

Friedrich-Alexander-Universität Erlangen-Nürnberg
Technische Fakultät, Department Informatik

Bilal Zaghloul
MASTER THESIS

A Theory of Problems and Solutions in German/Chinese and American/Chinese Software Engineering Collaborations

Submitted on:

26.02.2014

Supervisors:

Prof. Dr. Dirk Riehle, M.B.A.

Professur für Open-Source-Software

Department Informatik, Technische Fakultät

Friedrich-Alexander University Erlangen-Nürnberg

Prof. Minghui Zhou, Ph.D.

School of Electronics Engineering and Computer Science

Peking University

zhmh@pku.edu.cn

Versicherung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Nürnberg, den 26.02.2014

License

This work is licensed under the Creative Commons Attribute 3.0 Unported license (CC-BY 3.0 Unported), see http://creativecommons.org/licenses/by/3.0/deed.en_US

Nürnberg, den 26.02.2014

Acknowledgements

I would like to express my deepest gratitude and appreciation to my supervisor, Prof. Dirk Riehle, for the useful remarks, comments, and engagement through the research process of my master thesis. I am also very grateful to him for introducing me to Prof. Minghui Zhou from Peking University, whose knowledge, constructive feedback, and guidance were invaluable. Prof. Zhou, thank you for accepting me as a member of your group, and for being such a great host. Prof. Riehle, and Prof. Zhou, thank you so much for offering me the opportunity to work with you on this thesis. A one simply could not wish for better advisors.

Moreover, I would like to thank all our interview partners, who have generously and willingly shared with us some of their experiences in the field. I truly appreciate the time they took from their busy schedules to participate in our interviews. Their contributions were priceless. Furthermore, a special thanks goes out to BayChina for funding my thesis work in China. I sincerely appreciate it.

In addition, I am truly thankful to all my colleagues in Peking University, in particular: Zhu Jiaxin, Shi Tingxun, and Hong Haiyan for making my six-month stay in Beijing very pleasant and memorable. They were there whenever I needed them, thus, I am extremely grateful to them.

Last but not least, no words can describe how thankful and grateful I am to my parents and family for all the love and support they gave me through my entire life. This work would not have been possible without their continuous support and encouragement. Their prayers, best wishes, and support were irreplaceable. Also, a very special thanks goes to all my friends for their encouragement and best wishes.

Abstract

Global software development projects are rapidly increasing due to the globalization of the software industry. However, 69% of cross-regional projects fail completely or partially, because of the lack of cross cultural understanding. This thesis presents a qualitative study of the impact of cultural differences on Global Software Development (GSD). We show a theory of problems and solutions of German/Chinese and American/Chinese collaborations in GSD. The theory presents two main categories: communication, and trust. We believe that the resulting theory may help companies not only understand the cultural problems, but also knowing how to overcome them based on solutions adopted by big multinational software companies.

Keywords: Global Software Development, Cultural Differences, Qualitative Research

Table of Content

Versicherung.....	I
Acknowledgements	II
Abstract	III
Table of Content	IV
List of Figures	V
List of Tables	VI
1 Introduction.....	1
1.1 Original Thesis Goals	1
1.2 Changes to Thesis Goals.....	1
2 Research Chapter	2
2.1 Introduction	2
2.2 Research Question & Approach	4
2.3 Cases Studied.....	5
2.4 Data Analysis Process.....	7
2.5 Resulting Theory	8
2.5.1 Communication	8
2.5.2 Trust.....	14
2.6 Limitation	16
2.7 Conclusion	17
3 Elaboration of Research Chapter	18
3.1 Emergence of Categories and Concepts	18
3.1.1 Emergence of Communication	18
3.1.2 Emergence of Trust	21
3.2 Profiles.....	23
References	VII
Appendix A – Informed Consent.....	X
Appendix B – Interview Questions	XI
Appendix C – Code Hierarchy	XVII
Appendix D – Illustration of Resulting Theory.....	XX

List of Figures

Figure 1: Research process.....	5
Figure 2: The emergence of the category “Communication”.....	8
Figure 3: Illustration of the emergence of “Communication”.....	21
Figure 4: Illustration of the emergence of “Trust”	23

List of Tables

Table 1: List of interview partners	5
Table 2: Profiles of all projects.....	7
Table 3: Code samples for the concept “Communication Behaviors”	8
Table 4: Percentage of participants for each concept in “Communication” category.....	9
Table 5: Problems related to “Reporting Failure”	10
Table 6: Solutions related to “Reporting Failure”	10
Table 7: Problems related to “Communication Behaviors”	11
Table 8: Solutions related to “Communication Behaviors”	12
Table 9: Problems related to “Collaboration across Regions”	13
Table 10: Solutions related to “Collaboration across Regions”	14
Table 11: Percentage of participants for each concept in “Trust” category	14
Table 12: Problems related to “Transparency”	14
Table 13: Solutions related to “Transparency”	15
Table 14: Problems related to “Delegation and Traveling”	16
Table 15: Solutions related to “Delegation and Traveling”	16
Table 16: Samples of codes for the concept “Communication Behaviors”	19
Table 17: Samples of codes for the concept “Reporting Failure”	20
Table 18: Samples of codes for the concept “Collaboration across Regions”	20
Table 19: Samples of codes for the concept “Transparency”	22
Table 20: Samples of codes for the concept “Delegation and Traveling”	23
Table 21: Full overview of the profiles of the projects	31

1 Introduction

1.1 Original Thesis Goals

The thesis develops a theory of problems and solutions in German/Chinese software engineering collaborations. The goal is to better understand the challenges of such cross-cultural, cross-time zone collaborations and how they affect productivity and/or software quality. We perform a series of interviews, analyze them, and develop a theory of the challenges. From the theory, several pertinent hypotheses as to problems and solutions and how they affect productivity and/or software quality are derived. We then design and execute a hypothesis-testing survey. We analyze the survey results to validate or invalidate these hypotheses.

1.2 Changes to Thesis Goals

We did a few changes to the original goals due to some circumstances occurred during the research work. First, depending only on the performed interviews, we were not able to clearly relate the impact of cultural differences on productivity and software quality. Thus, we decided to maintain our focus on the overall project. Second, we agreed to omit the validation part due to time restrictions. Third, we extended our focus to include American/Chinese collaborations for the following reasons:

- There were many American software companies that have development centers in China. Thus, this provided us with a new channel for interviews.
- We had some difficulties in scheduling interviews with German companies, mainly due to their busy schedules.

2 Research Chapter

2.1 Introduction

Interest in global software development (GSD) is growing as the software industry is experiencing an increase of globalization of business (J.D. Herbsleb & Moitra, 2001). The reasons motivating GSD are reduction of costs, access to a pool of skilled labor, necessity of getting closer to customers, time difference utilization, and improving the quality of work (Carmel, 1999). Yet, according to studies from major Big 4¹ auditor, 69% of all outsourcing projects fail completely or partially. The main reason is the lack of cultural compatibility between the vendor and the client and a poor relationship management (OSF Global Services, 2012). Casey (2009) reports that cultural differences can negatively impact software teams. A survey conducted by Accenture reveals that miscommunication and a lack of cross-cultural understanding can hinder the effectiveness of global sourcing (Accenture, 2006).

The impact of cultural differences on GSD is not a new topic; there is already a body of literature that acknowledges it. To elaborate, we divide existing literature into quantitative, and qualitative studies. As examples for quantitative studies, Bird, Nagappan, Devanbu, Gall, and Murphy (2009) study the development of Windows Vista, and compare the failures of the components developed in a distributed fashion with the failures of the components developed locally. Ramasubbu and Balan (2007) present a two-year study that investigates the effects of dispersion on the productivity and quality of distributed software development. Damian and Zowghi (2003) report a field study that investigates requirements engineering challenges introduced by the geographical dispersion in a multi-site organization. In their report, they present a model of how remote communication and knowledge management, cultural diversity and time differences negatively impact on requirements gathering, negotiation and specification. Herbsleb and Mockus (2003) present an empirical study of communication and speed in GSD. They found that distributed work items take about two and one-half times as long to complete as similar items where all the work is collocated. Furthermore, Herbsleb, Mockus, Finholt, and Grinter (2001) report an empirical study that explores the delay in a multi-site software development organization. Their results show that compared to same-site work, cross-site work takes much longer and requires more people for work of equal size and complexity.

Moreover, as to qualitative studies, Abraham (2009) presents an experience report of the issues that arise from cultural differences in interacting among team members. In his report, he contrasts working style, communication style, and behavioral style between Indian and non-Indian team members. Holmstrom, Conchúir, Ågerfalk, and Fitzgerald (2006) show the impact of temporal, geographical, and socio-cultural distances on GSD. They also present some solutions based on qualitative interviews of American and Irish companies. From their findings, language barriers have the biggest impact in the socio-cultural distance. Moreover, Dorairaj, Noble, and Malik (2011) present strategies adopted by Agile practitioners to overcome the cultural differences between the US and India. Shah and Harrold (2013) study the effect of cultural differences between the US and Japan on software testing approaches. Furthermore, Brockmann and Thaumuller (2009) explore the cultural challenges of requirements engineering in GSD during a German-Chinese joint software project. Herbsleb, Paulish, and Bass (2005) present an experience report capturing the results of a multiple-case study of nine

¹ The Big Four are the four largest international professional services networks:
[http://en.wikipedia.org/wiki/Big_Four_\(audit_firms\)](http://en.wikipedia.org/wiki/Big_Four_(audit_firms))

software development projects in a large, geographically-distributed corporation. In their report, they state experience and lessons in issues of project management, division of labor, on-going coordination of technical work, and communication.

In addition, there are also many articles in this area, for instance, Herbsleb and Moitra (2001) explain how cultures differ on many critical dimensions, particularly communication styles. They show that cultural differences can lead to serious misunderstandings among team members who do not know each other well. Dhir, Sahay, and Walsham (2004) explore best practices for effective management of global software teams. They discuss on-the-job cross-cultural training to help teams achieve a better understanding of cultural differences. Carmel and Agarwal (2001) provide several emerging approaches that can be applied across a range of geographically distributed projects. Ebert and De Neve (2001) present also lessons learned from GSD.

However, most of the quantitative studies explore certain parts of the development process in GSD, rather than concentrating on the whole process, e.g., the impact of distance on communication. Moreover, they do not investigate problems and solutions that are limited to certain contexts, e.g., German/Chinese collaborations. Thus, we believe that our results are relatively different from results of quantitative studies. As for qualitative studies, most of the studies show only the problems without solutions. Yet, only a limited number present problems and solutions in specific contexts, e.g., between the US and India. Nonetheless, we also believe that our findings are unique and not similar to such studies, mainly due to different contexts, difference in data collection, and different way of analysis.

Furthermore, most of the qualitative studies in this area apply predefined frameworks such as, Hofstede² five dimensions model, Hall³ cultural factors, and Trompenaars and Hampden-Turner⁴ model. Many researchers have opposed such approaches, because predefined frameworks make it difficult to deeply understand the effects of culture on GSD (Boden, Avram, Bannon, & Wulf, 2012; Cater-Steel & Toleman, 2008; Shore, 1996). Moreover, culture in these frameworks is limited only to a number of predefined dimensions (Hutchins, 1995; McSweeney, 2002; Shore, 1996). Given the importance of the culture's role in GSD, it is important to conduct appropriate research studies to gain more knowledge about it (Boden et al., 2012; Cater-Steel & Toleman, 2008; Irani & Dourish, 2009).

In this chapter, we present the results of a qualitative study on how cultural differences affect GSD. We provide a theory of problems and solutions of German/Chinese and American/Chinese collaborations in GSD. The theory can help software companies not only understand the problems associated with the cultural differences, but also knowing how to overcome them based on best practices from large multinational software companies.

The main contributions of this research are:

- A theory of problems and best practices of German/Chinese and American/Chinese collaborations in GSD.
- The addition of profiles of the studied cases (companies). Each case is associated with the corresponding problems caused by the cultural differences, as well as the implemented solutions.

Our data were gathered through six interviews with five big multinational software companies that have software development centers in China. The interviews were either face-

² Hofstede, Geert. *Culture's Consequences, Comparing Values, Behaviors, Institutions, and Organizations Across Nations* Thousand Oaks CA: Sage Publications, 2001 - for every detail on Hofstede's research.

³ Hall, E. 1997 *Beyond culture*, Anchor Book, 2nd edition (originally published in 1977).

⁴ Hampden-Turner, C., & Trompenaars, F. 1997 Response to Geert Hofstede, *International Journal of Intercultural Relations*, 21-1, p. 149- 159.

to-face or over the phone. We interviewed mainly project managers that involve directly in the collaboration process across the regions. We applied the Grounded Theory (GT) approach to analyze the data. GT allowed us to gain insight into the major encountered problems, in addition to the developed solutions. Our findings present two main categories that affect the collaboration process: communication, and trust.

The chapter is structured as follows: Section 2.2 covers the research question and research approach, and explains the reasons for choosing GT. Section 2.3 lays out the cases studied, including case selection, method selection, and cases profiles. Section 2.4 then illustrates the data analysis, including an example from the actual data. Section 2.5 presents the resulting theory. Section 2.6 contains the limitation of our results. And section 2.7 draws on the conclusion.

2.2 Research Question & Approach

We adopted the GT approach in our research study. GT is a general method of analysis that utilizes all kinds of data it can get, for example from interviews, for qualitative data analysis. It is an inductive research method developed by Barney Glaser and Anslem Strauss. It is “the systematic generation of theory from systematic research”. The goal is to directly and systematically derive a theory from the data. (Glaser & Strauss, 2009)

We believe that GT is a suitable approach for our research for the following:

- GT is suitable in areas that need further exploration or a new perspective that might be beneficial. (Schreiber & Stern, 2001)
- A limited number of qualitative studies on the effects of cultural differences in GSD that do not apply predefined frameworks (e.g. Hofstede, and Hall). Thus, by using GT, we can help increase the literature in this area.
- GT is capable to answer our research question: What are the problems and solutions caused by the cultural differences of German/Chinese and American/Chinese collaborations in GSD?

In GT research, a researcher starts by defining a broad area of interest, rather than a specific research question (Glaser, 1992). This lets him/her not preconceive ideas about the problem, rather discover them from the early stages of the data analysis (Glaser, 1992, 1998). For this reason, we started our research by addressing a broad area of interest, which is culture and GSD. Then, we started to collect the data. We chose interviews as method of data collection. Our sample included six individuals in five German and American multinational software companies. We targeted individuals with direct engagement in the development process between either Germany and China or the US and China. Moreover, they all have significant international bi-cultural exposure and work experience in international projects. Table 1 shows the list of our interview partners.

Participant's Code	Role	Company's HQ	Cross-Cultural Experience
P1	Senior Consultant	Germany	5 years
P2	Director R&D	Germany	9 years
P3	Development Manager	US	10 years
P4	Senior Developer	US	4 years
P5	Development Manager	US	10 years

P6	Manager R&D	Germany	10 years
----	-------------	---------	----------

Table 1: List of interview partners

Our research process started by reviewing the existing literature including papers, articles, books, etc. This allowed us form an initial research question that served as a starting point for interviews preparation, “What is the impact of cultural differences of German/Chinese collaborations in GSD on productivity and software quality?” The preparation for each interview was by preparing open-ended questions in various areas to be addressed during the interview. These questions were used solely to keep track of the interview, and were not given to the interviewee beforehand. Furthermore, we refined questions after each interview depending mainly on the analysis results of the previous interview, as well as the existing literature. The analysis provided us with focused areas that we could address further in the next interview. After six interviews, we reached data saturation, i.e. we did not receive new problems, rather repetitive ones. At this point, we decided to move forward towards forming the theory. Figure 1 briefly illustrates our research process.

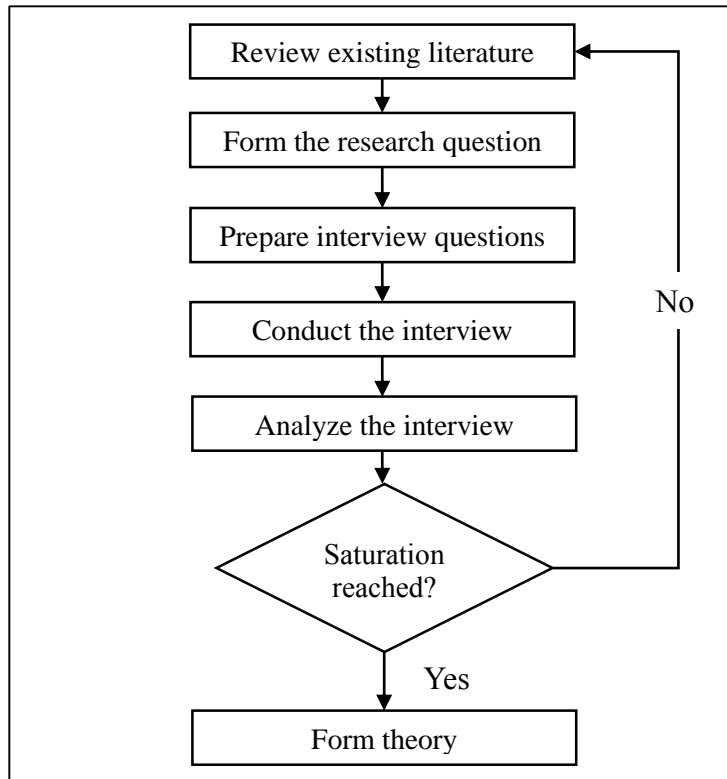


Figure 1: Research process

2.3 Cases Studied

We interviewed six individuals in five multinational software companies. The interviews were divided equally between German and American companies. Moreover, we ensured to have interview partners from different sectors in the software industry, e.g. electronics, and enterprise solutions. This variation allowed us gain more knowledge about the effects of culture in software development from different angles, and also to see if different sectors have different problems or not, more or less. We targeted individuals with direct engagement in the software development process between either Germany and China or the US and China. We had two personal face-to-face interviews that took place in China, and four interviews over the phone.

At first, we contacted our interviewees via email. Before each interview, we ensured to provide them with sufficient information about our research, interview structure, as well as other details (see Appendix A). Each interview lasted between 1 – 1.25 hours. We got permission to audio record all interviews except for one, where the interviewee did not give us permission. Moreover, taking notes was present at all interviews, and it was very intense at the non-recorded interview. As for interviews’ preparation, we prepared a set of guiding questions before each interview to stay focused on some main topics. We were careful in initiating questions, thus, all questions were open-ended in order not to get biased answers. For example, “Could you tell me about an incident caused by German/Chinese collaborations?” or “How do you manage the collaboration across regions?” Besides this, refining questions was ongoing after each interview (see Appendix B). We refined questions to keep our attention focused on the main concerns. Equally important was the analysis process of interviews using the GT approach. The analysis let us explore and assess the main concerns of each interview, which was directly reflected on the refinement of questions. Section 2.4 illustrates the analysis process in detail.

As for our projects, table 2 shows their profiles. Each profile contains relevant information about each project, e.g. sector, software methodology, and projects’ distribution. Due to privacy and ethical consideration, we refer to the participants as “P”, and to the companies as “X”. We have six participants (P1 to P6), and five companies (X1 to X5).

Company Participant	Sector	Software Methodology	Projects’ Distribution	Project’s Duration	Team Size	Development Centers
X1 P1	Consulting	Scrum	Germany – China – India – Slovakia	24 months	Germany (12); China (3); India (10); Slovakia (8)	Germany (1); China (1); India (1); Slovakia (1);
X1 P2	Energy & Automation	Plan-driven	Germany – China – India	12 – 15 months	Germany (350); China (140); India (250)	Germany (2); China (2); India (2)
X2 P3	Platforms	Agile & Plan-driven	US – China – India	12 – 18 months	China (10); US (5); India (10)	China (2); US (2); India (1)
X3 P4	Platforms	Scrum	US – China – Canada	6 – 12 months	China (20); US (50 – 60); Canada (10)	China (1); US (2); Canada (1)
X4 P5	Enterprise Solutions	Plan-driven	US – China – India	6 – 12 months	China (160); US (300); India (500)	China (1); US (1); India (1)
X5 P6	Electronics	Plan-driven	Germany – China –	9 – 36 months	Germany (3 – 10);	Germany (1 – 3);

France – Swe- den	China (3 – 10); France (3 – 10); Sweden (3 –10)	China (1 – 2); France (1); Swe- den (1 – 2)
----------------------	---	---

Table 2: Profiles of all projects

2.4 Data Analysis Process

In GT, data collection and data analysis should be done simultaneously. Therefore, we ensured to do the analysis after each interview. We used MAXQDA⁵ software to help in the analysis. The software provides a set of features that helps the researcher organize and process all of his data in a smooth way. After transcribing the interview, we scanned the interview line-by-line to search for similarities and differences (Georgieva & Allan, 2008). Then, we located key points in the text, and assigned a *code* to each key point. This phase in the process is called *coding* (Allan, 2003b). Coding should be performed with an open-mind, i.e. the researcher should not force looking for evidence to support establishing some preconceived ideas (Glaser & Strauss, 2009). As we identified codes, we compared constantly each code with other codes of the same interview as well as codes of other interviews. The codes related to a common theme were grouped together to form a second level of abstraction called a *concept*. This approach is called the constant comparative method (Glaser, 1965). As we identified all concepts, we continued to perform constant comparison to form a third level of abstraction called a *category*. The explanation of concepts and categories is the Grounded Theory: a theory that is truly grounded in the data.

In order to explain the process, we present the analysis of data that leads to the emergence of the category *Communication*. The analysis starts with the *key point coding* technique (Allan, 2003a). In this approach, the researcher uses an identifier (e.g. “K”) to identify key points, and a suffix to identify the participant (e.g. “P”). For instance, “K_{P1}4” refers to key point 4 made by the first participant. Thus, we can trace back to the actual quote and context of each key point.

However, because we used MAXQDA as a tool for the analysis, using identifiers was not necessary, as the software provided a smoother way for organizing and tracing the codes. Therefore, we used only the participant “P” to identify codes. Table 3 shows the emergence of the concept *Communication Behaviors*.

The code *Respecting hierarchy* was compared with all other codes in all interviews to identify if similar codes occurred frequently. The codes shown in table 3 have a common theme: *Communication Behaviors*. The common theme is a concept that emerged from the codes. Moreover, while performing coding on other key points, other concepts emerged: *Reporting Failure*, and *Collaboration across Regions*. By applying the constant comparative method to these concepts, we were able to identify the category *Communication* that emerged from the concepts. Figure 2 shows the emergence of the category *Communication*.

ID	Key Point	Code
P2	As a lead there [in China], I would just talk to a developer or a tester without bringing the team lead with me. But this is not normal in China.	Respecting hierarchy

⁵ MAXQDA – Qualitative Data Analysis Software: <http://www.maxqda.com/>

P5	US developers are quite open. Whenever they have problems they will ask, whether it is valid or not. Sometimes, they even will raise questions to ask for clarifications.	Open discussion
P3	Sometimes in the QA session, we will ask a developer [a Chinese developer]: do you have any questions, do you fully understand the requirements? And the answer most likely would be yes, okay.	Affirmation
P5	In China, if the developers don't have confidence on the idea, the solution, or the approach, they will not talk about it in the meeting.	Difficulty with open discussion
P1	If a German product owner or whatever his role is tells a story, tells the requirements, and then asks for questions there [in China], there won't be any questions.	<ul style="list-style-type: none"> • Open discussion • Respecting hierarchy

Table 3: Code samples for the concept “Communication Behaviors”

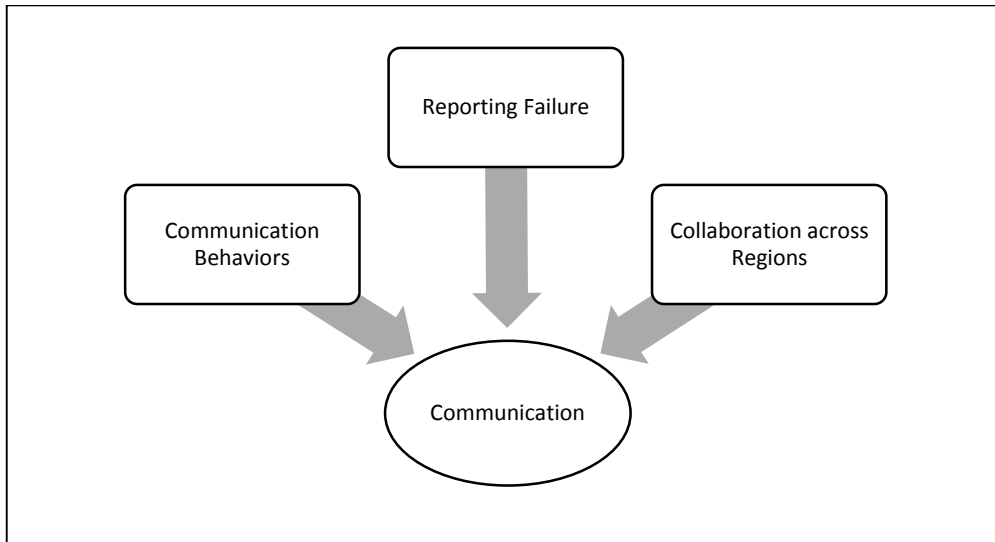


Figure 2: The emergence of the category “Communication”

2.5 Resulting Theory

As we performed the analysis on our data, two main categories emerged: *Communication*, and *Trust*. In this section, we explore the emergence of these categories by providing explanations from the literature, as well as some actual quotations of our participants. The emergent theory presents problems, and solutions of German/Chinese and American/Chinese collaborations.

2.5.1 Communication

This category emerged from three concepts: *Reporting Failure*, *Communication Behaviors*, and *Collaboration across Regions*. Data analysis showed that 153 out of 199 quotations are related to *Communication*, i.e. 77% of all quotations relate to *Communication*. Table 4 shows the percentage of participants who made explicit comments about each concept, as well as the percentage of quotations for each concept.

Concept	Participants		Quotations	
Reporting Failure	3 (P1,P3,P4)	50%	34	17%
Communication Behaviors	4 (P1,P2,P3,P5)	67%	37	19%
Collaboration across Regions	5 (P1,P3,P4,P5,P6)	83%	82	41%

Table 4: Percentage of participants for each concept in “Communication” category

Reporting Failure

Reporting mistakes during projects was a noticeable difference in the collaboration between either Germans and Chinese or Americans and Chinese. Half of our participants (3, or 50%) mentioned explicitly some situations where they encountered this problem:

“... for any culture you know, as long as you have a strong reason and you actually gave your best, the manager or the boss can actually accept it, but in China, the culture is if you cannot deliver, whatever or how hard you try, if you still cannot deliver, that’s bad.”—P3, Development Manager

“...they [Chinese developers] are also hesitant to tell you something is wrong in here. And you, as part of the project, have to know this, and of course if you don’t know about this problem, you will just accept whatever they tell you and you would not like to ask.”—P1, Senior Consultant

We relate the samples above to one main reason: the face culture. Existing literature explains this culture clearly. Ho (1976) says that the face is lost when the individual, either through his action or that of people closely related to him, fails to meet essential requirements placed upon him by virtue of the social position he occupies.

Furthermore, the analysis revealed some of the effects of this problem on projects, e.g. delays, and additional costs. One example explains:

“The developers in China will keep trying and trying, after work hours and put much effort, and they still want to meet the deadline until they cannot. And then always you’ll get a notification only minutes before the deadline.”—P3, Development Manager

Fear of losing face could be a possible explanation, as it makes it difficult for Chinese developers to report a delay, because it may be understood as failing the boss. Table 5 illustrates the problems, and the number of participant(s) that explicitly mentioned them. Each problem is identified with an ID “PR” that we use later to map problems to solutions.

ID	Problem	Participants
PR1	Chinese developers are more reluctant than Germans and Americans to report mistakes during projects.	2 (P1,P3)
PR2	Chinese make late notifications if not being able to meet the deadline. Yet, they are willing to spend long after work hours to finish the task.	1 (P3)

PR3	Strictness of Chinese team leaders makes it difficult for some developers to report mistakes or delays.	1 (P4)
-----	---	--------

Table 5: Problems related to “Reporting Failure”

Furthermore, the data analysis also uncovered solutions adopted by our interview partners to overcome some of these problems. Table 6 presents the solutions. Each solution is associated with the problem ID(s) it solves, as well as the number of participant(s) that explicitly mentioned it. We use “SL” to identify solutions. Some solutions have the symbol “-” or none in the *problem’s ID* column, because such solutions were used to either improve a process, or avoid expected problems.

Solution’s ID	Solution	Problem’s ID	Participants
SL1	Ask management to appreciate reporting mistakes during projects.	PR1	1 (P1)
SL2	Create workshops to articulate ideas with anonymous identities, because anonymity helps Chinese avoid sharp direct feedback.	PR1	1 (P1)
SL3	Apply a progress-tracking system, where a developer updates the status of his task daily, to avoid late notifications.	PR2	1 (P3)
SL4	Choose software development method that encourages communication, e.g. Scrum, or other agile methods.	PR1	1 (P1)
SL5	Assign a local Chinese expert to get back to in case of problems.	-	1 (P4)
SL6	Bring up the message that delays are acceptable in presence of a strong justification.	PR2	1 (P3)

Table 6: Solutions related to “Reporting Failure”

As for “SL2”, each participant writes his/her idea on a card, then pins it on a board with no identity. Afterwards, managers will check all cards and write feedback beside each one. In this way, the burden of sharp feedback will be avoided, thus, developers will be encouraged to participate and to be more interactive. Moreover, regarding “SL3”, if a task’s status is “20%” two days before the deadline, it indicates that a delay is expected to happen. Therefore, the manager can avert a late notification, and modify the timeline accordingly.

Communication Behaviors

The majority of our interview partners (4, or 67%) observed several differences in the communication styles of Chinese developers in comparison with their American or German

peers. For instance, direct German and American style vs. indirect Chinese style, formal Chinese style vs. informal style, and so forth. Some of our interview partners comment:

“Another thing is there about the communication style. You know most of the US colleagues have a fairly direct style of communication. But in China it is different. People may not explain the idea directly. That may lead to people making mistakes. So for the developers, they need to do more tests before they release the idea ... that is the direct and indirect communication style.”—P5, Development Manager

“I noticed already and from experience that it's hard to get the feedback in group [in China], so you give a speech on some topics and you ask questions or you ask for some feedback, no one will raise his hand. If they [Chinese developers] had questions, after your speech, go to your desk, then people [Chinese developers] start to grab you by the hand. They say: we have a better way, can we go to the meeting room, just two of us. Then they start to talk very openly. So one to one, they are quite open because I am not as superior, so they can tell me what they think. But in front of the group they won't do it somehow.”—P1, Senior Consultant

Furthermore, respecting hierarchy is a notable difference between China and other western countries. It corresponds to the *Power Distance* dimension in Hofstede's model (Hofstede, 2001). Hofstede says that the power distance is high in stratified societies where all powers are in the hands of the superior. In such societies, the subordinate feels that it is dangerous to question a decision of the superior. Thus, he will learn to behave submissively, at least in presence of the superior. China has a high power distance, while countries like Germany, or the US have relatively low power distance. This is a possible reason why Chinese developers do not tend to ask questions in group meeting in presence of the boss. Table 7 presents a list of related problems.

ID	Problem	Participants
PR4	Chinese developers have less tendency than Germans and Americans towards asking questions in group meetings.	3 (P1,P5,P3)
PR5	Chinese developers seldom argue or discuss their tasks with their superiors.	2 (P1,P2)
PR6	Chinese' “yes” or “no” has a different meaning for Americans.	1 (P3)
PR7	Chinese have a formal communication style, while Americans have an informal style.	1 (P5)

Table 7: Problems related to “Communication Behaviors”

Moreover, the analysis also revealed solutions to overcome some of these problems. Table 8 illustrates a list of related solutions that were made explicitly by our interview partners.

Solution's ID	Solution	Problem's ID	Participants
SL7	Ask the management to bring up repeatedly the message of the importance of open conversation.	PR4,PR5	2 (P2,P5)
SL8	Do not take “yes” or “no” for an answer. Chinese developers should write a document of their opinion after important sessions.	PR6	1 (P3)
SL9	Create a relaxing work environment for Chinese developers, where you can speak freely and informally with colleagues, and formally only with clients.	PR7	1 (P5)

Table 8: Solutions related to “Communication Behaviors”

Furthermore, solutions “SL2” and “SL4” that are listed in the *Reporting Failure* section were also mentioned to overcome problems “PR4” and PR5”.

Collaboration across Regions

This concept presents problems that may occur in managing the collaborations across regions, as well as some suggested solutions. The majority of our interview partners (5, or 83%) mentioned repeatedly several differences in capturing requirements, defining tasks, language barriers, and so on. Some of our interview partners state:

“This is a very big difference, the developers from the US will very eagerly try to join the design meeting and raise all kinds of questions on the implementation and the design. While for most of the China developers, they seldom join this meeting. This is a problem of culture, they [Chinese developers] ask about implementation details, about what they are working on, but the US developers don't like this.”—P4, Senior Developer

“The developers in China sometimes intend to work out or find out the requirement by themselves.”—P3, Development Manager

Table 9 presents a list of related problems that were explicitly mentioned by our interview partners.

ID	Problem	Participants
PR8	Chinese developers need detailed requirements about their tasks.	3 (P1,P3,P4)
PR9	Chinese cannot easily communicate in English due to language barriers.	3 (P3,P4,P6)
PR10	Chinese are detail-oriented, while Americans and Germans see the big picture.	3 (P1,P3,P4)

PR11	The US and China have different holidays. E.g. spring festival in China, and Christmas in the US.	1 (P4)
PR12	Chinese developers omit the context when discussing tasks.	1 (P5)
PR13	Chinese developers like to be challenged.	2 (P3,P5)
PR14	The terminology is documented only in the country's language, i.e. Chinese in China and German in Germany.	1 (P6)

Table 9: Problems related to “Collaboration across Regions”

As for “PR10”, our interview partners noticed that Chinese developers are only interested in their assigned tasks, while Germans and Americans have interests in the whole project. Moreover, regarding “PR8”, our interview partners mentioned that Chinese developers fill out any missing in the requirements by their own without getting back to the boss. And, as to “PR12”, our interview partners said that Chinese developers neglect providing context about their tasks when discussing them with their American peers, while Americans give full context about their tasks before discussing details.

Yet, our interview partners pointed out several solutions to overcome these problems. Table 10 shows a list of related solutions.

Solution's ID	Solution	Problem's ID	Participants
SL10	Create smaller and deeper tasks deliberately and associate them with specs and context.	PR8	1 (P1)
SL11	Split up the development cycle across regions, where you can exploit the benefits of each one, e.g. Americans deal with customers, Chinese design, and Indians implement.	-	1 (P3)
SL12	Pay more attention when defining requirements in order not to leave any space for guessing.	PR8	2 (P1,P3)
SL13	Use a wiki-like system for technical discussions. Yet, it is not useful in case of urgent issues.	PR9	1 (P4)
SL14	Remind Chinese developers repeatedly of the importance of providing context when discussing tasks.	PR12	1 (P5)
SL15	Create small discussion groups, where Chinese developers can discuss technical problems with their American colleagues.	PR13	1 (P5)
SL16	Provide Chinese developers with challenging tasks regularly.	PR13	1 (P5)

SL17	Create English training sessions in both sides. The training focuses on business terms used in the industry.	PR9	1 (P6)
SL18	Unify the terminology by creating a map table in the three languages: English, Chinese, and German, where each term is associated with its corresponding in English.	PR14	1 (P6)

Table 10: Solutions related to “Collaboration across Regions”

2.5.2 Trust

This category emerged from two concepts: *Transparency*, and *Delegation and Traveling*. Data analysis showed that 46 out of 199 quotations are related to *Trust*, i.e. 23% of all quotations relate to *Trust*. Table 11 shows the percentage of participants who made explicit comments about each concept, in addition to the percentage of quotations for each concept.

Concept	Participants		Quotations	
Transparency	4 (P1,P2,P3,P6)	67%	16	8%
Delegation and Traveling	4 (P1,P2,P4,P5)	67%	37	15%

Table 11: Percentage of participants for each concept in “Trust” category

Transparency

The majority of our interview partners (4, or 67%) mentioned the importance of the transparency across the collaborated regions. One example says:

“If it is the white box, I already know what teams are responsible for and what’s the progress they have made, another team will feel fine, if I am with you or them that’s the best I can do, so they have that kind of trust”—P3, Development Manager

Our data analysis revealed some related problems that were explicitly mentioned by our interview partners. Table 12 lists these problems.

ID	Problem	Participants
PR15	Chinese have a higher staff turnover rate than Germans.	2 (P1,P2)
PR16	Transferring knowledge to China is a serious problem, because of the low IPR value.	3 (P1,P2,P6)

Table 12: Problems related to “Transparency”

Regarding “PR16”, some interview partners pointed out particularly the intellectual property right (IPR) concern as a very serious issue especially in a county like China with a low IPR. Managers must think thoroughly before meeting any decision related to knowledge transfer (e.g. source code), because any mistake in this matter may endanger the whole company.

Furthermore, the analysis uncovered some suggested solutions to help overcome these problems. Table 13 shows a list of these solutions.

Solution's ID	Solution	Problem's ID	Participants
SL19	Launch a product line for the Chinese market to build loyalty and trust. This dramatically reduced the high turnover rate for Chinese.	PR15	1 (P2)
SL20	Make the work visible. Thus, all collaborating regions can see what other regions do.	-	1 (P3)
SL21	Establish a transparent relationship with Chinese. For example, explain to Chinese the reasons why transferring knowledge to China is not easy. And Chinese will understand.	PR16	1 (P2)
SL22	Do frequent meetings between Chinese and American developers to help establish relationships and trust.	-	1 (P5)

Table 13: Solutions related to “Transparency”

Delegation and Traveling

The majority of our interview partners (4, or 67%) stated the importance of delegation, as an important way to bridge cultures. The following examples say:

“I think expats are really great when it comes to conferences and research communities to show their presence. We had in February this conference in China where we had some internal software initiative and product development in free market. It was interesting to see that there were quite a lot of local expats. So they’ve been there for quite a while and now they ended up in some R&D positions ... And I think this is important within the community, not only in X1 but also within the knowledge management community that they are there and to show that X1 is also committed here [in China].”—P1, Senior Consultant

“In china it is quite hard to identify the ideal candidates, but in US they can easily find the candidates with similar backgrounds, with the similar industry domain. But in china because the product is not so popular, we cannot find the best candidates at least in this product domain. This made the US director reject a lot of the candidates in the beginning. But later after he travelled to China and talked to the people, to the local people, to the managers, to the directors, to other colleagues. He got to know the background in China, thus he had a reasonable idea, and reasonable expectations about the candidates. Finally we closed the position.”—P5, Development Manager

Table 14 shows a list of related problems that were explicitly mentioned by our interview partners.

ID	Problem	Participants
PR17	A few number of volunteers with intercultural experience want to go for delegation.	1 (P2)
PR18	Americans spend much time when hiring Chinese for certain positions, because they apply the same standards used for the American market.	1 (P5)

Table 14: Problems related to “Delegation and Traveling”

As for “PR17”, our interview partners mentioned that most people apologize for delegation due to family obligations. Moreover, as to “PR18”, one interview partner said that the lack of traveling between regions caused a gap of cultural understanding. Yet, after traveling to China, this gap started to diminish, because US managers had a closer look at the culture by meeting people in person. Then, they realized that applying the same standards for both countries is invalid. Thus, the cultural difference should be taken into consideration. Table 15 shows a list of suggested solutions by our interview partners to overcome the related problems.

Solution’s ID	Solution	Problem’s ID	Participants
SL23	Encourage traveling in both ways.	PR17,PR18	4 (P1,P2,P4,P5)
SL24	Remind of the benefits of delegation on the delegate’s carrier.	PR17	1 (P1)

Table 15: Solutions related to “Delegation and Traveling”

The data analysis showed that traveling and delegation help bridge cultures, and spread cultural awareness among collaborated members. Furthermore, they help in maintaining face-to-face communication, and building personal relationships with team members, as well as having an intermediary role between the headquarters and the subsidiary.

2.6 Limitation

A limitation of a GT research is that the findings are grounded in the specific contexts explored in the research. These contexts were dictated by our choice of research destination which was limited to the collaborations of German/Chinese and American/Chinese. Thus, we do not claim that our findings are generally applicable to all GSD projects, but they rather characterize the contexts studied. Moreover, we got only one perspective from each interview partner. We preferred to interview people from different projects, rather than concentrating only on one project. Yet, it might be better to get at least two perspectives from each partner, i.e. one Chinese and one German in German companies, and one Chinese and one American in American companies.

2.7 Conclusion

In this research, we presented a GT study that explored the problems and solutions of German/Chinese and American/Chinese collaborations. Our sample was gathered through six interviews with six participants in five multinational software companies that have development centers in China. Several studies were made in this area, however most of them applied pre-defined frameworks, e.g. Hofstede (2001), Hall (1977), and Hampden-Turner and Trompenaars (1997). Our results showed a theory of two main categories: *Communication*, and *Trust*. Each category contained a number of concepts, where each concept identified a set of problems, in addition to some adopted solutions. We believe that this theory may provide companies with in-depth insights about the problems they might encounter, along with solutions adopted by multinational software companies.

3 Elaboration of Research Chapter

In this chapter, we present the full data analysis of the conducted interviews. It serves as an extension of section 2.4 *Data Analysis*. We provide more quotations and evidence for each of the emerged concepts and categories. Appendix C shows the codes hierarchy in MAXQDA, and appendix D shows an illustration of the resulting theory.

3.1 Emergence of Categories and Concepts

3.1.1 Emergence of Communication

The category *Communication* emerged from three main concepts: *Communication Behaviors*, *Reporting Failure*, and *Collaboration across Regions*. To explain the emergence of the category, we start with full explanation of the emergence of each of the concepts.

Communication Behaviors

Each of the codes shown in table 16 was compared constantly with all other codes of other interviews to form a second level of abstraction or a *concept*. We believe that all of these codes, e.g. *Respecting hierarchy*, and *Open discussion*, have a common theme which is *Communication Behaviors*.

ID	Key Point	Code
P2	As a lead there [in China], I would just talk to a developer or a tester without bringing the team lead with me. But this is not normal in China.	Respecting hierarchy
P5	US developers are quite open. Whenever they have problems they will ask, whether it is valid or not. Sometimes, they even will raise questions to ask for clarifications.	Open discussion
P5	For example, people in the US, they speak in an informal style of communication, but in China the conversation especially the conversation at work is usually more formal, not only in certain professional areas, but almost all the conversation at work in China is more formal. This is different from the conversation with the US colleagues.	Formal vs. Informal
P3	Sometimes in the QA session, we will ask a developer [a Chinese developer]: do you have any questions, do you fully understand the requirements? And the answer most likely would be yes, okay.	Affirmation
P5	In China, if the developers don't have confidence on the idea, the solution, or the approach, they will not talk about it in the meeting.	Difficulty with Open discussion
P1	If a German product owner or whatever his role is tells a story, tells the requirements, and then asks for questions there [in China], there won't be any questions.	<ul style="list-style-type: none">• Open discussion• Respecting hierarchy

P5	I think the reason could be because the developers in China, they don't know much about all the products, and the features, especially in the design phase. They need time to get to know the requirements first.	Reason behind open Discussion
----	---	-------------------------------

Table 16: Samples of codes for the concept “Communication Behaviors”

Reporting Failure

Each of the codes shown in table 17 was compared constantly with all other codes of other interviews to form a concept. We believe that all of these codes, e.g. *Fear of losing face*, and *Reporting failure*, have a common theme which is *Reporting Failure*.

ID	Key Point	Code
P1	Some Germans would go to China to bring the message again and establish an atmosphere of trust ... and that reporting failure needn't to be punished, rather appreciated.	Appreciating reporting mistakes
P1	It comes to a different respect in China. Losing face occurs if they [Chinese developers] don't achieve the goal or what they were asked to deliver and then they are also managing to get the fear of communicating failure or problems.	Fear of losing face
P3	The Chinese always try to please the boss, right. So in trying to please the boss, and because they just feel if the boss sets a timeline on Friday, and they cannot deliver on Friday, they feel they have failed the boss and the boss will not be satisfied about their deliverables.	Fear of losing face
P4	If I [Chinese developer] find a problem that blocks my progress, I'll report this to my team leader. It's very easy. I just need to tell him, he's okay, I just go forward to his office. But in China that team leader will say hey, why do you – he's very strict on that.	Difficulty with reporting failure
P1	A huge benefit or a very important that is you try to keep in person. Not going over documents, not going over some reward session, but you may have a workshop with them [Chinese developers] where they can use maybe some anonymous means to articulate their ideas.	<ul style="list-style-type: none"> • Avoid losing face • Workshops
P3	It's a cultural thing, because in China, the developer likes to dig in. They love to dig in and figure out the problem, and solve it ... They trust themselves because they [Chinese developers] do this every day.	<ul style="list-style-type: none"> • Fear of losing face • Long after work hour • Digging into details
P3	Actually for the US -- for any culture you know, as long as you have a strong reason and you actually gave your best ... the manager or the boss can actually accept it, but in China, the culture is if you cannot deliver, whatever or how hard you try if you still can-not deliver, that's bad.	Fear of losing face

Table 17: Samples of codes for the concept “Reporting Failure”

Collaboration across Regions

Each of the codes shown in table 18 was compared constantly with all other codes of other interviews to form a concept. We believe that all of these codes, e.g. Big picture vs. Detail oriented, and *Misunderstandings*, have a common theme which is *Collaboration across Regions*.

ID	Key Point	Code
P4	For Chinese, they're always like to do what is the other saying but for U.S. they're more curious about everything. I think the difference is due to the different cultures.	Big picture vs. Detail oriented
P5	When the people from the US talk about certain topics they will give a more context, more information about the background of the topic, very clear information about that to avoid miscommunication. But in China it is a bit different because people really talk based on some context, some existing background and they assume that people already know.	<ul style="list-style-type: none"> • Misunderstandings • Need for context
P6	It's specifically related to the company's terminology. For the Chinese company, they have the terminology written in Chinese, and in German for the German team. So, when they tried to translate from Chinese – English, and from German – English, they got different results. This also caused delays, and additional costs.	<ul style="list-style-type: none"> • Misunderstandings • Terminology
P5	Also internally we have some small groups to discuss technical challenges. That has helped a lot. People in China, in the local team, they like to discuss more technical challenges internally or externally with the US developers.	<ul style="list-style-type: none"> • To be challenged • Technical groups
P6	In comparison to Germans, the number of Germans who speak English are more than Chinese.	Language barrier
P4	The developers in China sometimes intend to work out or find out the requirement by themselves.	<ul style="list-style-type: none"> • Defining requirements • Clarification
P1	And there is a chance to create some features and some definitions that are quite narrow in the scope but quite as deep in implementation. This is at least what I experienced and past showed us. It is just working just well in China.	Tasks scope

Table 18: Samples of codes for the concept “Collaboration across Regions”

Eventually, we compared all concepts of all interviews constantly to form a third level of abstraction or a *category*. Thus, we believe that the explained concepts above have a common theme which is *Communication*. Figure 3 illustrates the emergence of the category.

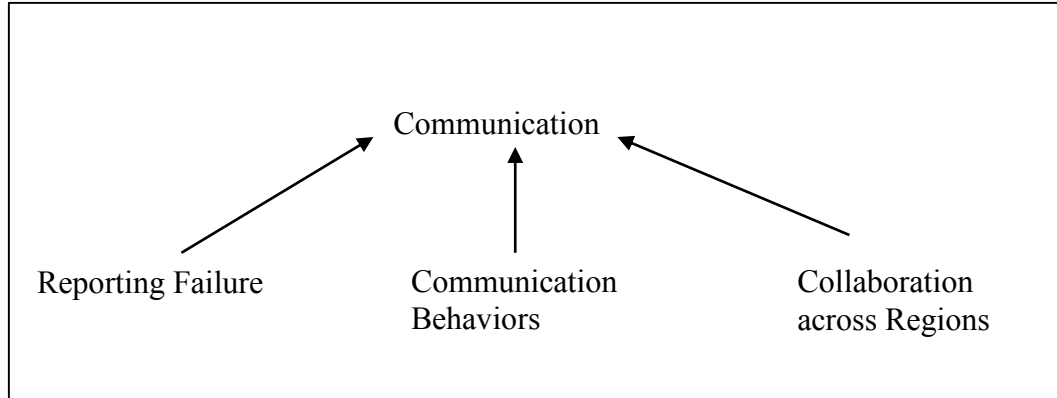


Figure 3: Illustration of the emergence of “Communication”

3.1.2 Emergence of Trust

The category *Trust* emerged from two main concepts: *Transparency*, and *Delegation and Traveling*. To explain the emergence of the category, we start with full explanation of the emergence of each of the concepts.

Transparency

Each of the codes shown in table 19 was compared constantly with all other codes of other interviews to form a concept. We believe that all of these codes, e.g. *Visibility of work*, and *Trust and control*, have a common theme which is *Transparency*.

ID	Key Point	Code
P3	If it is the white box, I already know what teams are responsible for and what’s the progress they have made, another team will feel fine, if I am with you or them that’s the best I can do, so they have that kind of trust.	<ul style="list-style-type: none"> • Transparency • Visibility of work
P2	We [Germans] said okay, we will not give you [Chinese] the software in China because it's recently developed in a university and therefore we offer you a German colleague, who implemented it, to implement it for your device, and that was a solution. And I think the colleagues in China understood that because we also have some restrictions within Germany	<ul style="list-style-type: none"> • Transparency • Avoiding Mis-trust
P3	We talk about the meeting notes, so those information would be public for the whole team. So anytime you have time or you have interest, you know you have the place.	<ul style="list-style-type: none"> • Transparency • Visibility of work

P2	What we did is we developed recently a project relay for the Chinese market and it seems to be quite successful there, and before, when we started, it was more or less a workbench for Germany also in the area of protection and power quality and substation automation.	<ul style="list-style-type: none"> • Transparency • Reducing turnover
P6	We [Germans] maintain a transparent relationship between both sides. The relationship is based on trust and control. We can't trust 100%, especially when it comes to intellectual property rights and knowledge transfer.	<ul style="list-style-type: none"> • Establishing trust • Trust and control

Table 19: Samples of codes for the concept "Transparency"

Delegation and Traveling

Each of the codes shown in table 20 was compared constantly with all other codes of other interviews to form a concept. We believe that all of these codes, e.g. *Delegation*, *Traveling*, and *Cultural awareness*, have a common theme which is *Delegation and Traveling*.

ID	Key Point	Code
P2	A delegation process is normally two to three years and after two to three years you have to go back ... in Germany, the last one [Chinese] was in Germany for half a year and now he went back to China and is now in development in Nanjing again.	Delegation duration
P4	In my opinion, I think that communication is very, very important. Although we can call over the phone every two days, but I don't think it's as efficient as we face to face.	<ul style="list-style-type: none"> • Traveling • Face-to-Face communication
P2	We constantly ask our people [Germans] who want to go to China and who want to go to Germany from the Chinese and to be honest you don't have too much volunteers on each side.	Difficulty with delegation
P5	In China it is quite hard to identify the ideal candidates, but in US they can easily find the candidates with similar backgrounds, with the similar industry domain. But in China because the product is not so popular, we cannot find the best candidates at least in this product domain. This made the US director reject a lot of the candidates in the beginning. But later after he travelled to China and talked to the people, to the local people, to the managers, to the directors, to other colleagues. He got to know the background in China, thus he had a reasonable idea, and reasonable expectations about the candidates. Finally we closed the position.	<ul style="list-style-type: none"> • Traveling • Cultural awareness • Hiring
P2	Yeah we often had, at least at the beginning, we had a lot of traveling from China to Germany, and vice-versa.	<ul style="list-style-type: none"> • Traveling • Cultural awareness

Table 20: Samples of codes for the concept “Delegation and Traveling”

Eventually, we compared all concepts of all interviews constantly to form a category. Thus, we believe that the explained concepts above have a common theme which is *Trust*. Figure 4 illustrates the emergence of the category.

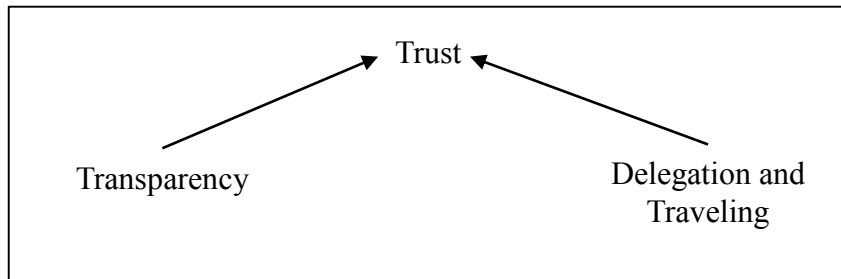


Figure 4: Illustration of the emergence of “Trust”

3.2 Profiles

Table 21 shows the full profiles of our projects. Each profile contains relevant information about each project, e.g. sector, software methodology, projects’ distribution, challenges and problems, and some implemented solutions and best practices. We believe that such profiles may provide software companies, which have interests in extending their business to China, with a breadth and depth of insight of the problems they may encounter along with the used solutions to overcome them.

Company Participant	Sector	Software Methodology	Projects' Distribution	Project's Duration	Team Size	Development Centers	Problems	Solutions
X1 P1	Consulting	Scrum ⁶	Germany – China – India – Slovakia	24 months	Germany (12); China (3); India (10); Slovakia (8)	Germany (1); China (1); India (1); Slovakia (1);	<i>Communication:</i> <ul style="list-style-type: none"> Chinese developers are more reluctant than Germans to report mistakes during projects. Chinese developers have less tendency than Germans towards asking questions in group meetings. Chinese developers seldom argue or discuss their tasks with their superiors. Chinese developers need detailed requirements about their tasks. 	<i>Communication:</i> <ul style="list-style-type: none"> Ask management to appreciate reporting mistakes during projects. Choose software development method that encourages communication, e.g. Scrum, or other agile methods. Create workshops to articulate ideas with anonymous identities, because anonymity helps Chinese avoid sharp direct feedback. Create smaller and deeper tasks deliberately and associate them with specs and context.

⁶ K. Schwaber and M. Beedle. Agile Software Development with Scrum. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

							<i>Trust:</i>	<i>Trust:</i>
							<ul style="list-style-type: none">Chinese have a higher staff turnover rate than Germans.Transferring knowledge to China is a serious problem, because of the low IPR value.	<ul style="list-style-type: none">Encourage traveling in both ways.Build relationships by encouraging traveling during projects.Remind of the benefits of delegation on the delegate's carrier.
X1 P2	Energy & Automation	Plan-driven	Germany – China – India	12 – 15 months	Germany (350); China (140); India (250)	Germany (2); China (2); India (2)	<i>Communication:</i> <ul style="list-style-type: none">Chinese have much higher respect for hierarchy than Germans.Chinese have indirect behaviors when reporting issues, which is different from Germans.	<i>Communication:</i> <ul style="list-style-type: none">Ask the management to bring up repeatedly the message of the importance of open conversation.
							<i>Trust:</i>	<i>Trust:</i>
							<ul style="list-style-type: none">Chinese have a higher staff turnover rate than Germans.A few number of volunteers with in-	<ul style="list-style-type: none">Launch a product line for the Chinese market to build loyalty and trust. This dramatically reduced the high turnover rate for Chinese.

							<p>tercultural experience want to go for delegation.</p> <ul style="list-style-type: none"> Transferring knowledge to China is a serious problem, because of the low IPR value. 	<ul style="list-style-type: none"> Support traveling in both ways to establish relationships. Establish a transparent relationship with Chinese. For example, explain to Chinese the reasons why transferring knowledge to China is not easy. And Chinese will understand.
X2 P3	Platforms	Agile & Plan-driven	US – China – India	12 – 18 months	China (10); US (5); India (10)	China (2); US (2); India (1)	<p><i>Communication:</i></p> <ul style="list-style-type: none"> Chinese make late notifications if not being able to meet the deadline. Yet, they are willing to spend long after work hours to finish the task. Chinese developers have less tendency towards asking questions in group meetings. 	<p><i>Communication:</i></p> <ul style="list-style-type: none"> Apply a progress-tracking system, where a developer updates the status of his task daily, to avoid late notifications. Bring up the message that delays are acceptable in presence of a strong justification. Do not take “yes” or “no” for an answer. Chinese developers should write a document

	<ul style="list-style-type: none"> • Chinese’ “yes” or “no” has a different meaning for Americans. • Chinese cannot easily communicate with Americans due to language barriers. • Chinese are detail-oriented, while Americans see the big picture. • Chinese developers sometimes, due to lack of requirements, fill some requirements based on assumption or understanding. 	<p>of their opinion after important sessions.</p> <ul style="list-style-type: none"> • Split up the development cycle across regions, where you can exploit the benefits of each one, e.g. Americans deal with customers, Chinese design, and Indians implement. • Pay more attention when defining requirements in order not to leave any space for guessing. <hr/> <p><i>Trust:</i></p> <ul style="list-style-type: none"> • Make the work visible. Thus, all collaborating regions can see what other regions do.
--	---	---

X3 P4	Platforms	Scrum	US – China – Canada	6 – 12 months	China (20); US (50 – 60); Canada (10)	China (1); US (2); Canada (1)	<i>Communication:</i> <ul style="list-style-type: none"> • Strictness of Chinese team leaders makes it difficult for some developers to report mistakes or delays. • Chinese are detail-oriented, while Americans see the big picture. • Chinese cannot easily communicate with Americans due to language barriers. • The US and China have different holidays. E.g. spring festival in China, and Christmas in the US. 	<i>Communication:</i> <ul style="list-style-type: none"> • Assign a local Chinese expert to get back to in case of problems. • Use a wiki-like system for technical discussions. Yet, it is not useful in case of urgent issues. <hr/> <i>Trust:</i> <ul style="list-style-type: none"> • Support traveling in both ways. It helps maintain face-to-face communication, and establish relationships.
X4 P5	Enterprise Solutions	Plan-driven	US – China – India	6 – 12 months	China (160); US (300); India (500)	China (1); US (1); India (1)	<i>Communication:</i> <ul style="list-style-type: none"> • Chinese are more reluctant to ask questions in group meetings than Americans. • Chinese developers do not raise issues directly, which is different 	<i>Communication:</i> <ul style="list-style-type: none"> • Bring up the message of the importance of asking questions regardless of how big or small they are. • Remind Chinese developers repeat-

	<p>from the direct American style.</p> <ul style="list-style-type: none"> • Chinese have a formal communication style, while Americans have an informal style. • Chinese developers omit the context when discussing tasks with Americans. • Chinese developers like to be challenged. 	<p>edly of the importance of providing context when discussing tasks.</p> <ul style="list-style-type: none"> • Create small discussion groups, where Chinese developers can discuss technical problems with their American colleagues. • Provide Chinese developers with challenging tasks regularly. • Create a relaxing work environment for Chinese developers, where you can speak freely and informally with colleagues, and formally only with clients.
--	---	--

							<i>Trust:</i>	<i>Trust:</i>
							<ul style="list-style-type: none"> Americans spend much time when hiring Chinese for certain positions, because they apply the same standards used for the American market. 	<ul style="list-style-type: none"> Support traveling to China, because it helps know the Chinese market better, and what to expect. Do frequent meetings between Chinese and American developers to help establish relationships and trust.
X5 P6	Electronics	Plan-driven	Germany – China – France – Sweden	9 – 36 months	China (3 – 10); France (3 – 10); Sweden (3 – 10)	Germany (1 – 3); China (1 – 2); France (1); Sweden (1 – 2)	<i>Communication:</i>	<i>Communication:</i>
							<ul style="list-style-type: none"> The terminology is documented only in the country's language, i.e. Chinese in China and German in Germany. A few Chinese developers can speak English. 	<ul style="list-style-type: none"> Create English training sessions in both sides. The training focuses on business terms used in the industry. Unify the terminology by creating a map table in the three languages: English, Chinese, and German, where each term is associated with its corresponding in English.

	<i>Trust:</i>	<i>Trust:</i>
	<ul style="list-style-type: none"> • Transferring knowledge to China is a serious problem, because of the low IPR value. 	<ul style="list-style-type: none"> • Establish a bi-directional relationship based on trust and control.

Table 21: Full overview of the profiles of the projects

References

- Abraham, L. R. (2009). Cultural differences in software engineering. *Proceeding of the 2nd Annual Conference on India Software Engineering Conference - ISEC '09*, 95. doi:10.1145/1506216.1506234
- Accenture. (2006). *Improved Cross-Cultural Communication Increases Global Sourcing Productivity*. Retrieved from http://newsroom.accenture.com/article_display.cfm?article_id=4376
- Allan, G. (2003a). A critique of using grounded theory as a research method. *Electronic Journal of Business Research Methods*, 2(1), 1–10.
- Allan, G. (2003b). The use of grounded theory as a research method: warts all. In *European Conference on Research Methodology for Business and Management Studies* (pp. 9–19).
- Bird, C., Nagappan, N., Devanbu, P., Gall, H., & Murphy, B. (2009). Does distributed development affect software quality? An empirical case study of Windows Vista. *2009 IEEE 31st International Conference on Software Engineering*, (May), 518–528. doi:10.1109/ICSE.2009.5070550
- Boden, A., Avram, G., Bannon, L., & Wulf, V. (2012). Knowledge sharing practices and the impact of cultural factors: reflections on two case studies of offshoring in SME. *Journal of Software: Evolution and Process*, 24(2), 139–152. doi:10.1002/smr.473
- Brockmann, P. S., & Thaumuller, T. (2009). Cultural Aspects of Global Requirements Engineering: An Empirical Chinese-German Case Study. *2009 Fourth IEEE International Conference on Global Software Engineering*, 353–357. doi:10.1109/ICGSE.2009.55
- Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*. Prentice Hall PTR.
- Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *Software, IEEE*, 18(2), 22–29.
- Casey, V. (2009). Leveraging or Exploiting Cultural Difference? In *2009 Fourth IEEE International Conference on Global Software Engineering* (pp. 8–17). IEEE. doi:10.1109/ICGSE.2009.9
- Cater-Steel, A., & Toleman, M. (2008). The impact of national culture on software engineering practices. *International Journal of Technology, Policy and Management*, 8(1), 76–90. doi:10.1504/IJTPM.2008.016182
- Damian, D. E., & Zowghi, D. (2003). RE challenges in multi-site software development organisations. *Requirements Engineering*, 8(3), 149–160.
- Dhir, S. K., Sahay, S., & Walsham, G. (2004). Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, 47(4), 62–66.

- Dorairaj, S., Noble, J., & Malik, P. (2011). Bridging cultural differences: A grounded theory perspective. In *Proceedings of the 4th India Software Engineering Conference* (pp. 3–10). ACM.
- Ebert, C., & De Neve, P. (2001). Surviving global software development. *Software, IEEE*, 18(2), 62–69.
- Georgieva, S., & Allan, G. (2008). Best practices in project management through a grounded theory lens. *Electronic Journal of Business Research Methods*, 6(1), 43–52.
- Glaser, B. G. (1965). The constant comparative method of qualitative analysis. *Social Problems*, 12(4), 436–445.
- Glaser, B. G. (1992). *Emergence vs forcing: Basics of grounded theory analysis*. Sociology Press.
- Glaser, B. G. (1998). *Doing grounded theory: Issues and discussions* (Vol. 254). Sociology Press Mill Valley, CA.
- Glaser, B. G., & Strauss, A. L. (2009). *The discovery of grounded theory: Strategies for qualitative research*. Transaction Books.
- Hall, E. T. (1977). *Beyond culture*. Random House Digital, Inc.
- Hampden-Turner, C., & Trompenaars, F. (1997). Response to Geert Hofstede. *International Journal of Intercultural Relations*, 21(1), 149–159.
- Herbsleb, J. D., & Mockus, a. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29(6), 481–494. doi:10.1109/TSE.2003.1205177
- Herbsleb, J. D., Mockus, A., Finholt, T. A., & Grinter, R. E. (2001). An empirical study of global software development: distance and speed. In *Proceedings of the 23rd international conference on Software Engineering* (pp. 81–90). IEEE Computer Society.
- Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software*, 18(2), 16–20. doi:10.1109/52.914732
- Herbsleb, J. D., Paulish, D. J., & Bass, M. (2005). Global software development at siemens: experience from nine projects. *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005.*, 524–533. doi:10.1109/ICSE.2005.1553598
- Ho, D. Y. (1976). On the concept of face. *American Journal of Sociology*, 867–884.
- Hofstede, G. H. (2001). *Culture's consequences: Comparing values, behaviors, institutions and organizations across nations*. Sage.
- Holmstrom, H., Conchúir, E. Ó., Ågerfalk, P. J., & Fitzgerald, B. (2006). Global Software Development Challenges : A Case Study on Temporal , Geographical and Socio-Cultural Distance.
- Hutchins, E. (1995). Cognition in the Wild. 1995. *MIT Press, Cambridge, USA*, 14, 399–406.

- Irani, L. C., & Dourish, P. (2009). Postcolonial interculturality. In *Proceedings of the 2009 international workshop on Intercultural collaboration* (pp. 249–252). ACM.
- 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. doi:10.1177/0018726702551004
- OSF Global Services. (2012). *Overcome cultural differences in the outsourcing process*. Retrieved from http://www.osf-global.com/assets/uploaded_files/en/overcome-cultural-differences-OSF-white-paper.pdf
- Ramasubbu, N., & Balan, R. K. (2007). Globally distributed software development project performance: an empirical analysis. In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* (pp. 125–134). ACM.
- Schreiber, R. S., & Stern, P. N. (2001). *Using grounded theory in nursing*. Springer Publishing Company.
- Shah, H., & Harrold, M. J. (2013). Culture and Testing: What is the Relationship? In *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on* (pp. 51–60). IEEE.
- Shore, B. (1996). *Culture in Mind: Cognition, Culture, and the Problems of Meaning*. Oxford University Press.

Appendix A – Informed Consent

Research

A Study on the impact of cross-cultural differences of German/Chinese and American/Chinese collaborations on GSD. In brief, we are trying to know more about the challenges and problems encountered by the collaborations between both cultures in software development, in addition to solutions developed to overcome such problems.

For this reason, we conduct a qualitative research, where we collect our data through interviews with German and American software companies that have subsidiaries in China. Therefore, we get to know more about the problems they faced or still facing through the collaborations, as well as the solutions they implemented. Afterwards, we perform a text-analysis method to analyze the interviews, and eventually to form a theory of challenges and Solutions.

Structure

- The interview will be conducted in English.
- The interview could be conducted in person, over the phone or Telco.
- The duration of interview is expected to be around 1 hour.
- Unless you give us permission to use the information you tell in the interview in any publications that may result from this research, such information will be confidential.
- We would like to audio record the interview so that we can use it for analysis while proceeding with the research study. We will not record the interview without your permission.
- The interview will be as a normal conversation, where we would like to listen to your experience in German/Chinese (or American/Chinese) collaborations in software development.

Interviewer

Bilal Zaghloul
MSc Student of International Information Systems (IIS)
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
bilal.zaghloul@studium.uni-erlangen.de
(CN) +86 13261447847

Supervisors

Prof. Dr. Dirk Riehle
Friedrich-Alexander-Universität Erlangen-Nürnberg
dirk.riehe@fau.de, <http://osr.cs.fau.de>

Prof. Minghui Zhou, Ph.D.
Peking University
zhmh@pku.edu.cn

Appendix B – Interview Questions

1st & 2nd Interviews

Opening Questions

1. Can you give me a brief description of your job title and job responsibilities?
2. For how long does Siemens Software department operate in China? And is there more than one office for software development in China?
3. What kind of software projects are you doing?
4. Do you remember a particular incident caused by Chinese/German cooperation? And how did it affect the project? Upon the answer, I might know which dimensions to start with.
5. In your opinion, what do productivity and software quality mean to you? And how do you measure them?

Hierarchy

1. Could you tell me about the hierarchy of the software development department?
2. When assigning tasks, which developers (German/Chinese) tend to argue and discuss tasks with their supervisor? And which ones tend to obey without argument, even if they might have different perspectives?
3. In your experience, do you remember any issues caused by this matter? And what was the impact on productivity and software quality?

Assigning Tasks

1. Can you tell me how are tasks assigned to developers? And do they usually have dependencies on each other?
2. Which developers (German/Chinese) prefer to discuss their tasks with other developers?
3. In your opinion, is it better to assign independent/dependent tasks to developers? And how would it affect the productivity and software quality? Any strategy you use?
4. Which developers (German/Chinese) are more precise regarding details? Example?
5. Which developers (German/Chinese) are more careful and accurate in delivering bug-free tasks? And what is the impact on productivity and software quality?

Software Methodology

1. What kind of software development methodology you use? Which ones (German/Chinese) are adapted more to it?
2. Do you encounter problems related to employees' turnover? If yes, which ones (German/Chinese) have a higher turnover rate in your opinion? Why? And how does it affect the productivity and software quality?

Temporal Distance

1. Did you encounter problems related to time-zone differences? If yes, examples? And how did you resolve them? (Solutions)
2. In your perspective, did such problems affect productivity and software quality? And how?

Communication

1. How do you communicate with developers regarding tasks? And how often? Best practices?
2. What kind of problems did you encounter as a consequence of miscommunication? And how did you resolve them?
3. What is its impact on the productivity and software quality?

Geographic Dispersion

1. Is there a certain strategy you use to form teams? In other words, are there teams with members dispersed between China and Germany? If yes, how do you establish a feeling of “teamness” between them?
2. What kind of problems did you encounter regarding this matter? And how did you control them?
3. In your opinion, how critical is it to control the dispersion for the productivity and software quality?

3rd & 4th Interviews

Opening Questions

1. Background
2. Mutual software projects between German and Chinese
3. Problems from both sides caused by the collaborations

At the Beginning of a Project

1. How do you present a new project? E.g. group meetings, shared documents, emails, etc.
 - a. Do you go into details from the beginning?
 - b. What is the reaction after the project’s proposal? E.g., questions, raising problems, etc.
 - c. How frequent is it to raise questions or issues about the project at the beginning?
 - d. How would problems regarding this matter affect the project eventually?
 - e. Which solution did you implement?
2. How differences in your opinion surface, are expressed?
3. How do project managers receive feedback and critique?

4. How do you assign tasks?
5. How much detail is provided for each task?
 - a. Examples
 - b. How would problems regarding this matter affect the project eventually?
 - c. What solution/best practice do you have?
6. How do you form teams?
 - a. Overview about the team structure
 - b. Which skills are required for team members?
 - c. How frequent do you change teams? Does this depend on the project?
 - d. What are the effects on the project eventually?

During a Project

1. How do team members communicate with each other?
2. How frequent is it to find issues during a project?
3. How do developers react if they discovered a problem during implementation?
 - a. How does the management handle this?
 - b. How did this affect the project?
 - c. How did you solve it?
4. How frequent is it to exchange/hand-over tasks between developers?
 - a. How do developers react to this?
 - b. How long does it take to do it? Is it easy?
 - c. How would this affect the project?
 - d. How did you solve it?
5. How high is the turn-over rate, especially during the project?
 - a. Reasons/assumptions
 - b. Age
 - c. What privileges/benefits do they seek?
 - d. What did you do to recover?
 - e. How did it affect the project?
 - f. How did you solve it?
6. How many project do you have for local products in China?
 - a. Examples of products
 - b. How do employees feel about that?
 - c. Relation between local production and turn-over rate
 - d. How would that affect the project?
7. What do delegates do to bridge the gap between both countries?
 - a. How many delegates do you have in both sides?

- b. Which positions do they have?
- c. Duration of stay

5th & 6th Interviews

Opening Questions

- 1. Background
- 2. Mutual software projects between the US and China
- 3. Problems from both sides caused by the collaborations

Time Difference

- 1. How do you handle the time-difference between the US and China?
- 2. What are the effects on the project/product?
- 3. How do developers (Chinese/American) react to the time-difference?
- 4. What did you do to solve this problem?

Development Life Cycle

- 1. Could you tell me about the phases of the development's life cycle? Explain each briefly.
- 2. How do you manage the whole phases across both countries?
- 3. How do you divide the phases between regions? Reasons?
- 4. How can you manage the collaboration process between both regions?
- 5. What are the consequences on the product/project if something went wrong? Any examples?
- 6. What are the solutions you implemented or best practices you're using to avoid incidents?

Quality Definitions

- 1. Any idea, what are the most things an (American/Chinese) developer care about when working on a certain task? Any differences?
- 2. In case of any differences, how would it affect the project? Pros & Cons?
- 3. What is the practice you're using to manage the differences?
- 4. Do you remember a certain incident that was caused by such differences?

During the Development

- 1. How frequent is the interaction between both regions during development? Opinions?
- 2. What are the differences between the Chinese behaviors compared to the American's one?

3. How frequent is it to raise questions or initiate discussions about specific tasks/problems? Any differences between Americans and Chinese.
4. How do you maintain a balance between both behaviors?
5. Do you remember a certain incident caused by such differences?
6. If any, what were the effects on the project/product?
7. What did you do to solve it?

Reporting Mistakes

1. How do team members communicate with each other?
2. How frequent is it to find issues during a project?
3. How do developers react if they discovered a problem during implementation?
 - a. How does the management handle this?
 - b. How did this affect the project?
 - c. How did you solve it?

Capturing Requirements

1. How do you set the requirements for a certain project? And how do you pass the requirements to the developers?
2. How can you ensure that the developers have understood the task's requirements? Any observations on the Americans/Chinese?
3. How frequent is it for misunderstanding to occur for the Americans/Chinese? Opinions or assumptions?
4. How do you maintain the process?
5. Can you remember a certain incident and its effect on the project?
6. What is the practice you're using to manage this process?

Innovation

1. In your opinion, which developers (Chinese or Americans) are more interested to use the latest technology in projects? Assumptions?
2. What do you do to ensure using new technologies in your products?
3. What would happen if something went wrong? Any incidents?
4. If any, how did you solve it?

Meeting the Deadline

1. How often is it for developers to commit to deliver their tasks on time? Any observations about Chinese or Americans?
2. What would developers do if they can't meet the deadline? Differences between Chinese and Americans.

3. How often is it for Chinese developers to report/admit mistakes to their managers compared to Americans?
4. How would project managers or teams leads receive delays?
5. Do you remember any incident occurred regarding this matter?
6. If any, how did you solve it?

Trust

1. How do you establish trust between Chinese and Americans?
2. Do you remember any incidents occurred regarding this matter?
3. If any, what did you do (solutions)?

Turnover Rate

1. How high is the turn-over rate, especially during the project?
 - a. Reasons/assumptions
 - b. Age
 - c. What privileges/benefits do they seek?
 - d. What did you do to recover?
 - e. How did it affect the project?
 - f. How did you solve it?

Appendix C – Code Hierarchy

Category	Concept	Code	Number of Quotations
Communication	Reporting Failure	Appreciating reporting mistakes	2
		Scrum encourages reporting failures	1
		Fear of losing face	4
		<ul style="list-style-type: none"> Fear of losing face Late notifications 	3
		<ul style="list-style-type: none"> Fear of losing face Long after work hours 	3
		<ul style="list-style-type: none"> Fear of losing face Long after work hours Digging into details 	7
		<ul style="list-style-type: none"> Avoiding losing face Tracking progress 	3
		<ul style="list-style-type: none"> Avoiding failures Tracking progress 	4
		<ul style="list-style-type: none"> Avoiding losing face Workshops 	3
		Difficulty with reporting failures	2
		<ul style="list-style-type: none"> Reporting failures Local experts 	1
	Communication Behaviors	<ul style="list-style-type: none"> Respecting hierarchy Open discussion 	5
		Respecting hierarchy	1
		Open discussion	2
		Reason behind open discussion	1
		Effects of open discussion	3
		Difficulty with open discussion	2
		<ul style="list-style-type: none"> Open discussion Improvement 	1
		Direct vs. Indirect	5

	Formal vs. Informal	11
	<ul style="list-style-type: none"> • Behavior changes • Experience 	1
	Affirmation	6
Collaboration across Regions	Big picture vs. Detail oriented	11
	<ul style="list-style-type: none"> • Defining requirements • Capturing requirements 	2
	<ul style="list-style-type: none"> • Defining requirements • Avoiding missing 	11
	<ul style="list-style-type: none"> • Defining requirements • Clarification 	1
	Avoiding misunderstandings	4
	Benefits of collaboration across regions	12
	Risks of collaboration across regions	1
	Tasks Scope	1
	Effects of Scrum	4
	<ul style="list-style-type: none"> • Difficulty with Scrum • Exchange tasks 	1
	Language barrier	7
	Imbalance of language skills	1
	<ul style="list-style-type: none"> • Language barrier • Trainings 	3
	Effects of the language barrier	1
	<ul style="list-style-type: none"> • Communication between developers • Channels 	4
	Problems of communication channels	2
	<ul style="list-style-type: none"> • Misunderstandings • Need for context • Improvement 	1
	<ul style="list-style-type: none"> • Misunderstandings • Need for context 	8
	<ul style="list-style-type: none"> • Misunderstandings • Terminology 	2
	<ul style="list-style-type: none"> • Misunderstandings • Terminology 	1

		<ul style="list-style-type: none">• Map table	
		Holiday problems	1
		<ul style="list-style-type: none">• To be challenged• Discussion groups	1
		To be challenged	2
Trust	Transparency	<ul style="list-style-type: none">• Establishing trust• Trust and control	3
		<ul style="list-style-type: none">• Establishing trust• IP Rights	2
		<ul style="list-style-type: none">• Transparency• Reducing turnover	1
		<ul style="list-style-type: none">• Transparency• Visibility of work	8
		<ul style="list-style-type: none">• Transparency• Avoiding Mistrust	1
	Delegation and Traveling	Traveling frequency	3
		<ul style="list-style-type: none">• Traveling• Face-Face communication	3
		<ul style="list-style-type: none">• Traveling• Cultural awareness• Hiring	2
		<ul style="list-style-type: none">• Traveling• Cultural awareness	7
		<ul style="list-style-type: none">• Delegation• Cultural awareness	2
		Difficulty with delegation	4
		Delegation prerequisites	1
		Delegation duration	3
		Challenges of establishing trust	1
		Delegation frequency	3
Delegation benefits	1		

Appendix D – Illustration of Resulting Theory

