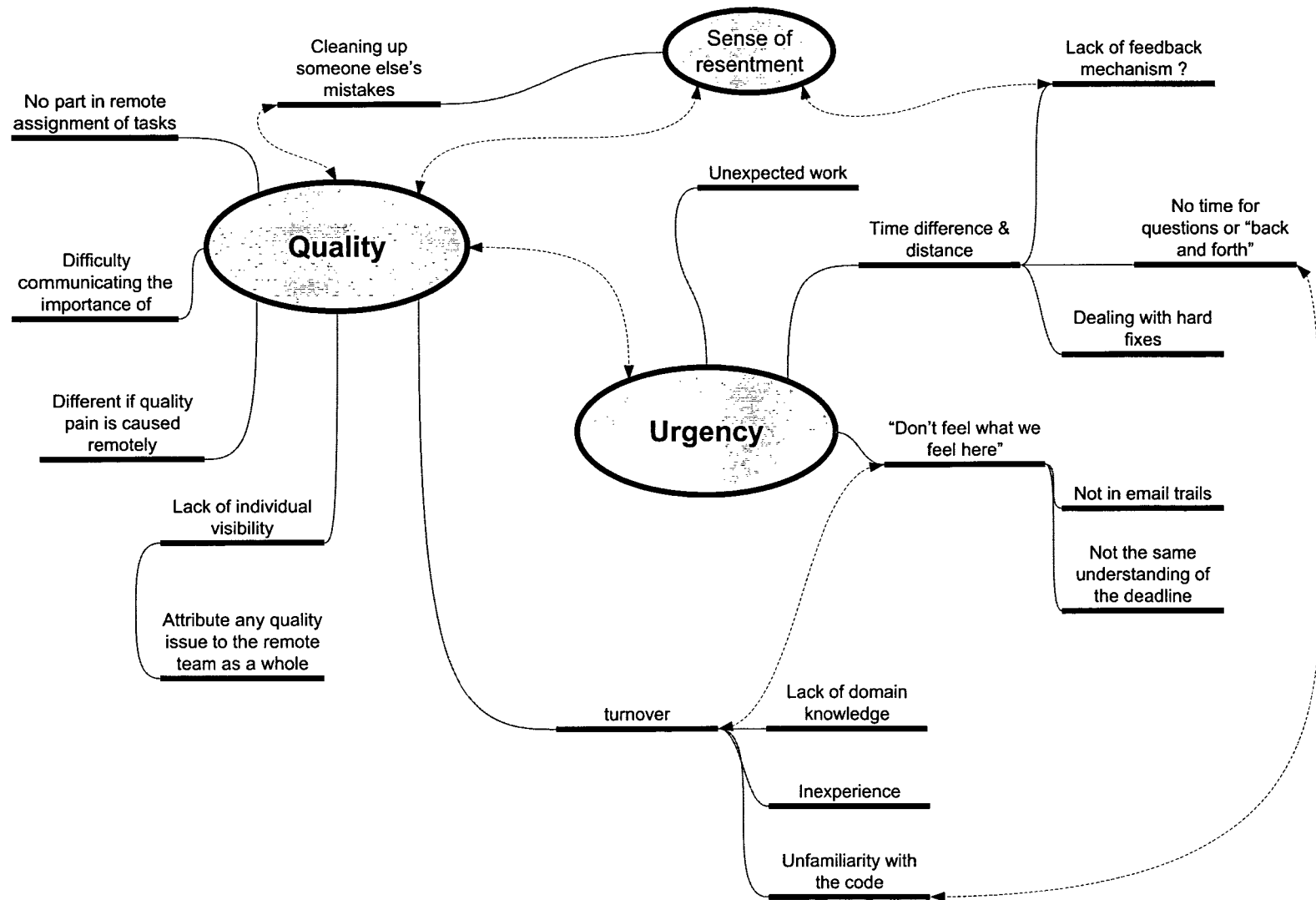
| Code | Instances |
|--|-----------|
| Cultural element | 1 |
| Star | 1 |
| Very modest | 1 |
| Extra regression testing | 1 |
| Uncertainty about outsourcing | 1 |
| Lunch benefits | 1 |
| Stretched too thin | 1 |
| Change | 1 |
| Feeling insecure | 1 |
| Tracking | 1 |
| Quality issue | 1 |
| Analyze – lesser exposure | 1 |
| Concern about local commitment | 1 |
| Negotiating tactics | 1 |
| Work flow | 1 |
| Version control system | 1 |
| Kickoff meeting | 1 |
| Leaving before the boss | 1 |
| Work hours flexibility | 1 |
| Understanding expectation | 1 |
| Requirements management | 1 |
| Perception of vendor-client relationship | 1 |
| Staffing flexibility | 1 |
| Communicate security | 1 |
| Not just cheap labour | 1 |
| Productivity difference | 1 |
| Use same tools | 1 |
| Metrics | 1 |

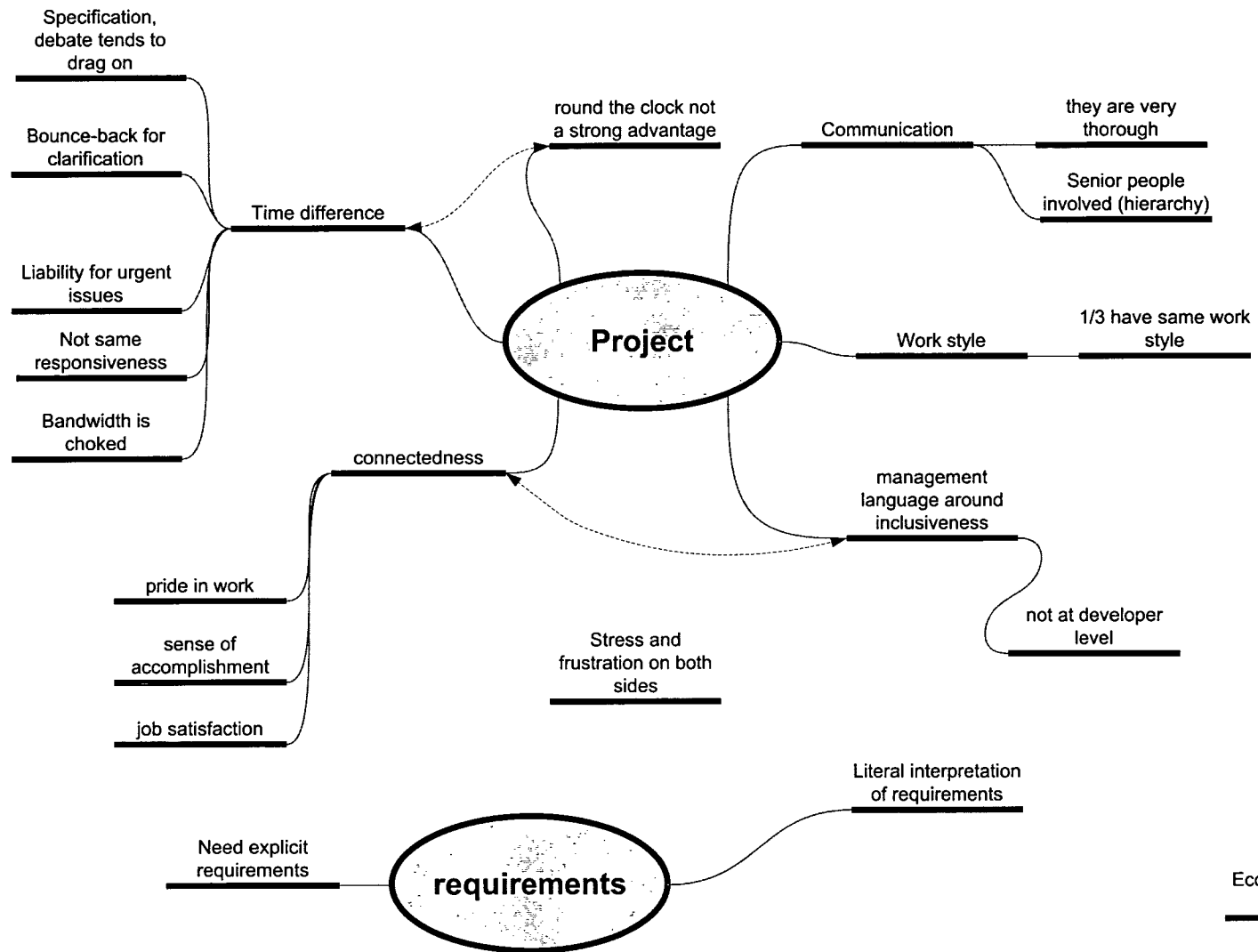
| Code | Instances |
|------------------------------------|------------|
| Reality of situation | 1 |
| Stress and frustration | 1 |
| High visibility on progress | 1 |
| Around the clock development | 1 |
| Isolation | 1 |
| Sense of accomplishment | 1 |
| Worked well | 1 |
| Dependency visibility | 1 |
| Feedback mechanism | 1 |
| Escalation | 1 |
| No issues with them | 1 |
| Unfamiliarity with the code | 1 |
| Meeting expectations | 1 |
| Ambitious | 1 |
| Stabilization | 1 |
| Lots of reported defects | 1 |
| Feature owner | 1 |
| Communication during stabilization | 1 |
| Lifecycle | 1 |
| Not pushing back | 1 |
| I can't get it done | 1 |
| Assigning bug fixes | 1 |
| Communicate daily pressures | 1 |
| Person | 1 |
| | |
| Total | 554 |

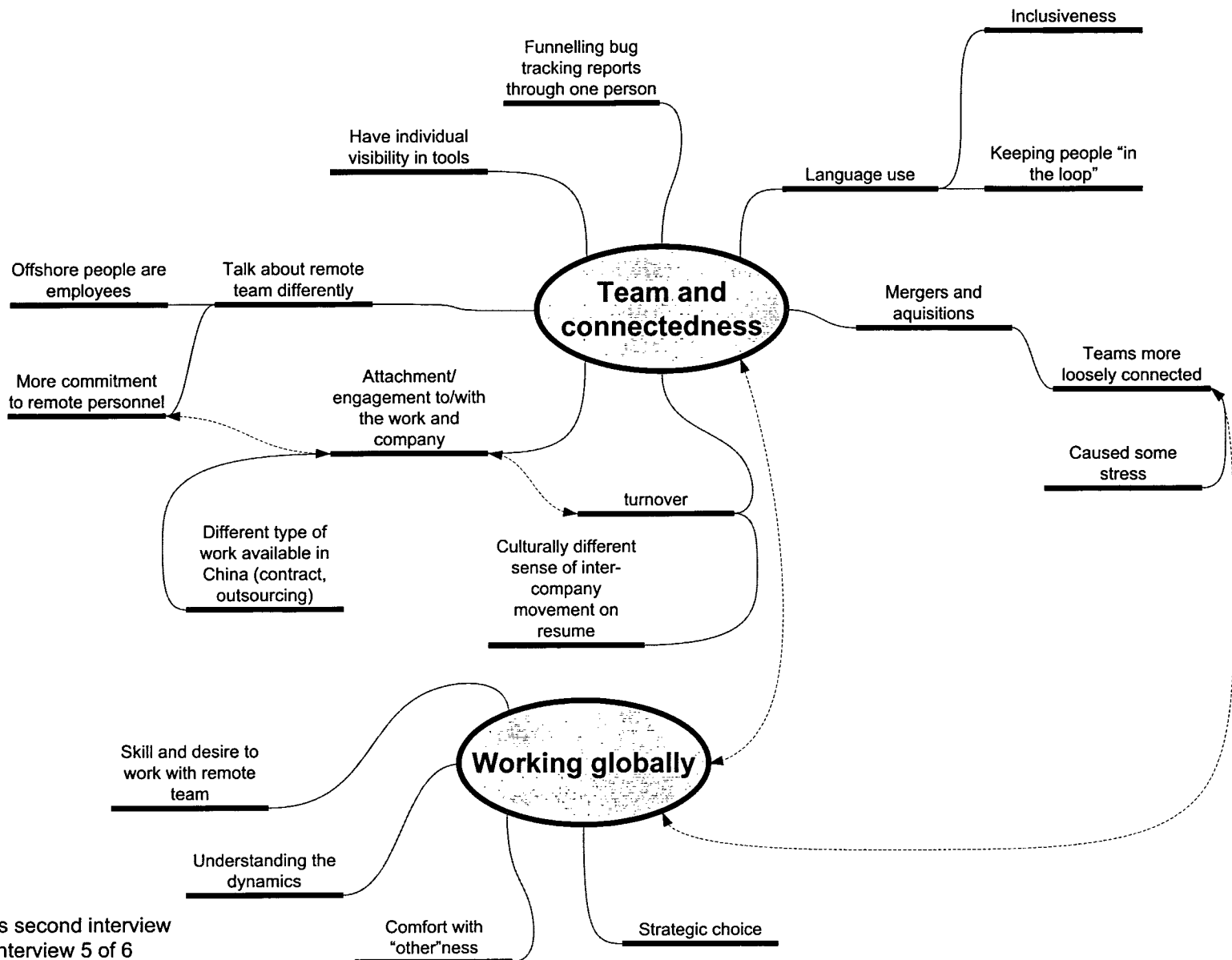
A.3 Conceptual Maps of Topics and Themes from Each Interview



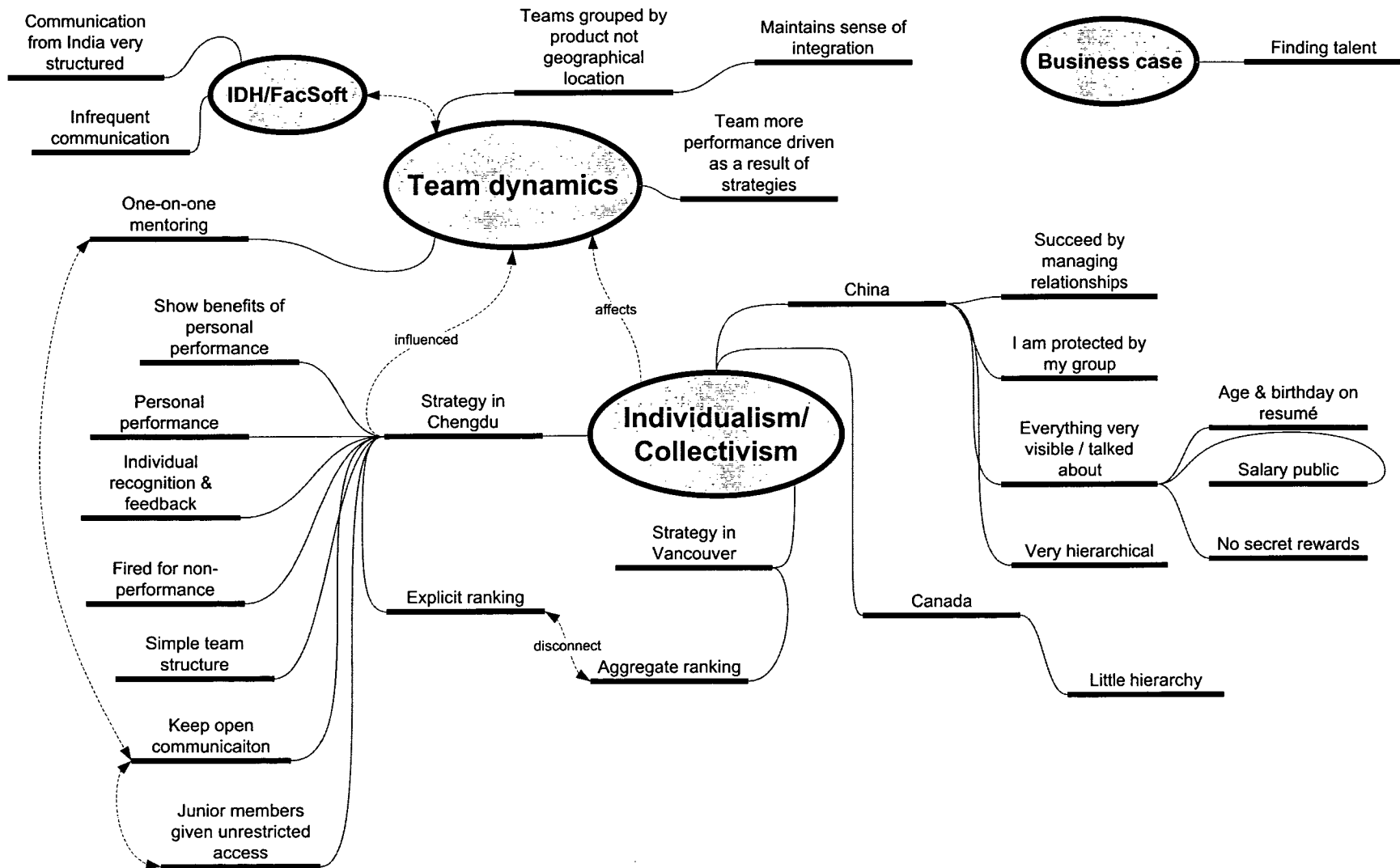
Ajay's first interview
Interview 2 of 6

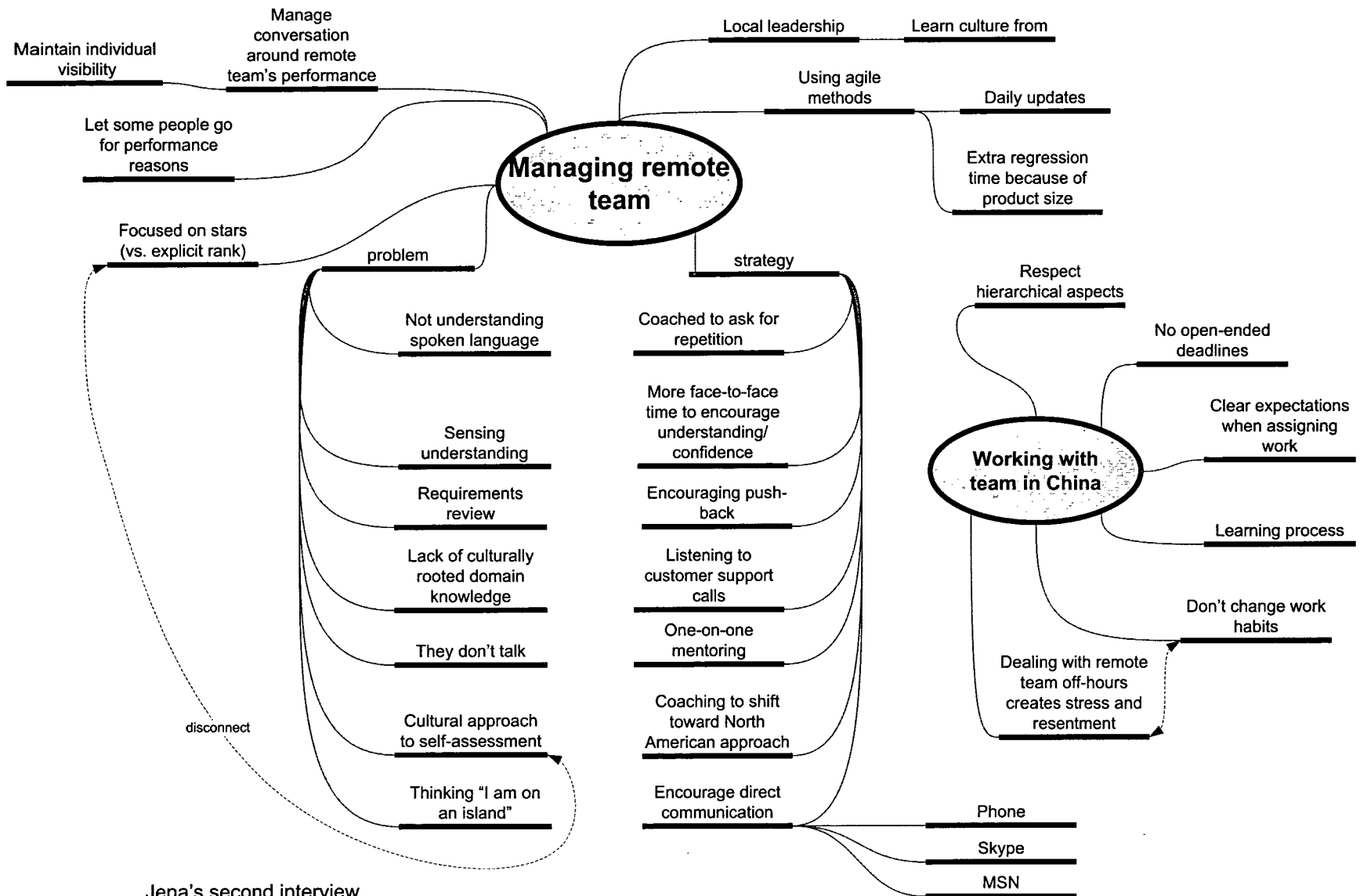






Alex's second interview
Interview 5 of 6





Jena's second interview
Interview 6 of 6