

Analyzing the Impact of Chinese Cultural Factors on Agile Software Development

Master Thesis by Xiuyu Qiao

Master Thesis

Master of Science in Business Informatics

Utrecht University

Department of Information and Computing Sciences

Faculty of Science

July, 2018





Universiteit Utrecht

Department of Information and
Computing Sciences
Faculty of Science
Utrecht University

Author

Xiuyu Qiao

Student number:

5842107

Email:

x.qiao@students.uu.nl

First Supervisor

Name:

Dr. Sietse Overbeek

Department:

Information and Computing Sciences

Utrecht University

Email:

s.j.overbeek@uu.nl

Second Supervisor

Name:

Dr. Sergio España Cubillo

Department:

Information and Computing Sciences

Utrecht University

Email:

s.espana@uu.nl

ABSTRACT

In the past decades, the IT industry has applied agile software development (ASD) methods because of the need to cope with constant change. ASD has become a common practice in western countries, yet Chinese Software Producing Organizations (SPOs) are still having problems with agile adoption due to cultural factors. To stay competitive in a global market, Chinese SPOs, are required to adapt to market needs swiftly. As a result, for Chinese SPOs, adopting ASD methods rigorously and reasonably is a critical issue to be solved.

This research project summarises the relationship between Chinese culture and ASD theories. The goal is to investigate how the cultural dimensions based on the theory of Hofstede (2011) interplay with existing agile values and principles, to better support Chinese SPOs in customising their agile implementation. The research is done by answering the main research question: “How do Chinese cultural factors influence the adoption of ASD approaches within Chinese SPOs?”

By conducting the literature review, a theoretical framework of the research is built. We made hypotheses between Chinese cultural factors and agile practices in the beginning. With data collected by four in-depth expert interviews involving different roles in agile development teams in China, fragments in existing software development methods which would harm the agile implementation in China are identified, and suggestions are provided to better avoid those impeding factors in the design of the software development methods in China.

In conclusion, it can be seen that high-Power Distance in China has a conflict with agile implementation, while its low-Uncertainty Avoidance, Collectivism, Masculinity, and Long-term Orientation are beneficial for agile implementation.

Keywords: *Agile software engineering, Hofstede cultural dimensions, Chinese culture, agile adoption, Chinese software development*

ACKNOWLEDGEMENT

The master thesis is the final deliverable of my Master of Business Informatics. During the past two years, I found it meaningful to study in the Netherlands. This thesis combines the knowledge of software product management and Chinese culture. To be honest, I feel fairly happy to conduct a project which I am really fond of.

I would like to take advantage of this opportunity to thank the people who helped me during the graduation project. First and foremost, I appreciate my first supervisor, dr. Sietse Overbeek from Utrecht University. It was his patience and earnestness that encouraged and inspired me to finish the thesis. He made efforts to negotiate with other colleagues and coordinate schedule to help me graduate on time. I would also like to thank my second supervisor, dr. Sergio España Cubillo from Utrecht University. Both of my supervisors offered their helpful feedback on my project. Besides, I would like to thank my colleagues, the students at Master in Business Informatics, for their advice and friendliness. Also, thanks to the unsung heroes participating in the research. Your valuable advice contributed to the success of the research.

Last but not least, I would like to thank for the support of my family.

Thank you all.

TABLE OF CONTENTS

ABSTRACT	II
ACKNOWLEDGEMENT	III
TABLE OF CONTENTS	IV
1. INTRODUCTION	1
1.1 Research Background	1
1.2 Problem Statement	1
1.3 Research Objectives	2
1.4 Research Questions	3
1.5 Research Contributions	4
1.6 Research Outline	4
2. RESEARCH METHOD	5
2.1 The Design of Empirical Cycle	5
2.2 Research Approaches	6
2.2.1 Literature Review	7
2.2.2 Semi-Structured Interview	8
2.2.3 Situational Method Engineering	8
2.3 Research Validity	9
2.4 Research Execution	9
3. LITERATURE STUDY	12
3.1 Agile Software Development	13
3.2 Hofstede's Cultural Dimensions	16
3.3 Chinese Culture	18
3.4 Bimodal IT	20
3.5 Software Product Management	21
3.6 Research Gap Analysis	22
4. ASSUMED RELATIONSHIP BETWEEN AGILE THEORIES AND CHINESE CULTURE	24
4.1 Power Distance and Self-organizing Teams	24
4.2 The Attitude towards Changing Requirements	24
4.3 The Impact of Masculinity on Development	25
4.4 The Conflict between Individuals and Team Cooperation	25
4.5 Long-term Orientation and Customer Collaboration	26
4.6 Summary of Assumptions	26
5. SOFTWARE DEVELOPMENT PRACTICES IN CHINESE SPOS	27
5.1 Alpha Software Company	27
Software Development Workflow	27
Supporting Infrastructure	30
Insights into Chinese Culture	32
5.2 Beta Software Company	33
Software Development Workflow	33
Supporting Infrastructure	35
Insights into Chinese Culture	36

5.3 Gamma Software Company	37
Software Development Workflow	37
Supporting Infrastructure	39
Insights into Chinese Culture	40
5.4 Delta Software Company	41
Software Development Workflow	41
Supporting Infrastructure	43
Insights into Chinese Culture	43
6. SUMMARY OF THE FINDINGS	44
7. FINDINGS AND ANALYSIS	46
7.1 PDI Hypothesis	46
7.2 UAI Hypothesis	47
7.3 MAS Hypothesis	47
7.4 IDV Hypothesis	47
7.5 LTO Hypothesis	48
8. DISCUSSION	49
8.1 Does Chinese Traditional Culture Really Impede Agile Implementation?	49
8.2 How to Use the Results?	50
8.3 Limitations	50
9. CONCLUSION AND FUTURE WORK	51
9.1 Conclusion	51
9.2 Future Work	52
REFERENCES	53
APPENDICES	58
Appendix A. Interview Protocol	58
Appendix A-1: English Version	58
Appendix A-2: Chinese Version	60
Appendix B. Company Profiles	63
B.1 Alpha Software Company	63
B.2 Beta Software Company	63
B.3 Gemma Software Company	63
B.4 Delta Software Company	64
Appendix C. PDD Notation	65
Appendix D. PDD Documentation	67

1. INTRODUCTION

This chapter introduces the background of the research, the problems, the research objectives, the research questions, the expected contributions, and the structure of this thesis.

1.1 Research Background

Classical software development methods like Waterfall software development methods were unable to cope with complex business environments, project uncertainty, and increasing customer demands on product specification flexibility (Zhang & Shen, 2010). However, Chen and Luo (2011) found that it would be counter-productive if the agile modes were applied simply and mechanically. The social aspect of the agile development process is a critical factor in its successful application (Nerur, Mahapatra, & Mangalaraj, 2005) and this aspect is shaped by the cultural context. In China, agile development methods are not that common, due to Chinese culture, which is typically influenced by a collectivistic pattern and other traditional cultural factors like the need for affiliation and modesty (Ramesh, Cao, Kim, Mohan, & James, 2017). Regarding the individualistic-collectivistic spectrum of Triandis proposed in 1995, China is placed in the vertical collectivistic quadrant, which is the same as India and Korea (Bhawuk, 2001). Compared with India and Korea, which emphasise the value of the group but have structural hierarchies and place importance on social status and rank, China's traditional culture may be inclined toward hierarchies while the State advocates horizontal themes like cooperation (Triandis & Gelfand, 2012). This conflict makes Chinese people hard to strike a balance between the structural hierarchies and internal collaboration, especially when both sides are not in equal status. Besides, another conflict between the capacity of Chinese organisations and their desired benefits also can be bad for implementing agile methods in China. On the one hand, many Chinese IT-related organisations are still governed by an ingrained officialism that is content with their existing software development methods. On the other hand, as a non-English-speaking country, Chinese developers are generally not proficient in English. The Chinese book market, which should be at least half a year later than current English literature as a result of the time spent on translation, is the major source for them to get their desired knowledge ("Software Development in China," 2006). Although agile adoption in China faces barriers introduced before and remains in the early stage (S1314857, 2014), implementing agile development methods which accord with its national conditions should be favourable for software development organisations in China (Eriksson, 2015).

1.2 Problem Statement

Within the courses Method Engineering and Software Product Management as part of the Master in Business Informatics program at Utrecht University, the Netherlands, the author of this thesis conducted two interviews related to ASD. The first one was conducted on March 13th of 2017, and the interviewee was Xu Dong (personal communication, 2017), a game developer in Shanghai, China. The second one was conducted in May 2017. The interviewee was named Changjian Xu (personal communication, 2017), who was a product manager of

Tencent¹. The purpose of the interviews was to examine the workflows and related artefacts based on the agile methods they adopted, to assess the product management maturity of the companies. Based on the results of the interviews, it was found that they preferred Waterfall software development methods instead of agile ones. The reason why they stated this is that Chinese ASD is too agile from their point of view. *“In iterative processes, members in the agile team are forced to work for extra hours without getting paid. The quality of the product cannot be assured, and the atmosphere of the team is badly-tempered.”* The Chinese product manager, Changjian Xu, stated that it did save much time during the process of development by using a Scrum software development framework. However, the development teams in his company spent more time fixing bugs which were found during the testing process than before. He insisted that it was caused by less communication. Besides, he said the quality of previous projects which applied Waterfall software development methods was higher. This idea was also explained by Fredheim (2013) in Quora as an answer to the question when the application of agile development is a bad idea: If the nature of the product is heavily sequential; if the organisation does not really want agile development; if the time budget and required resources are not allowed.

As of 2017, China’s software industry had a revenue of 4.9 trillion Yuan (\$775 billion), with 16% year-on-year growth and more than 5.3 million people working in the industry (“2016 年中国软件行业发展现状分析及行业竞争现状分析,” 2017). Regarding one of the systematic reviews of Mitchell and Seaman (2009), it increased 50% in code productivity using XP compared to traditional methods, and the quality increased 65% in pre-release and 35% in post-release using XP. It is believed that properly applying ASD methods can boost the efficiency of software development. This research will contribute to the application and promotion of agile development methods in China by providing guidelines, considering its cultural context. The opinions towards implementing agile methods in a Chinese cultural context will be collected from Chinese SPOs, the results of which will be beneficial to companies in the sector.

1.3 Research Objectives

According to Wieringa (2014), a goal hierarchy is shown in [Figure 1-1](#). The goals on the left side concern improvement of the real world, while those on the right concern our beliefs about the world.

¹ Tencent Holdings Limited (Chinese: 腾讯控股有限公司; SEHK: 700) is a Chinese multinational investment holding conglomerate whose subsidiaries specialize in various Internet-related services and products, entertainment, AI and technology both in China and globally. See: <https://www.tencent.com/en-us/index.html>

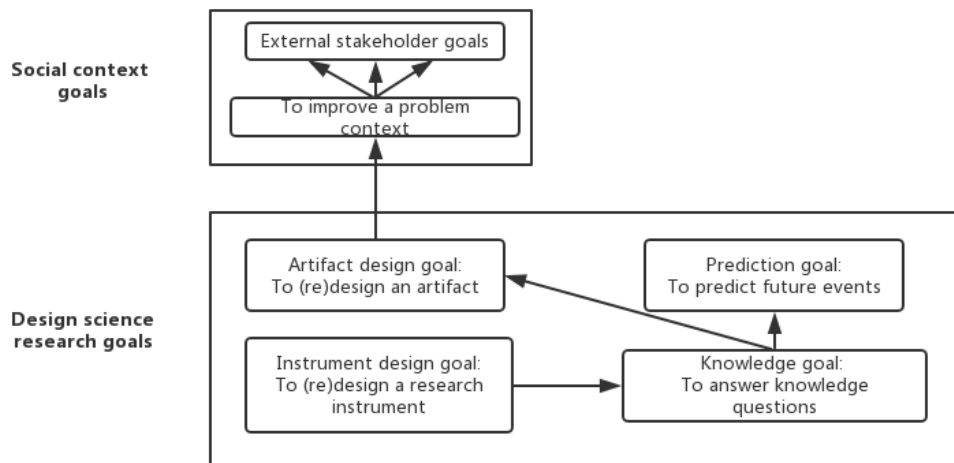


Figure 1-1: Goal structure of a design science research project

Regarding the goal hierarchy, the objectives are defined as follows: the research objective is a knowledge goal, to identify the relationship between agile theories and Chinese culture. By identifying the relationships, impeding cultural factors are scoped, and therefore Chinese IT-related organisations can avoid them.

1.4 Research Questions

To fulfill the research goals, we design the research questions as follows.

The main research question (MRQ) is defined as: “How do Chinese cultural factors influence the adoption of ASD approaches within Chinese SPOs?”

Regarding Wieringa (2014), our research question is an empirical knowledge question, which needs data in the world to settle them. In contrast, analytical knowledge question is answered by conceptual analysis, which does not require the data from the world. One way to classify the empirical knowledge question is by the knowledge goal. Some are descriptive questions, which are simply journalistic questions focusing on the description of what happened. However, our research also intends to know “why”. In this way, our research question should be classified as an explanatory question, asking why the relationship between Chinese culture and ASD theories are formed.

To answer the MRQ, two sub-questions (SQs) were formulated. To investigate the relationship between Chinese cultural factors and ASD theories, SQ1 aimed to provide an answer to the relationship we intend to investigate.

SQ1: What are the beneficial/impeding cultural factors for adopting ASD approaches?

By understanding the relationship between Chinese cultural factors and ASD theories, SQ2 aimed at discovering the value of the scoped relationship in Chinese SPOs.

SQ2: How can the identified relationship help Chinese SPOs better develop their products?

1.5 Research Contributions

In this way, we desire to explore the relationship between Chinese culture and ASD theories, analyse the status of Chinese software development, and improve existing agile methods by taking its national cultural factors into consideration, to help Chinese practitioners better know the potential cultural barriers when they adopt ASD methods.

1.6 Research Outline

After this chapter, Chapter 2 will introduce the research method adopted in our project. Chapter 3 will firstly elaborate the related background knowledge of the research and state analyse the research gap between our project and existing researches. Chapter 4 arrives at the assumed relationship between the Chinese cultural factors and ASD theories. Chapter 5 depicts the results of the semi-structured interviews, by introducing their development methods, supporting infrastructure, and opinions towards the impact of Chinese culture. Chapter 6 summarises the findings in Chapter 5 and lists the differences between the four involved Chinese SPOs. Chapter 7 compares the hypotheses made in Chapter 4 and the actual findings in Chapter 5. This is followed by the discussion of the research in Chapter 8. Conclusions and future work are presented in Chapter 9 lastly.

2. RESEARCH METHOD

This chapter depicts the research methods used in this project. Design science is followed as the research framework. Literature review, interviews, case studies, and situational method engineering approaches are chosen to support the research. Based on the design science framework, the research project is divided into several research tasks, and they will be explained in this chapter.

2.1 The Design of Empirical Cycle

According to Wieringa (2014), two activities are iterated over: designing an artefact that improves something for stakeholders and empirically investigating the performance of an artefact in a context. Wieringa treats design as problem-solving, while the two major problem-solving cycles are the design cycle and the empirical cycle. The former one, as part of engineering cycle, is composed of three tasks, involving problem investigation, treatment design, and treatment validation. The results of the treatment validation should be transferred to the real world. The engineering cycle solves a problem with the structure shown in [Figure 2-1](#). The latter one, empirical cycle, is a rational way to answer scientific knowledge questions and is organised as a checklist of issues to decide when researchers prepare to set up their researches and want to analyse the data produced in the research. Wieringa divides the research problems into design problems and knowledge questions. Design problems are proposed to call for a change in the real world, and an analysis of actual or hypothetical stakeholder goals is necessary. For design problems, it is required to follow the design cycle. For knowledge questions, the desired knowledge should be investigated and validated, and the empirical cycle is to be followed.

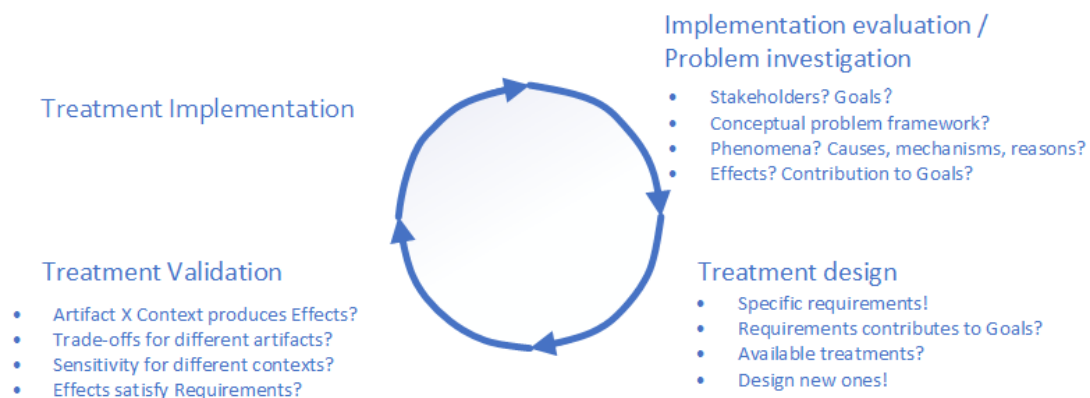


Figure 2-1: The engineering cycle of design science (Wieringa, 2014)

Our main research question, “How do Chinese cultural factors influence the adoption of ASD approaches within Chinese SPOs”, is corresponded with a knowledge question. Here we adopt the empirical cycle as the guideline to answer the scientific knowledge question. The empirical cycle has the structure of a rational decision cycle, which is similar to the engineering cycle, which we can see from Figure 2-2:

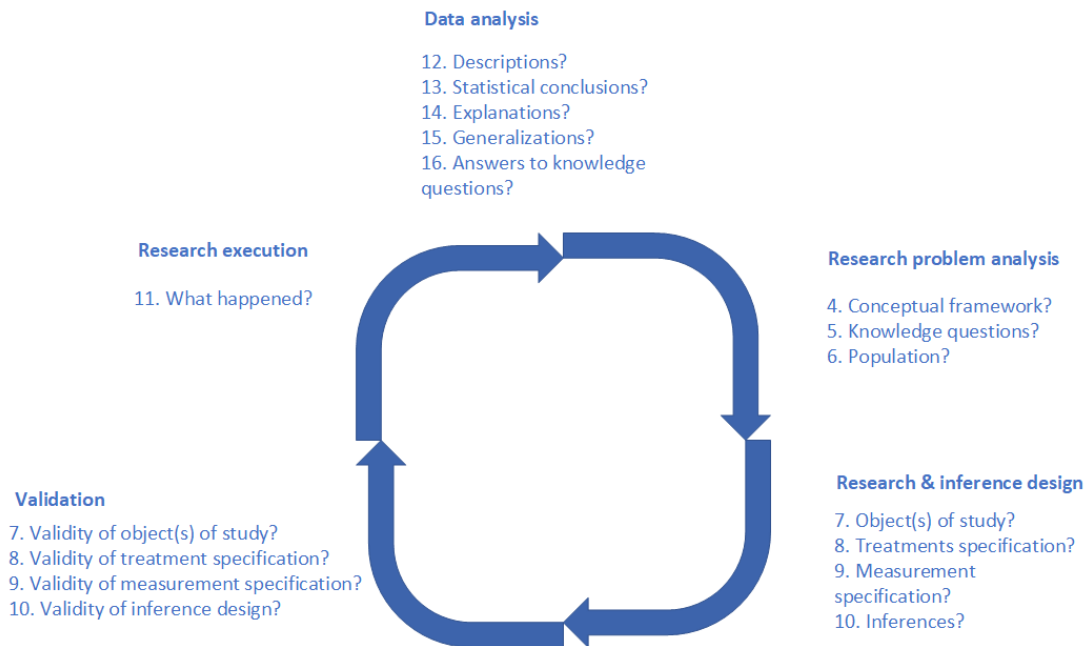


Figure 2-2: The empirical cycle

Wieringa (2014) structured the empirical cycle as a checklist of issues which can be the guideline of the research design. Here we illustrate our research based on the checklist in five steps:

1. Research problem analysis: the knowledge question, i.e. our research question, has already been introduced in [Section 1.4](#), which is to explore the relationship between Chinese culture and agile theories.
2. Research & inference design: this step tackles the problems of what we are going to do to solve the knowledge question and how we are going to conclude from the data collected during the research. For our research, we use literature review to build our theoretical propositions. From these propositions, hypotheses are then derived and tested by conducting semi-structured interviews. Our conclusions are drawn regarding the results of the hypotheses verification.
3. Validation: here we are supposed to check if the research design and inferences can match each other and if the knowledge question can be covered sufficiently.
4. Research execution: the semi-structured interviews are executed.
5. Data analysis: the results of the semi-structured interviews are analysed based on the inference design, and conclusions which interpret the relationship between Chinese culture and agile theories are drawn.

2.2 Research Approaches

Two basic research approaches are used in academic research: inductive approach and deductive approach. For deductive approach, it is concerned with “*developing a hypothesis based on existing theories, and then coming up with a research strategy to test the hypothesis*” (Wilson, 2014). [Figure 2-3](#) has already illustrated the deductive research approach of Bryman and Bell (2015). Besides the guidelines of the empirical cycle, the deductive research approach can also help us design the research. The common process of deductive approach starts from

the deducing hypothesis from a proposed theory. The hypothesis proposes relationships between two variables. By testing the hypothesis with the application of chosen research methods, we get the outcome of the test and confirm or reject the hypothesis. In the end, the theory proposed is supposed to be improved when the hypothesis is rejected.

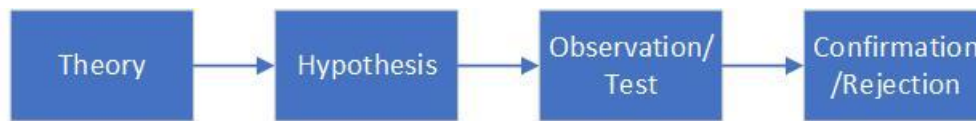


Figure 2-2: Deductive research approach (Bryman & Bell, 2015)

According to Bryman and Bell (2015), at the beginning of the deductive research approach, it is very important to build up a broad and solid literature review, which contributes to the full understanding of theoretical considerations related to the research topic. Our research has started with literature review in agile theory field and Chinese culture field. Agile theories are decomposed into agile values and agile principles proposed in Agile Manifesto, while Chinese culture is analysed based on the data from the Hofstede model. Based on our research questions, the theoretical framework and the related hypothesis should be built. Then, the empirical data should be collected regarding qualitative interviews, which should connect agile theories with Chinese culture, to interpret the relationship between the two objects from the feedback of the interviewees. Later, the raw data should be organised by the agile practices and be compared with the hypothesis. In the end, our designed artefact is to be improved based on the data analysis, and investigate potential cultural factors which have not been included in our primary hypothesis.

2.2.1 Literature Review

As we desire to investigate the relationship between Chinese culture and ASD theories, it is of vital importance to overview what has been studied previously. To see what has already been done in prior researches, a database searching was performed. We used Web of Science, Google Scholar, and CNKI (China National Knowledge Infrastructure)² as our search engines. The search focused on contemporary (Chinese) software development methodologies. After that, the articles were categorised into two main scenarios: Agile methodologies and Chinese culture. The year of publication was limited from 1978 to 2017, considering Chinese economic reform³, which happened in 1978 and resulted in immense changes in Chinese society.

Here we define the keywords as Software development method, Waterfall, Agile, agile adoption, software product management, Chinese culture, software development method + Chinese culture, Chinese software industry.

Besides, the following requirements were also considered: It should be a journal article, conference proceedings paper, book, or a thesis; the paper should be written in English or Chinese; the paper can be accessed online.

Next, the scope of the literature was still expanded by iterating backward and forward snowballing (Webster & Watson, 2002). Backward snowballing is realised by scoping

² CNKI (China National Knowledge Infrastructure, 中国知网) is a key national information construction project under the lead of Tsinghua University. See: <https://en.wikipedia.org/wiki/CNKI>

³ The Chinese economic reform (simplified Chinese: 改革开放) refers to the program of economic reforms termed "Socialism with Chinese characteristics" in the People's Republic of China (PRC). See: https://en.wikipedia.org/wiki/Chinese_economic_reform

meaningful concepts and reviewing the reference lists from the identified literature to find previous related articles. On the other hand, forward snowballing is to find elaborated discussion by finding new articles which have cited those old articles. The search strategy is used to ensure that we have covered as much relevant literature as possible.

2.2.2 Semi-Structured Interview

Interview is an interesting way to collect data in addition to literature review. Semi-structured interviews are simply conversations in which you know what you want to find out about – and so have a set of questions to ask and a good idea of what topics will be covered – but the conversation is free to vary, and is likely to change substantially between participants (Fylan, 2005). Compared to structured interview, the design of the questions of semi-structured interview is flexible, and the order of the questions can be altered. However, the questions are predetermined, which makes it also different from unstructured interview.

Fylan (2005) introduces the characteristics of semi-structured interview as follows: on the one hand, the questions are simple, and their order can be altered. On the other hand, the questions are open, and the conversations can be stimulated from some angles. The complexity of the questions is inversely proportional to the degree of structure of the question.

Semi-structured interview is selected as our research needs the respondents' interpretation towards agile adoption, regarding their own prior experience. Given the pre-defined keywords and narratives, we wish more valuable information could be excavated from the interviewees.

Interview protocols are improved step by step, which means questions would be changed based on findings from earlier interviews. Before the interviews, interview protocols were sent to the interviewees to allow them better prepare and organise their opinions. Besides, if there is any information make them sensitive or uncomfortable, it is convenient to update the protocol in advance.

There are four companies which participated in our research, and the information will be introduced in [Chapter 5](#). The interviews were performed between January 2018 and May 2018. The interviews have covered the topics: agile methodologies, Chinese culture, organisational culture, organisational hierarchy, customer collaboration, agile tools, and software product management. Each interview took around 60 minutes. The interview protocol is provided in [Appendix A](#).

2.2.3 Situational Method Engineering

With the technology continuing to diversify, an increasing number of methods for developing information systems are developed. Nevertheless, methods are often not quite suited to the physical truth of the projects. Situational method engineering was therefore designed by Van de Weerd and Brinkkemper in 1996, to better deliver customised methods for information system (IS) evaluation. Method engineering is defined by Brinkkemper as: *“the engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems”* (Brinkkemper, 1996). The approach is supported by a meta-modelling technique, based on UML activity and class diagrams (van de Weerd & Brinkkemper, 2008). According to Brinkkemper (1996), the definitions of method and technique should be classified. A method is an approach to perform a system development project, consisting of

directions and rules, structured systematically in development activities with corresponding development products. By contrast, a technique is a procedure, possibly with a prescribed notation, to perform a development activity (Brinkkemper, 1996). We regard method as systems consisting of rules to realise certain goals, while the technique is the realisation of a method. In our project, general agile software development methods are clarified as methods, and specific agile methods like Scrum and XP are classified into techniques.

To analyse existing software development methods and design our improved method, we use the situational method engineering. Based on the ideas of Van de Weerd and Brinkkemper (2008), we develop a situational method based on these steps:

1. Analyze project situation and identify needs;
2. Select candidate methods which meet one or more aspects of the identified needs;
3. Analyze candidate methods and store relevant method fragments in a method base;
4. Select useful method fragments and assemble them in a situational method by using route-map configuration to obtain situational methods.

The latter two steps are supported by a meta-modelling technique named process-deliverable diagram (PDD), the left side of which is the process view, while the right of which is the deliverable view. The detailed description of the PDD notations is attached in [Appendix C](#).

2.3 Research Validity

Rigour in IS research is still one of the critical scientific issues (Straub, Boudreau, & Gefen, 2004). The scientific basis of the research is threatened without solid validation. We extract several validities from the theories proposed by Straub et al. (2004) which are crucial for research design: construct validity, reliability, and external validity.

Construct validity is to check whether the measures we choose fit together in such a way to capture the essence of the construct (Straub et al., 2004). The source of our research data is from literature study, interview, and case study. Besides, domain experts are invited to validate our solution. The more strategies used to demonstrate the validity of a test, the more confidence test users have in the construct validity of that test, but only if the evidence provided by those strategies is convincing (Brown, 2000).

While construct validity is an issue of measurement between constructs, reliability is an issue of measurement within a construct (Straub et al., 2004). It is important to ensure that later researchers can replicate the same results if they follow the procedure described by prior researchers. We have presented our interview protocol and search strategy of the literature review to make sure the data can be gathered reliably.

For external validity, the focus group of our research is targeted towards Chinese IT-related industry. It is hard to say that our research can be generalised. However, the research can be further extended to other developing countries which are influenced by Confucianism.

2.4 Research Execution

The research execution is illustrated in [Table 2-1](#), including its milestones and deliverables. The research is supposed to be finished in eight months.

Table 2-1: Project planning

Date	Task
8/12/2017	Short Proposal
9/12/2017	Start of Literature Review
10/2/2018	End of Long Proposal
6/3/2018	1st Colloquium Presentation
1/5/2018	Validated Customized Software Development Method
1/6/2018	Deadline exploratory Case Studies
18/6/2018	2nd Colloquium Presentation
30/6/2018	Draft Thesis Report
15/7/2018	Final Thesis Report
16/7/2018	Scientific Paper
31/7/2018	Thesis Defense

Given the Master thesis guideline for MBI students, the research is separated into two parts:

1. First phase: We are required to define the research method for the thesis and conduct a literature study in the field of the study. We are desired to develop a research plan and understand the relevant literature. This part is ended up with the first colloquium presentation.
2. Second phase: The research is executed in this phase based on the research plan proposed in the first phase. Here we are supposed to deliver our thesis report, and a scientific paper if possible. This phase is ended up with a second colloquium presentation and a thesis defence.

In [Figure 2-3](#), the activities and deliverables of the research are presented in a PDD. The left side of the diagram shows activities, and the right side shows concepts. The PDD is a meta-modelling technique proposed by Van de Weerd and Brinkkemper (2008), to present the process and deliverable of a method or a technique.

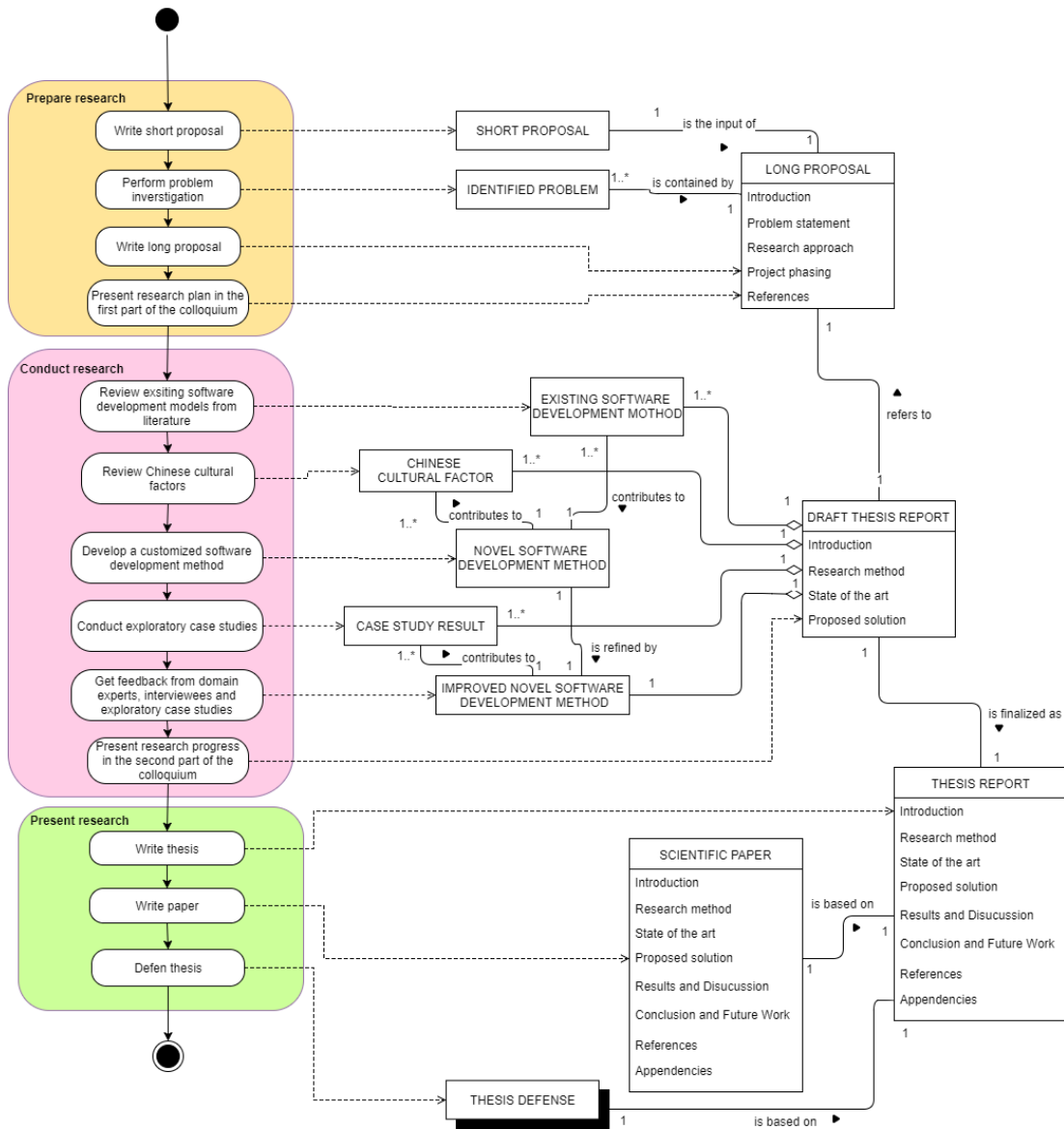


Figure 2-3: Project phasing in PDD

3. LITERATURE STUDY

This chapter presents the results of the literature study, which is the research background of this project. During the problem investigation of our project, we follow the literature review protocol provided in [Section 2.2.1](#), and a semantic net was constructed centred on the Chinese software development. In [Figure 3-1](#), key concepts which will be illustrated in this section have been summarised, and their relations have been pointed out.

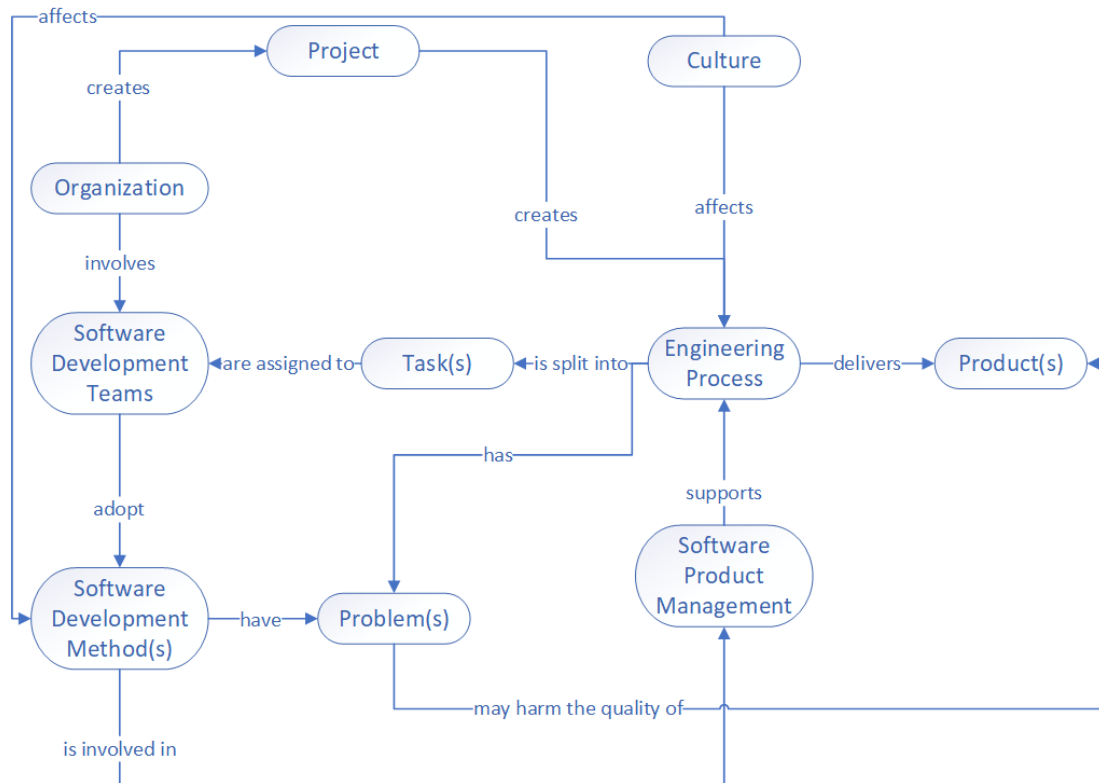


Figure 3-1: Semantic net of key concepts found during the literature review

The main concepts can be organised in a narrative to better summarise the results of the literature study: Chinese IT-related companies create projects to deal with business, where software development teams are involved. Engineering process is generated, and the process splits the business into tasks, which are assigned to development teams. During the development, problems caused by the conflicts between Chinese and Western culture are created in the design of their software development methods and engineering process, which would harm the quality of the products. By using methodologies of software product management, the product is to be delivered. Each company has its software development method, the candidates of which can be waterfall or agile software development methods. Besides, some hybrid software development methods like bimodal IT are adopted by some companies. The fact was found in an interview with Jiawei Zhu (personal communication, 2018), an experienced product director who has worked in the area for 13 years. For the engineering process, it is influenced by Chinese cultural factors, as the status quo of China is not the same as America, where the agile theories were brought up.

3.1 Agile Software Development

The traditional software development methods are said to advocate extensive planning, codified processes, and rigorous reuse to make development an efficient and predictable activity (Boehm & Turner, 2005).

In contrast, agile processes address the challenge of an unpredictable world by relying on “*people and their creativity rather than on processes*” (Nerur et al., 2005). Seventeen software developers proposed the Manifesto for Agile Software Development. Its values (Beedle et al., 2001a) are listed as below: *Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan.*

These four values imply the core idea of quick development and increasing customers’ satisfaction. The first two values (individuals and interactions & working software) correspond to quickly development, while the latter two (customer collaboration & responding to change) illustrate the attitude towards customer satisfaction. Additionally, there are also twelve principles (Beedle et al., 2001b) in ASD:

1. *Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.*
2. *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*
3. *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*
4. *Business people and developers must work together daily throughout the project.*
5. *Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.*
6. *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*
7. *Working software is the primary measure of progress.*
8. *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*
9. *Continuous attention to technical excellence and good design enhances agility.*
10. *Simplicity - the art of maximising the amount of work not done—is essential.*
11. *The best architectures, requirements, and designs emerge from self-organising teams.*
12. *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.*

Also, agile practices were studied by Ramesh, Cao, and Baskerville (2010). The data was collected from sixteen software organisations in the USA to explain the difference between traditional requirements engineering and agile requirements engineering. In the research, six agile requirements engineering practices were identified: Face-to-face communication, iterative requirements engineering, using extreme practices in requirements prioritisation,

using constant planning to manage requirements changes, prototyping, and using review meetings with acceptance tests.

Based on Chinese cultural factors to be analysed, these ASD theories will be verified to examine the conflict points.

Ericksson, Lyytinen and Siau (2005) defined agility as follows: Agility means to strip away as much of the heaviness, commonly associated with the traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like. Williams and Cockburn (2003) stated that agile development is *“about feedback and change”*, that agile methodologies are developed to *“embrace, rather than reject, higher rates of change”*.

In spite of the benefits agile development methods offered, Misra et al. (2012) organized several negative comments towards agile development methods of some authors, while the criticism mentioned can be debatable: For the human factor, it is hard to get the appropriate people involved; its efficiency can be high only when senior development team members are involved. For development process, there is limited support for building reusable artefacts and developing large and complex software. For cultural aspect, it requires significant cultural change for adoption in projects and may be hard to attain contractual negotiations. For its restrictions on execution, it will work only when the performance requirements are explicitly stated at the outset of the projects, and suitable test plans are planned; it does not work for development projects for safety-critical and reliable systems. Besides, in the Agile Manifesto 10th Anniversary, Philippe Kruchten (2011) listed the elephants in the room (a metaphor for the behaviour of people who deliberately ignore an impending issue) for ASD. Kruchten classified the elephants into several groups. One of the groups was failures and limitations of agile practices & context, the main cause of which was that many of the key players in the community had a direct financial interest in selling something agile so that negative news would be amplified. The other two elephants were politics and culture.

To better illustrate an agile software development method, we present Scrum (one of the agile software development methods) in a PDD in [Figure 3-2](#).

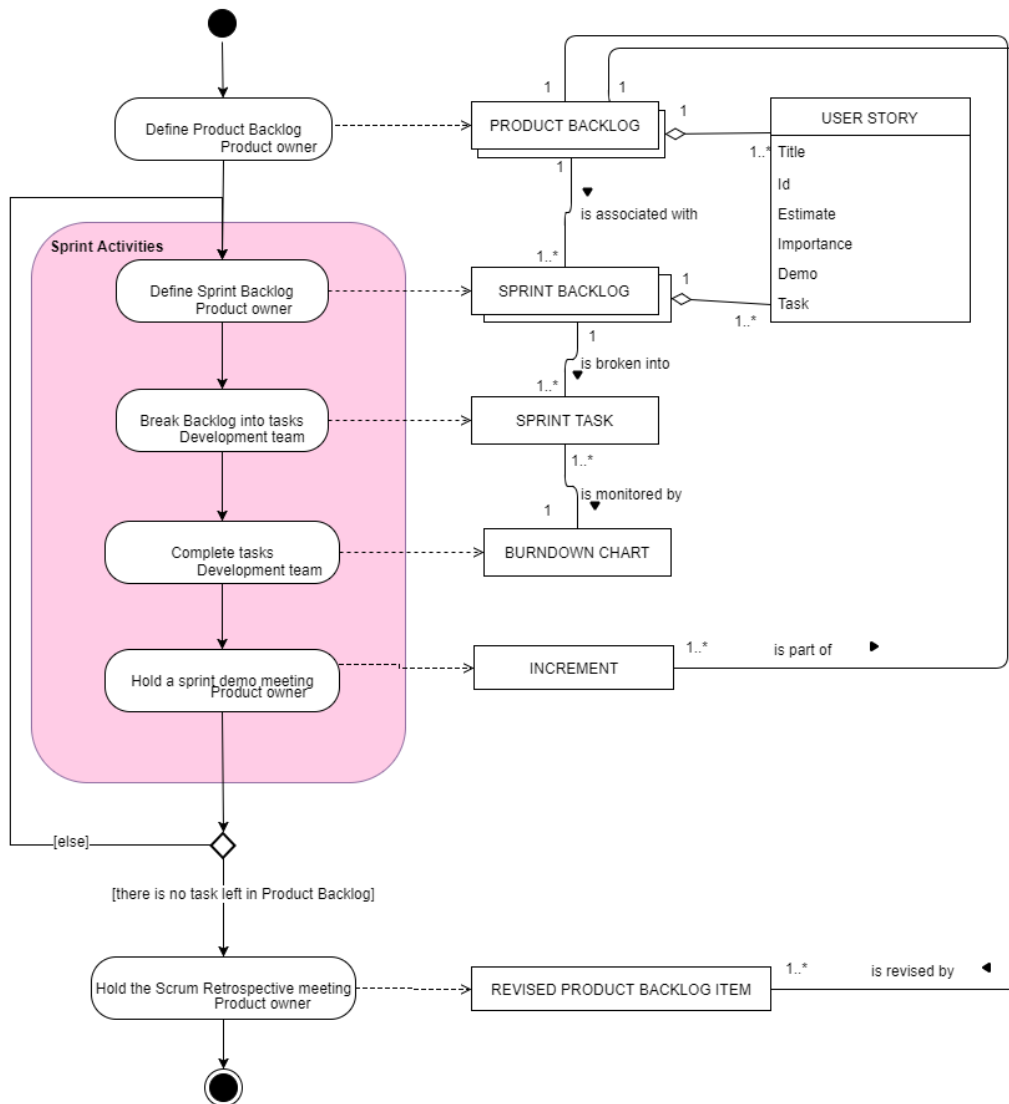


Figure 3-2: A PDD of Scrum

According to The Scrum Master Training Manual (Rad and Turley, 2013), they provided a basic idea of how a Scrum project works, which is shown in Figure 2. Here the product owner is responsible for defining product backlog and select user stories from product backlog to form a sprint backlog in an iteration, while team members are supposed to break the sprint backlog into tasks to fulfill them. At the end of each sprint, working software is to be delivered, and stories which are fulfilled should be marked as Done in the product backlog. At the end of the Scrum, there is also an ongoing activity named Product Backlog grooming, where product owner should prioritise the items in the product backlog. During the iterative processes, Scrum master is assigned to ensure that the Scrum framework is followed entirely.

In general, ASD theories are one of the background knowledge in our research. As a hot topic, agile can create and respond to changes in an environment full of turbulence and uncertainty (Highsmith, 2002). By having an insight into the agile values and principles, we can deduce the national cultures which correspond to the agile theories. On the other hand, one of the specific ASD method, Scrum, has been introduced. Scrum can be used as a standard method when we introduce the customised methods in [Chapter 5](#), to analyse their differences and explore the reasons.

3.2 Hofstede's Cultural Dimensions

For the global IT research, the majority of Information Systems (IS) issues have been studied from a national culture perspective (Gallupe & Tan, 1999). To analyse the national culture, there are a number of national culture models, and they can be categorized as single dimension models, multiple dimension models, and historical-social models (Morden, 1999). Based on Morden's findings, we have listed these three types of national culture models and their dimensions in [Table 3-1](#).

Table 3-1: Models of national culture (Morden, 1999)

Models	Cultural Dimensions
Single dimension models	<ol style="list-style-type: none"> 1. High and low context cultures (Hall & Hall, 1989) 2. Monochronic and polychronic cultures (Lewis, 1992) 3. High and low trust cultures (Fukuyama, 1995)
Multiple dimension models	<ol style="list-style-type: none"> 1. Hofstede's model (Hofstede, 2011) 2. The seven cultures of capitalism (Hampden-Turner & Trompenaars, 1993) 3. The four inter-related criteria (Lessem & Neubauer, 1994)
Historical-social models	<ol style="list-style-type: none"> 1. Euromanagement model (Bloom, Calori, & De Woot, 1994) 2. South East Asia Management Model (Chen, 2004; Cragg, 1995; Seagrave, 1995)

Muhammad and Shaiq (2011) summarised the drawbacks of the Hofstede's model like the definition problems and methodological simplicity. Despite all this, Søndergaard (1994) found 1036 quotations from Hofstede's model in journals between 1980 and 1993 from the Social Science Citation Index. Myers and Tan (2002) also examined 36 IS researches into the effects of national culture between 1983 and 2000, and 24 of them used Hofstede's work. Both prove that Hofstede's model should be the first choice when researchers study the areas related to national cultures. In this way, we use Hofstede's model as well to interpret Chinese culture.

Hofstede model describes the effects of a society's culture on the values of its members, and how these values relate to behaviour, using a structure derived from the factor analysis (Adeoye, 2014). Since Hofstede's original IBM survey, there have been several replications of his research, which confirm and complement results of the original study. It has been refined into six dimensions: Power Distance, Uncertainty Avoidance, Individualism-Collectivism, Masculinity-Femininity, Long-term Orientation, and Indulgence versus Self-restraint.

- Power Distance (PDI)

"Power Distance has been defined as the extent to which the less powerful members of organisations and institutions (like the family) accept and expect that power is distributed unequally" (Hofstede, 2011). Power distance describes the degree to which unequal distribution of power and wealth is tolerated (Jones, 2007). Hierarchy means existential inequality in society with large power distance, and people accept the hierarchical order by nature. For a society with small power distance, people are accustomed to being treated equally, and subordinates expect to be consulted instead of being told what to do.

- Uncertainty Avoidance (UAI)

“Uncertainty deals with a society's tolerance for ambiguity. It indicates to what extent a culture programs its members to feel either uncomfortable or comfortable in unstructured situations” (Hofstede, 2011). Also, uncertainty avoidance is the way people deal with future, whether they have inherent control or whether situations are out of their control (Jones, 2007). People with strong uncertainty avoidance may stay in the jobs even if they dislike the jobs, and they score lower on subjective health and well-being, while with higher stress. In contrast, people with weak uncertainty avoidance can change their job without problems, and they feel comfortable with ambiguity and chaos.

- Individualism versus Collectivism (IDV)

Individualism is *“a culture in which the ties between individuals are loose: everyone is expected to look after him/herself and his/her immediate family”* (Hofstede, 2011). On the contrary, collectivism stands for the degree to which people in a society are integrated into groups (Hofstede, 2011). This distinction can be identified based on whether a culture expects people to be independent at a very early age (Jones, 2007). According to Hofstede (2011), people in individualist societies are born into extended families which raise them up in exchange for loyalty, and harmony should always be maintained. For collectivist societies, everyone is supposed to take care of their individual belongings and to speak one's mind is healthy.

- Masculinity versus Femininity (MAS)

Masculinity depicts the degree to which masculine traits like authority, assertiveness, performance and success are preferred to female characteristics like personal relationships, quality of life, service and welfare (Jones, 2007). In masculine societies, people should be assertive and ambitious, and work prevails over family. By contrast, people in feminine societies are generally modest and caring, and they try to strike a balance between family and work.

- Long-term versus Short-term Orientation (LTO)

This dimension was first identified in a survey among students in 23 countries around the world, using a questionnaire designed by Chinese scholars (The Chinese Culture Connection, 1987). As all countries with a history of Confucianism scored near one pole which could be associated with hard work, the study's first author Michael Harris Bond labelled the dimension Confucian Work Dynamism (Hofstede, 2011). Values found in long-term orientation were perseverance, and ordering relationships ranked by status; values at short-term orientation were reciprocating social obligations, respect for tradition, protecting one's 'face', and personal steadiness and stability. People in long-term orientation countries attribute success to effort and failure to lack of effort and are trying to learn from other countries. However, people in short-term orientation countries attribute success and failure to luck, and they are supposed to be proud of their counties.

- Indulgence versus Restraint (IND)

This dimension was added in 2010, which uses Minkov's label Indulgence versus Restraint (2007). It is complementary to Long-term versus Short-term Orientation. Indulgence stands for a society that allows relatively free gratification of basic and natural human desires related to enjoying life and having fun. Restraint stands for a society that controls gratification of needs and regulates it using strict social norms (Hofstede, 2011). People in restraint societies hold the perception of helplessness: *“what happens to me is not my own doing”*, while people in indulgent societies emphasise personal life control. Besides, in restraint societies, freedom

of speech is not a primary concern, and leisure is not valued. For indulgent societies, the situations are to the opposite.

As we stated at the beginning of this section, Hofstede's model is most quoted in many publications (Muhammad & Shaiq, 2011). As a reference model, we simply use the six dimensions to interpret Chinese culture.

3.3 Chinese Culture

Culture offers both local and global challenges to software teams as they collaborate to understand requirements, build systems, and deliver products (Fraser et al., 2008). It is of vital importance to overcome potential cultural barriers to deliver the desired product on time. For China, its civilisation has lasted for more than 10,000 years. As one of the oldest homogeneous civilisations in the world, the Chinese culture has been influenced by its local culture and the experience that it was conquered by external groups of people (Eriksson, 2015). For its native culture, the most influencing one is Confucianism, which is proposed by Confucius who lived around 500 BC. Confucianism is characterised by human relations, the importance of an individual's consciousness of his or her place in the society, and the respect for the hierarchy that must exist for the society in harmony. This virtue of hierarchy and the consequent unequal relationships as a necessary basis for stability in society has continued to be an essential element of the Chinese culture for more than 2000 years, expressing itself as the obligation of a subject who has a low position in the hierarchy to respect and serve a higher positioned subject. Three major attributes can be collected from the paper of Ying Fan (2000): Collectivism, guanxi network, and moderation.

For moderation, it is one of the core value of Confucianism. It advocates people to follow the middle way instead of being outstanding. In a seminal book of Confucianism, "Zhong Yong" (中庸), the internal states of equilibrium and harmony were emphasized. In summary, in Confucianism, behavioural ideals require the individual not only to control his or her desires to comply with the pivotal virtue of benevolence, but also to maintain a peaceful state of mind without overly positive or negative emotions in their mental state (Lee, Lin, Huang, & Fredrickson, 2013).

The Chinese culture identifies collectivism as a central ideology, where individuals would be glad to sacrifice the society and groups they belong to, rather than focusing on their benefits (Richard D Lewis, 2006). It is a common phenomenon that members of a development team would volunteer to work for extra hours without payment, to meet the emergent requirements of the customer.

Guanxi is the Chinese translation of relationship. It is more linked with resources and benefits, rather than a simple definition of the relationship. For example, in the conflict resolution model of Kwang-Kuo Hwang (1997), a typical example can be described as: When petitioner (P) requests resource allocator (RA) to distribute a certain kind of resource under RA's control in such a way as to favor P, the first question RA has to consider is "*what is the guanxi between us?*"

The national culture of China also influences its characteristics and organisational culture.

For its characteristics, its basic identity is obtained from the Chinese tradition and culture in which their beliefs, attitudes and behaviours are shaped and guided (Fan, 2000). According to Confucianism, hierarchy and the consequent unequal relationships are emphasised. In this

way, loyalty is supreme in a collectivist culture. Besides, Mientze, which is also considered as Lian in Chinese, is related to a person’s moral character and personal integrity (Gao, 1998). It can be a taboo if you criticise someone in public, which leads to the deterioration of the relationship as people with strong self-esteem do not want to lose face.

For organisational culture, it is mainly formed by the practices it implements (Hofstede, 1991). The managerial practices in Chinese organisations are greatly impacted by the Chinese cultural values, which are rooted in the very influential Confucian ideologies that incorporated the need for harmony by avoiding conflicts, the Chinese concept of face and the respect for age and hierarchy, among many others (Lin, 2008). When agile development methods are implemented inside an organisation, national and organisational culture should be considered, which would possibly offer a deep insight in the specific part of the culture that impacts specific part of agile and agile adoption (Eriksson, 2015). Fan (2000) also concluded that culture is important and attractive to explain the differences in management behaviour or organisations system.

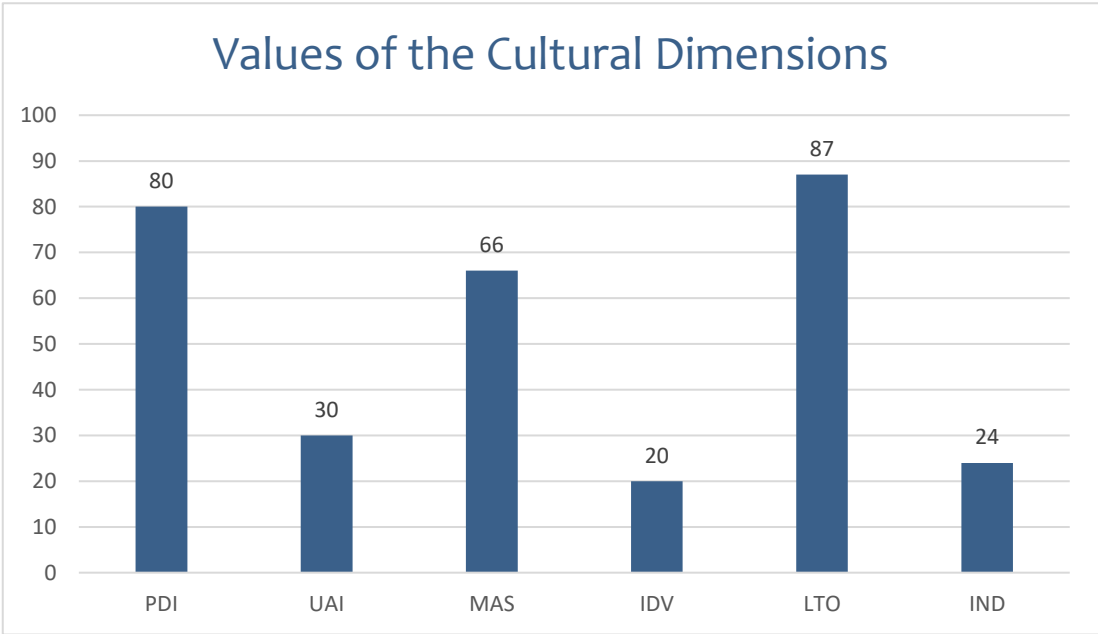


Figure 3-3: The results of Hofstede cultural dimensions of China

As can be seen from [Figure 3-3](#), the values for the six dimensions of China have already been provided by Hofstede’s work. China scores 80 in PDI, which means that inequalities and hierarchies are well accepted in the society, and people are accustomed to following the orders given by their superiors. For IDV, China is a collectivist country, which means that individuals are integrated into groups. The guanxi network makes personal relationships prevail over the task. As a masculine society, Chinese people lay more emphasis on success, sacrificing family and leisure time to work. And Chinese students value their scores in the exams very much, which is regarded as the criteria for success or not. China’s UAI is 30, which is relatively low. With the reform and opening up, China holds an open mind towards changing trends, and they are glad to learn from other outstanding countries. Besides, the Chinese language is full of ambiguity, which also made them flexible and adaptable. For LTO, China scores 87. As a pragmatic culture, people know truth depends on the situation, context and time. President Xi Jinping pledges sustainable development to build a “beautiful China” at the 19th National Congress of the Chinese Communist Party. Influenced by the Confucian thought, people are required to restrain themselves and bear hardships to attain higher levels.

By comparing LTO with IND, we find that both of them share similar meaning in Chinese culture: to stay a long-term orientation, it is of necessity to be self-disciplined. In China, it is common that people in first-tier cities like Beijing and Shanghai spend less time and money on entertainment, to save money to buy a decent apartment for stable residence. Long-term orientation is tightly correlated with buying property and providing their children with better growth environment. That is to say: indulgent life could hardly support the universal desired life in China. Therefore, we combine IND with LTO from now on.

Chinese culture is deeply influenced by Confucianism, while also becomes more inclusive after China's reform and opening up. Faced with the diverse culture, we use the Hofstede's model to understand it via six cultural dimensions. Besides, the guanxi network will also be connected with the customer collaboration in [Section 7.5](#).

3.4 Bimodal IT

To overcome the challenge which has been coined as digital transformation (Horlach & Drews, 2016), many organisations start to establish a new "digital IT". Gartner⁴, a leading IT research and advisory firm, formally introduced the concept of Bimodal IT ("What is Bimodal IT?", 2014). Bimodal IT is defined as "a parallel workflow of managing two separate styles of work – one focused on the predictability and the other on exploration". Ingmar, Bradley, and Benlian (2017) stated that the function of Bimodal IT resolves the conflicting goals of stability and experimentation.

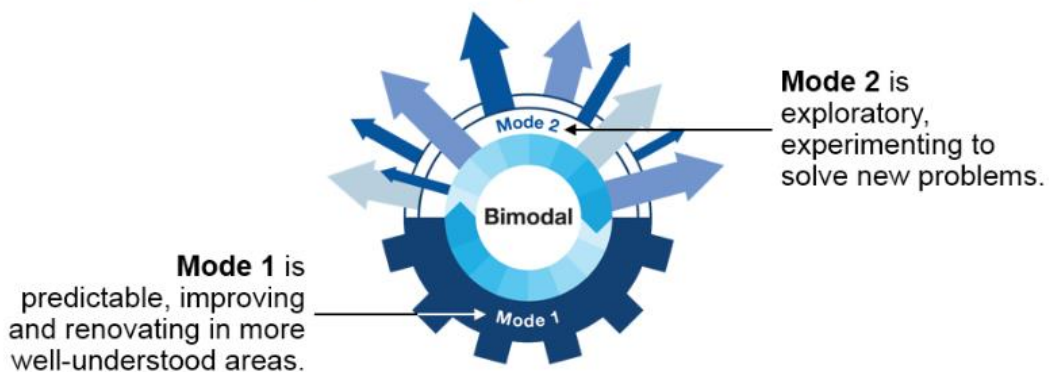


Figure 3-4: Two modes of Bimodal IT

The theory separates the organisations' IT work styles into two modes. As can be seen in [Figure 3-4](#), Mode 1 is predictable, improving and renovating in more well-understood areas, while Mode 2 is exploratory, experimenting to solve new problems. The traditional mode is responsible for the performance of the software and usually uses Waterfall-driven (sequential) approaches to managing IT projects and facilitates a risk-averse culture. The Agile mode focuses on customer experience and business outcomes, with teams often applying agile (iterative) project management methodologies (such as Scrum technologies) that target short release cycles and working on projects with less certain outcomes (Ingmar et al., 2017). The characteristics of both modes, which were summarised by Horlach & Drews (2016), are shown in [Table 3-2](#) below.

⁴ Gartner is an American research and advisory firm. See: <https://en.wikipedia.org/wiki/Gartner>

Table 3-2: Characteristics of Traditional and Agile IT

Traditional IT (mode 1, industrial / core IT)		Digital IT (mode 2, agile IT)
Stability	Goal	Agility & speed
IT-centric	Culture	Business-centric
Remote from customer	Customer proximity	Close to customer
Performance and security improvement	Trigger	Short-term market trends
Performance of services	Value	Business moments, customer branding
Security & reliability	Focus of services	Innovation
Waterfall development	Approach	Iterative, agile development
Systems of records	Applications	Systems of engagement
Slow	The speed of service delivery	Fast

By searching the comments towards Bimodal IT via the Internet, we found Bimodal IT could be a double-edged sword. According to Watts (2018), there are plenty of benefits to delineating two IT modes: Speed, innovation, agility, and reducing the use of unauthorised applications and software. Watts also supplemented that Bimodal IT is not a one-size-fits-all solution: The separation leads to a sense of distance between the organizations; the separation pursue that the two modes should not rely on each other, while the goal of mode 2 often relies on the legacy of mode 1; the separation can be confusing which makes the employees unwilling to change; the separation cannot guarantee innovation. One impressive blog found on the internet was written by Campbell (2016), whose idea can be concluded as Bimodal IT is new wine in an old bottle; Bimodal is too trivial while the real world is multi-modal; the Bimodal IT model does not even present a cogent long-term solution to the oversimplified agility-stability problem.

Bimodal IT should be an alternative solution to the conflict between stability and agility. It was once applied in the case study of Alpha Software Company ([Section 5.1](#)). During the interview with one of their product manager, we were told that the effect of Bimodal IT was not implemented successfully in their organisation. Mode 1 was satisfying while Mode 2 was disappointing. Due to the changing requirements from the operating department, members of the team had to work overtime for free to deliver the working software in required time. This phenomenon lasted half a year, and the team members all resigned as they were tired of the unsatisfying life.

3.5 Software Product Management

McDaniel and Gray (1980) introduced a new role in software development team called product manager, who is responsible for the integration of resources for effective management of individual products or groups of products.

Bekkers, van de Weerd, Spruit and Brinkkemper (2010) presents a Software Product Management (SPM) Competence Model as a comprehensive overview of all important areas of software product management ([Figure 3-5](#)). The model is developed from SPM reference

framework by De Weerd et al. (2006). The competency model depicts the interaction and information flows among stakeholders through four business functions: requirements management, release planning, product road mapping, and portfolio management. Each business function is elaborated with several focus areas which represent a coherent group of capabilities within a business function. Overall, the model explains the key competencies that should be fulfilled by stakeholders in software product management.

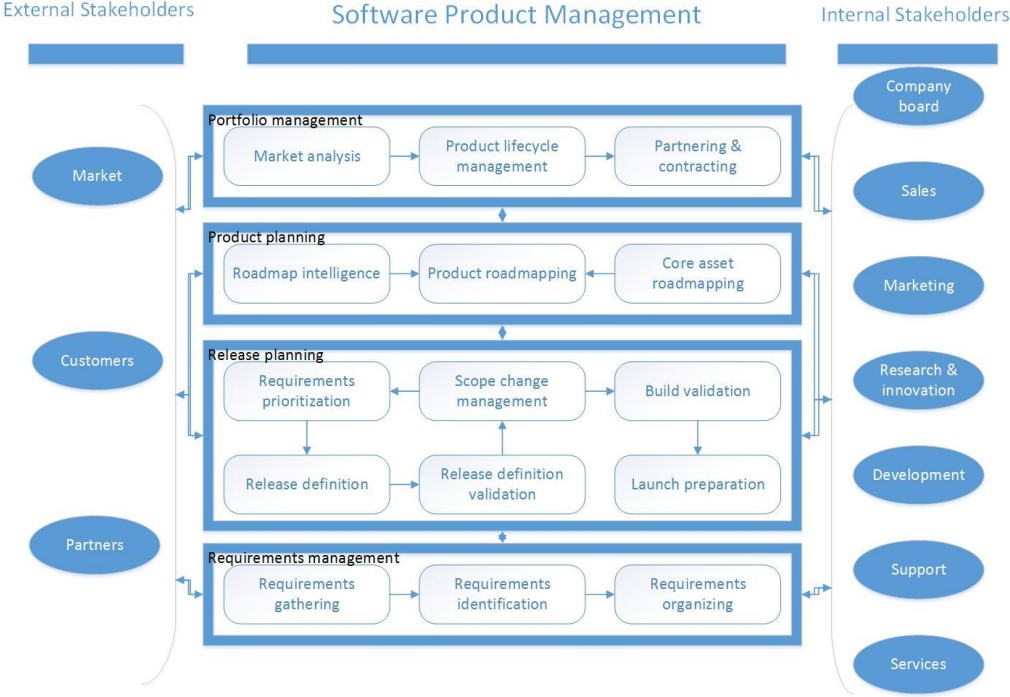


Figure 3-5: The Software Product Management Competence Model

The benefits we can gain from this model is that we can analyse existing Chinese SPOs’ software development methods regarding the business functions the model offered. Referring to the competence model, we can learn the corresponding stage of each company participating in the research. Hence, we will understand the reasons why the decision-makers customized their development methods in such ways.

3.6 Research Gap Analysis

Regarding existing research on agile implementation, we found there is a gap between agile development methods and Chinese cultural factors. Based on the research of Eriksson (2015), the communication aspect within Chinese software development organisations is a barrier in agile adoption in China, while the aspects of customer collaboration and knowledge sharing have not yet been determined. Eriksson identified three limitations of his research: One was the problem of collecting enough sample data, and another was that he found the respondents of the survey might be biased towards agile development, and the last one was that the size of the questionnaire was relatively small (18 questions), which was aimed at obtaining more respondents. However, it was restricted to analyse more answers and more data. Besides, interviews were not held in the context of the research, which leads to a loss to know the Chinese industry deeply.

On the other hand, as foreign researchers are not locals in China, a severe problem is that they cannot determine the accuracy of the survey. The face problem proposed by Gao (1998) is a typical phenomenon, the idea of which has deeply rooted in every Chinese. By saying “no” to seeking a relational partner, the Chinese constantly evaluate and explain their daily behaviour in accordance with commonly held cultural conventions. It is hard to determine if the participants of the researches are willing to tell the truth, considering the confidentiality provisions of their companies, which may lead them to produce fake answers to avoid the confidential information leaking out.

When it comes to Bimodal IT, an informal interview (personal communication, 2018) was conducted in March 2018. The interviewee was a front-end developer in a Chinese software company. In his department, there were two groups of staff, the first of which was responsible for developing the standard product (a B2C online shopping mall) of the department, and the second of which focused on solving the tasks of Party A companies offered. Due to the unclear requirements of a Party A company, the customised product was rebuilt for several times. The employees in the second group all resigned because they felt no sense of achievement and got a bad performance evaluation. It can be inferred that the interviewee thought it was a failed attempt through their tone. Objectively speaking, the size and the capacity of the company also cannot support the Bimodal IT pattern. The advice I got from the interviewee was that the common solution of the Chinese software organisations would be extracting experienced developers from the teams to form a digital group to solve a certain project, and the team will be disbanded after the project is finished.

For our part, we intend to do an internship in a Chinese organisation to better contact those software industry practitioners, to make friends with the colleagues, which contributes to the fact that they can open up to us while participating in the research. As defined by Bian and Ang (1997), a basic characteristic of *guanxi* is familiarity or intimacy: *“guanxi is not merely a relationship but a tie through which the parties exchange valued materials or sentiments”*. In this way, it is relatively easy to learn their real ideas towards agile adoption. Based on the questionnaire of Eriksson (2015), we desire to improve the existing one by extending the questionnaire with an organisational culture section and conduct interviews and case studies for the Chinese organisations to better understand their current situation of software development methods and their attitude towards agile adoption.

4. ASSUMED RELATIONSHIP BETWEEN AGILE THEORIES AND CHINESE CULTURE

By introducing the agile theories and Hofstede's cultural dimensions, assumptions of the relationship between agile theories and Chinese culture can be made. In the meantime, our main research question, "How do Chinese cultural factors influence the adoption of ASD approaches within Chinese SPOs", can be analysed.

4.1 Power Distance and Self-organizing Teams

According to Hofstede (2011), power distance is defined as the extent to which the members with less power of organisations accept and expect that power is distributed unequally. That is to say, in countries with high PDI index people would be used to an organisational structure with hierarchies. For agile theories, self-organising agile teams are not leaderless, while leadership in self-organising teams is meant to be light-touch and adaptive, providing feedback and subtle direction (Takeuchi & Nonaka, 2007). In this sense, leadership plays a less important role compared to traditional software development.

Sutharshan and Maj (2011) identify the relationship between PDI and agile principles. Cultures with a high PDI index are not expected to be involved in decision making, while in cultures with a low PDI index employees are not afraid of authority and the leaders are not bureaucratic. Meanwhile, Sutharsahn and Maj (2011) also propose a set of cultural specific agile attributes which associate with PDI. We selected several of them which we thought reasonable and interpreted them individually. For *transparency* and *authoritative*, due to the high PDI index in China, we infer that transparency should be low and authority should be strictly respected under the impact of an obsession with hierarchy. For *decision making*, it can be quick in China because the leader can play a crucial role while the employees are hardly invited to participate in it. This deduction can also be applied to *negotiation*, as the advance of the projects is usually in the form of orders.

Inspired by the concept of negotiation proposed by Sutharsahn and Maj (2011), there is also one agile value stating "*customer collaboration over contract negotiation*" (Beedle et al., 2001a). If the relationship between customers and suppliers is constrained by contract negotiation, the status of the customers is often considered higher than the suppliers. However, the collaborative relationship means the two parties are on relatively equal terms, which better promotes the development of the products.

As a conclusion, we assume that China's high PDI index conflicts with agile theories, which can be inferred from self-organising teams and collaborative relationships between the customers and the suppliers.

4.2 The Attitude towards Changing Requirements

Uncertainty avoidance indicates to what extent a culture programs its members to feel either uncomfortable or comfortable in unstructured situations (Hofstede, 2011). Countries with low UAI index should be comfortable with ambiguity and chaos. Based on the agile value "*responding to change over following a plan*" and the agile principle "*welcome changing*

requirements even late in the development" (Beedle et al., 2001a), we simply assumed that low UAI index helps to adopt agile development methods easily.

According to Newman and Nollen (1996), countries with low uncertainty work-related performance is better if rules are not well defined. Compared to traditional software development, ASD welcomes changing requirements even late in the development. In China, we found customers' requirements were changing because they were not familiar with the software development domain, and even some of them could not know the workflow of their product (Jiawei, 2018). As a country with low UAI, employees are used to changes, as they know their job is to develop software constantly, which means it does not matter how requirements are changing.

In conclusion, the hypothesis for uncertainty avoidance is made: China's low UAI index conforms to agile theories.

4.3 The Impact of Masculinity on Development

Masculine cultures lay importance on achievement. In the workplace, masculinity is reflected in merit-based opportunities for high-earnings, recognition, and rewards (Newman & Nollen, 1996). With a clear goal, achievements can be attained easily, and this can be reflected in the agile manifesto: "*working software over comprehensive documentation*" (Beedle et al., 2001a)".

According to the research of Newman and Nollen (1996), they found that employees with more masculine cultures performed better if they made more use of merit-based rewards for payment and promotion. In this sense, voluntarily working overtime is not a problem, as long as a worthy bonus can be offered. When projects are due, working overtime is unavoidable in China. Sacrificing leisure time to deliver a reliable product and get promotion reflects the masculine culture within the organisations. Meanwhile, this behaviour also boosts the development of the software, ensuring the agile principle: "*deliver working software frequently*" (Beedle et al., 2001a).

Thus, we assume that China's high MAS index conforms to agile theories.

4.4 The Conflict between Individuals and Team Cooperation

The first agile value is "*individuals and interactions over processes and tools*" (Beedle et al., 2001a). It advocates the significance of excellent individuals. Again, the individuals are mentioned in the agile principles: "*build projects around motivated individuals*" (Beedle et al., 2001a). In this way, a preliminary deduction is generated: agile theories emphasise individualistic values.

The IDV dimension implies the degree to which individuals are socialised, which means that low IDV figures can decrease the degree of interdependence within the team. Therefore, there is a conflict between the individualistic culture and team communication. High degree of interdependence makes team members hard to communicate and cooperate with each other, while the agile principles also stress face-to-face conversation as the most efficient and effective method of conveying information within a development team (Beedle et al., 2001a). Besides, performance was higher when managers emphasised individual employee contribution (Newman & Nollen, 1996).

Based on the evidence above, we generate the assumption: China’s low IDV index conflicts with agile theories.

4.5 Long-term Orientation and Customer Collaboration

In Hofstede’s cultural dimensions, long-term orientation refers to solving problems for the long run instead of making “quick fixed” (Newman & Nollen, 1996). For agile theories, it stresses customer collaboration in its values and sustainable development in its principles. What is reflected in the agile theories corresponds to the long-term orientation. As China scores high in LTO, the assumption in this dimension is that China’s high LTO index conforms to agile theories.

4.6 Summary of Assumptions

To solve the MRQ, “How do Chinese cultural factors influence the adoption of ASD approaches within Chinese SPOs”, we have made five assumptions for the five dimensions of Hofstede’s model, which can be seen in [Table 4-3](#):

Table 4-1: the assumptions between Chinese culture and agile theories

Hofstede’s cultural dimensions		Hypotheses
H1	PDI	China’s high PDI index conflicts with agile theories.
H2	UAI	China’s low UAI index conforms to agile theories.
H3	MAS	China’s high MAS index conforms to agile theories.
H4	IDV	China’s low IDV index conflicts with agile theories.
H5	LTO	China’s high LTO index conforms to agile theories.

5. SOFTWARE DEVELOPMENT PRACTICES IN CHINESE SPOS

In this chapter, we will introduce the results of case studies in Chinese software development. The profiles of the companies and the interviewees can be found in [Appendix C](#). The goals of the case studies are: (1) to learn the workflows of different Chinese SPOs and the reasons why they stick on their ways, and (2) to investigate the challenges of development in current stage associated with Chinese cultural factors. The PDDs which present the workflows of the development methods of the four companies are shown as below, and their detailed descriptions of the processes and concepts can be found in [Appendix D](#). The results cover the general software development process of each company, the supporting infrastructure during development, and the challenges they identified in the current stage. To investigate more helpful information, we adopted semi-structured interviews to collect the data, which allowed interviewees to share their opinion freely. Centred around the topics of “Agile software development” and “Chinese cultural influence”, the interviews were then conducted. The four participating companies are all Chinese SPOs, while they differ in size and field so that we can ensure the comprehensiveness of the results.

5.1 Alpha Software Company

We conducted two interviews with Alpha Software Company (ASC), which involved the director of the department who also played a role of product manager and a tester. Located in Shanghai, one of the first-tier cities in China, ASC has at around 1000 FTEs, which means that ASC is a large enterprise in China. ASC specialises in enterprise management software since 2000, and it has focused on omnichannel retailing for five years. Its product portfolio includes ERP system, supply-chain management system, CRM system, WMS system, and online shopping system.

The department which accepted our interviews was the shopping mall department. For the product, online-shopping system, it was designed for enterprises which desire to develop their online business in the boom era of net purchase. As a fast-developing domain, Chinese E-commerce has been tied to Taobao due to Taobao’s vast contribution to Chinese E-commerce. However, with the growth of settled merchants in Taobao, the merchants desire to develop their brands and bring in more user traffic to their official websites. Thus, the pain points have motivated the need for establishing a customised official shopping website.

Software Development Workflow

Due to the nature of the product, the developing process is divided into two parts: the standard product and the customised products. The standard product was invented firstly, and the customised products were developed based on the standard product, as expressed by the product manager in the interview data (INTV-A-1):

“We developed our standard product from scratch last year. The system was developed in a method combining Waterfall and Scrum. When we built the ground level of the system, Waterfall development was adopted. When it came to the implementation of specific business functions, we used Scrum as guidance to plan the schedule and priority of the requirements (INTV-A-1).”

Also, the product manager clarified the reason why they developed their product in this way:

“To construct the basic framework of the system, care and patience are required. The database design should be as accurate and comprehensive as possible, even if the table structure is often improved at a later stage. We would build the prototype of the system in the Waterfall process, which covers all the function modules of the system, although we would not realise any specific functions. Besides, the system interfaces should be planned and reserved. Later in the agile process, we ranked the priority of different function modules in terms of its importance degree and customer needs. After the standard product can be released, we will notify our partners to download or update our encryption source code from SVN, which allows them to customise the product regarding our customers (INTV-A-1).”

According to the statement of the product manager, great importance was laid on the base of the system. The rigours of the system design were prepared for better customer satisfaction and reputation in its industry.

When we asked what specific ASD activities had been covered, we found it varied somehow compared to existing Scrum practices:

“We don’t have stand meetings during developing. Also, we don’t have burndown charts. As it is not a standard agile software development process, no Scrum master was involved. The product manager and the development manager are jointly responsible for planning iterations. Besides, I produce requirements, and the development manager assigns tasks to corresponding developers. There is no planning poker in determining sprint item finish durations. Instead, we ask our developers how many person-days they can finish their task and record the provided person-day into the plan documents. We don’t have stand meetings and retrospective meetings. Time is always tight, and we don’t have a specified attendance time as we often work overtime in the day before (INTV-A-1).”

The development team just borrowed from the iteration idea of ASD and tailored the method in its way. For the artefacts produced during the development, the two interviewees both provided their own opinion:

“(As a product manager,) I am responsible for managing the development documentation. I designed the table structure of the database, and the development manager later modified it based on actual demands. Besides, the daily progress reports will be updated and sent to all the team members (INTV-A-1).”

“(As a tester,) I am required to write test cases before the testing process. The test cases are written regarding the requirement document, which is provided by the product manager. Personally, the requirement document is often vague in the description of system functions and cannot be updated in time. The test cases are usually presented in a sheet which contains the modules to be tested, the introduction of the function, the required input and output, and the desired results (INTV-A-2).”

In ASD process, heavyweight processes and comprehensive internal documentation are considered as no direct use to the end customers (Stettina & Heijstek, 2011). Luckily, the team has some documents at least, which depict the general development process and introduce each function of the system, although the feedback of the documents from the tester was not that optimistic.

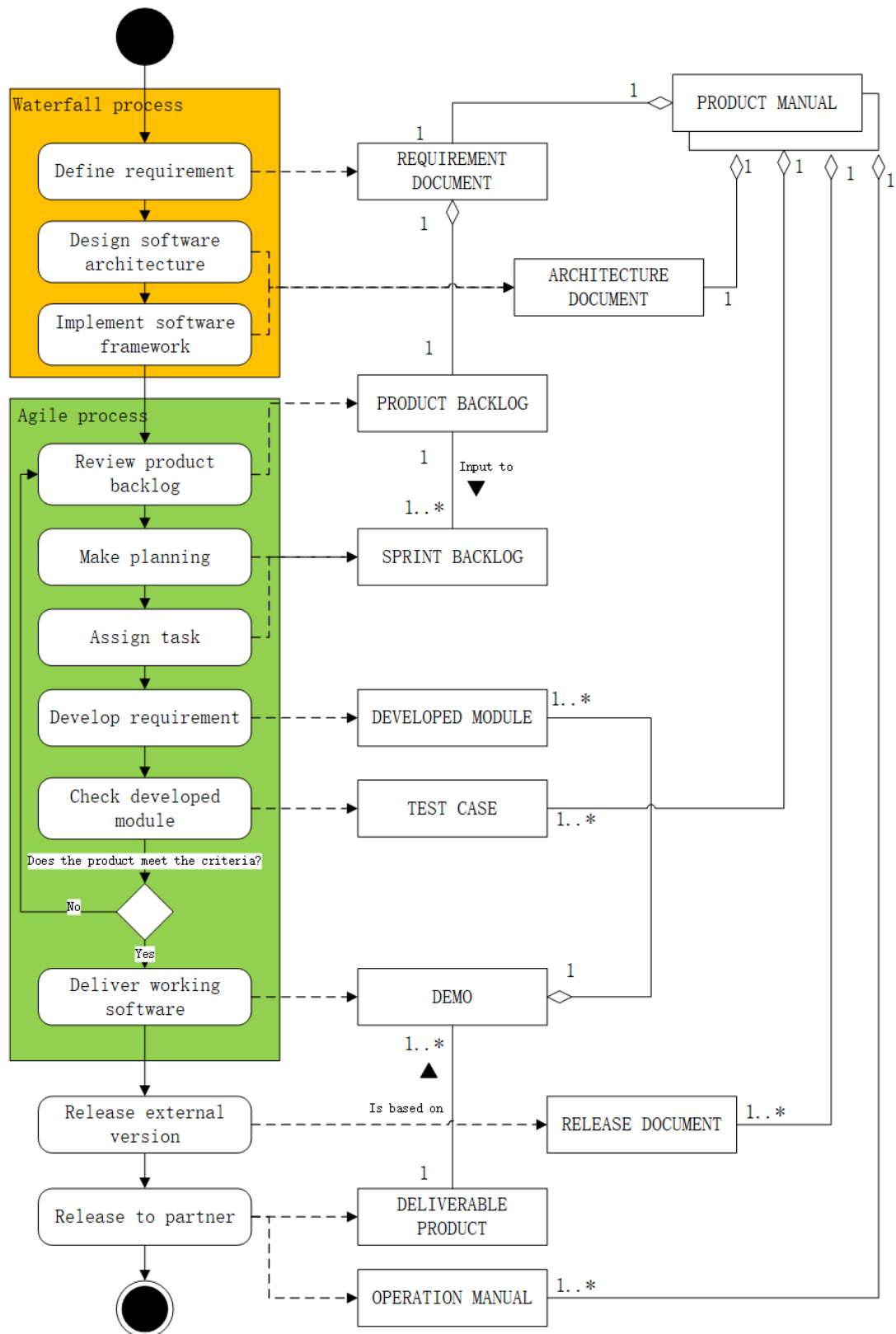


Figure 5-1: PDD of the software development method by ASC

According to the interviewees' statement, we can generate a PDD of their software development method as shown in [Figure 5-1](#). The method is divided into two parts: the first part realises the foundation of the system, and the second part deals with the specific functional requirements like sales management of the online shopping mall. After the product

is released, projects will be assigned to their partners, and a project manager will be involved in the secondary development together with the customers and partners, to ensure the product can be delivered on schedule and within budget.

Besides its current development pattern, we also need to mention the Bimodal IT which it adopted in the past. ASC once used Bimodal IT to develop their standard product and re-develop the product based on the requirements of the customers in parallel. According to the interviews, we explored the reason why the Bimodal IT failed. In terms of the statement of their product director, the leader of mode 2 (agile) group was supposed to shoulder the responsibility.

“The leader of mode 2 group was a developer. He was not good at managing the relationship among the team members. Limited by his experience, he commonly added new requirements into the fixed Sprint backlog, which made the developers within the team feel stressful and exhausted due to working overtime at the end of the Sprints (INTV-A-1).”

For the team members, they also should be responsible. As reflected by the director, product manager of the team did not review the requirements of the customers carefully, which caused the problem that customers felt unsatisfied when they verified the working software in iterations. Meanwhile, the developers also spoiled their worktime to do irrelevant stuff like playing mobile phones. The eight-hour working arrangement could be wasted, and working overtime was the only choice. The results of the Bimodal-IT pattern in ASC was that all employees of the mode 2 group resigned while the leader of the group joined the mode 1 (traditional way/waterfall) group.

For their current development approach, it was favoured by the team members. As the standard product is finished developing, the development team focused on customising the products based on the requirements given by the customers.

Supporting Infrastructure

For tools used in ASC, they did not develop their infrastructure, as they have already purchased powerful supporting tools before. For daily communication, they use DingTalk⁵ to contact each other, and the application can also be used to check daily attendance, book meeting rooms, live broadcast teaching, and file storage, sharing and sync service. To monitor the progress of the project, they use a Kanban-like tool named OSChina (see [Figure 5-2](#)).

⁵ DingTalk is a free communication and collaborative multi-terminal platform developed by Alibaba Group, which is available on PC, Webpage, and mobile phone, and supports the transfer of files between mobile phones and computers.

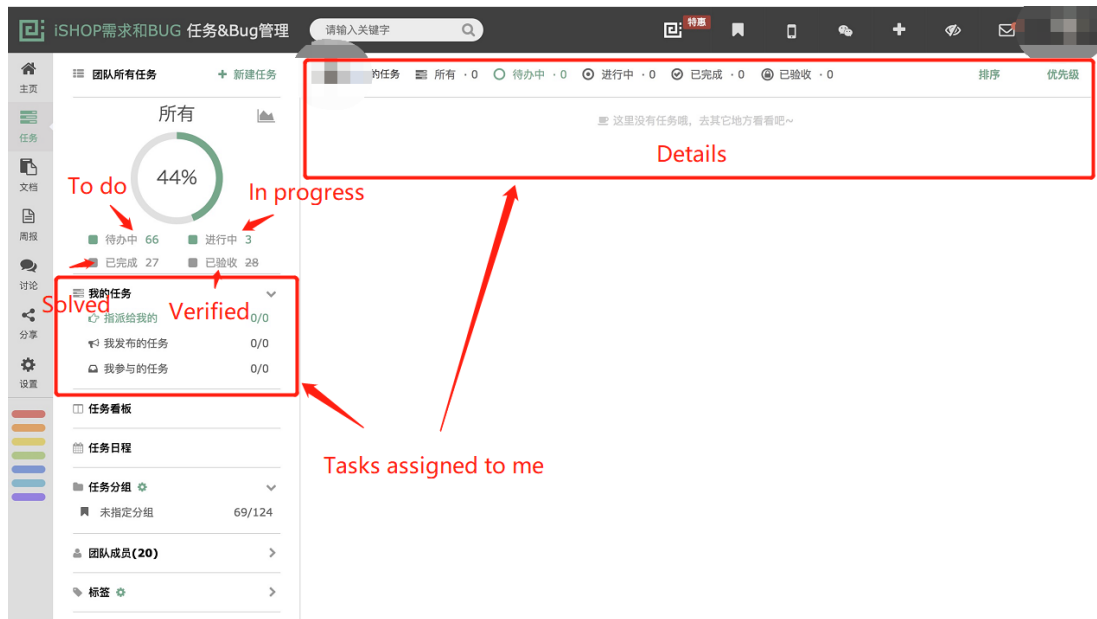


Figure 5-2: ASC's Kanban-like tool

However, when it comes to the product backlog management, there is no corresponding tool. The product manager keeps all requirements into the requirement document and marks the selected requirements in an iteration. It was illustrated by the product manager:

“For now, we don’t have a tool to manage the requirements in Sprints. The common solution is to mark those selected requirements from the requirement document and re-organise them into a new document. In this way, it can be a Sprint backlog, if you have to let me define how we manage our backlog... (INTV-A-1)”

To manage the progress of the development and make it available for all the team members, the team still uses a relatively traditional method: the product manager is supposed to update the progress of each member and organise the data into a sheet, and the sheet is sent to everyone via email. The sheet can be seen in [Figure 5-3](#), the left side of which shows the names of the team members, while the right side describes the start date of the project, the planned ending date, the lasting days, and the progress of completed work.



Figure 5-3: The screenshot of the project progress by ASC

Insights into Chinese Culture

The product manager has been aware of the benefits that agile methods have brought, as expressed in the interview data:

“It is true that we have successfully built an aggressive and optimistic atmosphere by adopting agile thought. Members of the team are self-disciplined as we can see the development progress of each other. When a new version is to be released, everyone is motivated and glad to work overtime to ensure the release goes smoothly (INTV-A-1).”

However, when it comes to the conflict between Chinese culture and agile thought, the product manager talked a lot. For the cultural dimensions of *Power Distance* and *Individualism* in the Hofstede model, he offered his opinion:

“You cannot neglect relationship between superior and subordinate in China. It means that self-organising team is not practical. Each team needs a leader. I think this can clash with the existing agile principles. If a team has no leader and people only care about their own business, it makes progress delay. I think it is necessary to centralise the power in the product owner or Scrum master, to better monitor the development (INTV-A-1).”

The figure of PDI in China is high, which means hierarchies are inevitable. Besides, people are used to following the orders from their superiors due to the collectivist culture. It is somehow easier to manage a team, as long as the leader can be experienced and has a strong sense of time. In this way, the first challenge is to select an appropriate candidate as a team leader.

For the second challenge, the product leader mentioned the changes in requirements from their customers, which corresponds with *Long-term Orientation* and *Uncertainty Avoidance* of Hofstede model:

“As we sign a contract with our customer, which means the requirements have already been determined before the development, we should reject the changes from the customer by definition. However, we desire long steadily and harmonious relationship with our customers. As long as it meets existing budget, we are often flexible with the changes and notify our customers that we will solve the changes during the development although they are not contained in the contract (INTV-A-1).”

To keep a long-term relationship with customers and reduce uncertainty, ASC has to compromise on changes of the requirements. A proper validation process for the customer requirements is necessary. Besides, the product manager thought it would be useless to allow customers to test the product before the product is released:

“Although an MVP (Minimum Viable Product) should be delivered, it misses the realisation of many specific functions. On the one hand, customers will lose trust in the product due to the deficiency of the product. On the other hand, a lot of time is wasted as we have to tell our customers that the system is still in progress and those problems are not developed temporarily instead of bugs. Usually, customers can test the product after we release the product (INTV-A-1).”

For the comments of their Bimodal IT experience, we draw conclusions that leaders should be experienced and professional enough to handle the business including the interpersonal relationship and requirements scheduling, or the leader-member relation will be spoiled and

inferiors will lose faith in their superiors. In this way, high power distance index can be necessary, on the condition that the leaders should be qualified.

5.2 Beta Software Company

Two semi-structured interviews were also conducted in Beta Software Company (BSC), and the roles of the participants are the director of the company and a developer respectively. BSC was established in 2000 and is still a small enterprise which has less than 20 employees. It is located in a provincial capital, which is a second-tier city. In the beginning, BSC was only focused on developing ERP systems for machine factories; later it also involved in the environmental protection business offered by local government.

The director of BSC accepted our requests for interviews in May 2018. The interviews were conducted offline. The participants first introduced their ERP product and later talked about their views on the impact of Chinese culture on ASD.

Software Development Workflow

The director divides their development into three steps: firstly, the director is responsible for scoping the business requirements; secondly the director designs the data modeling, human-computer interaction use cases, and writes the requirements documents; thirdly, the director splits the requirements into tasks and assigns the tasks to different developers to realize desired functions:

“(As the director of the company,) I am in charge of negotiating with the customers to study their requirements and assess the difficulty of the requirements. As a small company, I am the only product manager to handle the requirements. The structure of our company is fairly simple: I am the only leader, and my employees should follow my ideas. In this way, I assign tasks to appropriate developers, and they just need to complete the tasks and deliver them on due days (INTV-B-1).”

When the director is asked why there is no other product manager, he stated:

“I have been working in this area for 20 years. I clearly know what my customers need, and the projects in the past went smoothly under my control. Thus, I don’t think it’s really necessary to hire extra product managers, considering our budget and actual demand in budget and workforce (INTV-B-1).”

To have an objective recognition of the development process, we also learned the opinion of the developer, and his feedback towards the assigned tasks was positive:

“I just need to follow my boss’s order. It is really nice that the tasks are specifically described. When we think the tasks are unreasonable, we can approach our boss to discuss it. Of course, it rarely happens. Besides, we have the products which were finished in the past, which can be a reference when we have trouble developing. In a word, we just need to focus on our tasks without being bothered by other stuff (INTV-B-2).”

As the director claimed that their development method was influenced by agile thought, we also explored how they draw lessons from ASD and what are the differences. In terms of the director’s answer below, we can see that BSC basically agrees with agile thought. For the

disagreement, we will investigate it in detail in the section "[Identified Advantages and Drawbacks](#)".

"Influenced by agile thought, especially the thought of delivering working product frequently, we divide our development process into different iterations. What makes it different from the standard method are that (1) the partition criterion is defined based on the workflows of the customers; (2) we don't manage the backlog, and the requirements are all organised and updated in the requirements document, which means that selected requirements are just labelled in different iterations. For the agile principles, I can say that we all agree on it except the self-organising team. And agile methods can be relatively easy to implement in small teams. (INTV-B-1)."

"When it comes to the detailed information of the development, the team has one product manager, one project manager, one software architect, (at around) four developers, and a tester. Personally, I think the ideal group size of a development team can range from 3 to 5. Otherwise, it would be not that efficient. For the length of the projects, it can be one, three, or five months, based on the complexity of the system. For the length of each iteration, it lasts from one week to a month (INTV-B-1)."

For the artefacts generated during the engineering process, the director was in charge of managing the requirements document, while the test reports will be sent to the persons in charge after testing.

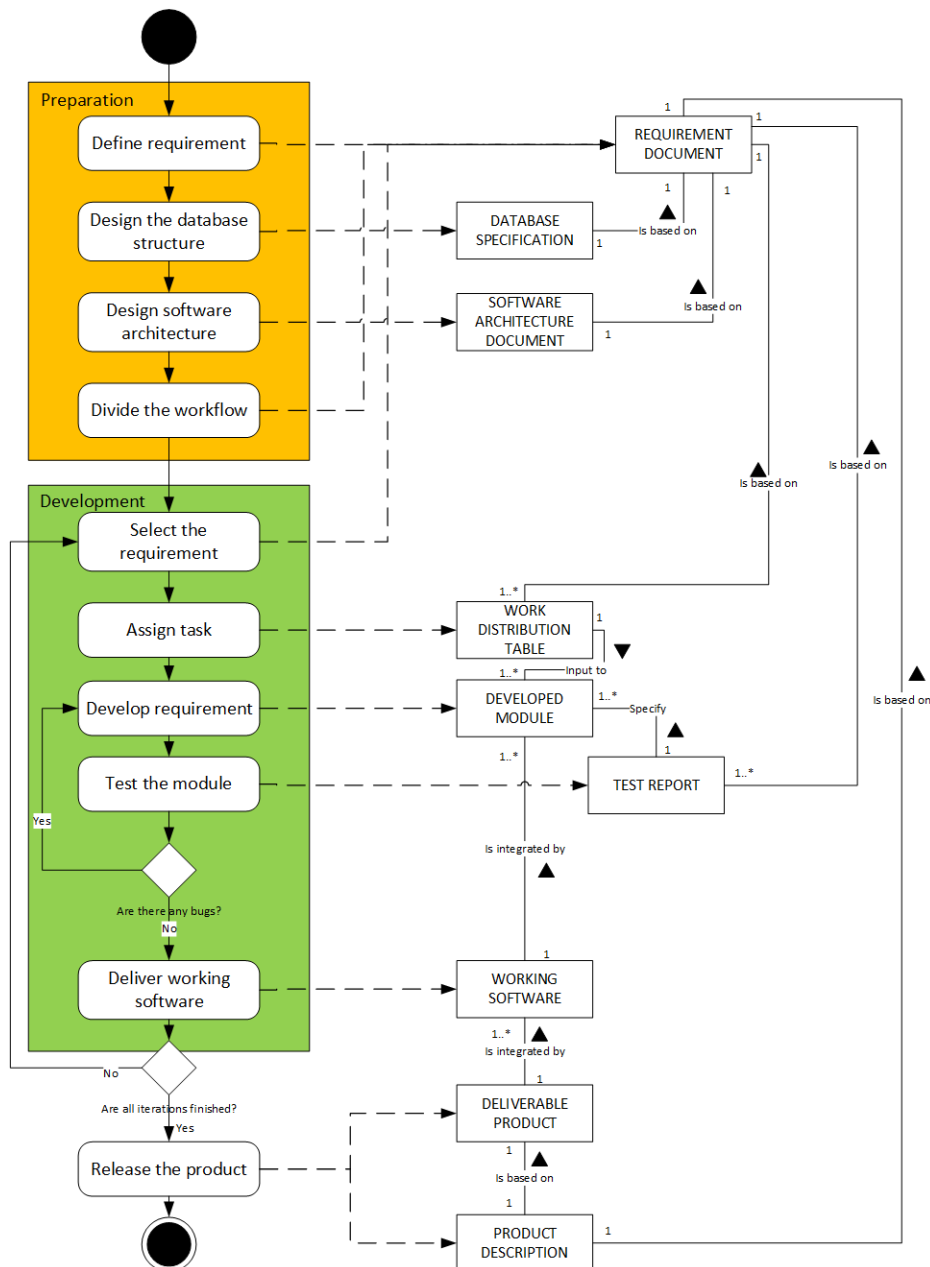


Figure 5-4: PDD of the software development method by BSC

As can be seen in [Figure 5-4](#), we draw the PDD of the software development method by BSC regarding the director's statement. Their method basically follows the agile thought. However, compared with Scrum, they do not have product backlog and Sprint backlog to manage the requirements during iterations.

Supporting Infrastructure

When we talked about the supporting infrastructure, the participants introduced their tools. Restricted by the capacity and the size of the company, they use QQ⁶ to communicate remotely. To maintain versions of the artefacts like source code and documentation, SVN is

⁶ QQ: an instant messaging software developed by Tencent, which is similar to MSN. See: <https://im.qq.com/index.shtml>

selected. For testing, Bugzilla was used in previous time, while the company now prefers digital documents to record and report the bugs:

“We used Bugzilla in the past. The effect was not ideal. Now we just describe the bugs found during the testing phase in a Word document and send the document to the developers. After they repair the bugs, we will test them again and change the status of the defects in the document once the problems are solved indeed (INTV-B-1).”

For daily communication, the interviewees both stated that they prefer face-to-face communication, as it is considered as the most direct and efficient way to convey their ideas:

“We love face-to-face communication. When I (the director) assign tasks to the developers, they can bring forward their opinions when they are confused, or they think the task is unreasonable. We only use QQ when there is someone who cannot show up in the workplace (INTV-B-1).”

“I prefer face-to-face communication to QQ. When you use QQ, you need to wait for others’ response, and you can never know if the guy has read your message. It can be convenient when we just talk face-to-face, as the number of our colleagues are not many and they just sit in the company during working hours (INTV-B-2).”

Insights into Chinese Culture

For the impact of Chinese culture on ASD, the director offered his ideas. When it comes to the high PDI index and low IDV index of China, the director regard it as a favourable factor in corporate governance and promoting the development, and he stated that he valued an obedient team more than gifted individuals:

“Our company structure is simple. I am the boss, and my employees should listen to my order. The self-organising team is hard to form in China, as people are used to listening to their superiors. There is a famous proverb: one Chinese can be a dragon, while ten Chinese would be a worm. I am an experienced expert, and I know what my customers need, so the employees need to follow the instructions. In this way, an obedient team is preferable. It is true that we all like talented individuals, but one powerful developer cannot deliver a product within our time budget. Also, it costs more to hire an experienced developer, while hiring novices can be economical and practical (INTV-B-1).”

For high MAS index, the director stressed that their primary goal is to deliver the product on time. Besides, the practices like working overtime willingly and respecting the experienced experts indicate that masculinity does exist in the company and plays a positive role.

As China scores low in UAI, it means that Chinese welcome changes. The director agrees on this phenomenon, and talked about how to manage the changing requirements based on his experience:

“We don’t think the changing requirement is a bad thing. Sometimes even our customers don’t know what they really want, as they may never know the software domain. This is also the reason why the customers in the early phase often changed their requirements during the development. Now I would first introduce my experience in developing ERP system in the phase of requirements investigation, and tell them what their system should be designed, based on their business. This is also a process gaining agreement from the clients. Once they think you

are really a domain expert and you have already known their pain points, they would trust you and hardly change their requirements (INTV-B-1)."

The director said long-term orientation is also favoured by them. Compared to getting short-term benefits by delivering products irresponsibly, he made friends with his customers by constantly maintaining the products. Besides, he was also accustomed to the typical Chinese negotiation way, which is the culture of drinking. Drinking can ease tension and facilitate social exchange among Chinese people in business (Hao & Young, 2000). Guanxi network was attached importance to by the director. On condition that several companies can offer similar products and the decision makers of the stakeholders are indecisive, guanxi can be used to strive for the completion of the contract. Gift-giving and entertainment used as a foundation for future business relationships are common in many cultures (Thomas W. Dunfee, 2001):

"As our major customers are from the government, there are many competitors. On the one hand, the customers are often won by my rich experience and the good reputation of our products in the past. On the other hand, gift-giving and drinking are necessary. Once the customers feel satisfied and consider you as a friend, the business is basically done (INTV-B-1)."

5.3 Gamma Software Company

Gamma Software Company (GSC) is a state-owned enterprise which focuses on the power sector. GSC has more than 20000 employees, and its business area has covered most of the provinces in China. Compared to private companies, its secrecy rules are stricter. We tried to invite the employees from GSC to accept an interview, and a developer participated in our project. The participant belongs to the IT department, which is located in Beijing, one of the first-tier cities in China. This department is responsible for developing software used by internal employees and cooperative partners. The interview was conducted at the end of May 2018.

Software Development Workflow

The structure of the department is different from other companies participating in our project. The department of GSC is divided into seven groups, and a group has at around seven persons. Most of the groups are responsible for developing, while only one or two of the groups are in charge of testing. What interesting is that there is no product manager in the department, and the person who manages the requirements is the department manager:

"The department manager receives the requests from the business division or his superiors. He will reorganise the requirements into modules and estimate software effort. Based on the estimated effort, he assigns each module to different groups (INTV-C-1)."

Once the group where the interviewee belongs is assigned a module, there is also a specification indicating everyone's corresponding task. In this way, the development becomes goal-oriented and simple:

"I just needed to do what I was assigned. Commonly the description of the task is clear, and we can ask our manager when we have trouble in understanding the requirements. What we should do is following the orders and finish the task within the stipulated time (INTV-C-1)."

Once the module is developed, the testing groups start to verify the module. When all modules are all verified and have no defects, the modules are integrated together, and all the testers in different testing groups will test the software systematically. Then the software will be released if the test passes.

As a representative sample in our project, we intend to explore their attitude towards documentation. One of the agile values is “*working software over comprehensive documentation*”, and this value is often proudly quoted by those companies which lay no importance on documentation. According to our interviewee, they still pay attention to carefully-written documentation, as it is regulated by the company:

“Documentation is indispensable as our manager require us to deliver it together with the modules. What’s more, unified code annotation and format are inherent. All of these behaviours help better review the code and serve as a way of knowledge management. Although we have iterations within the development and utilise the thoughts from ASD, pure coding without documentation is inadvisable. (INTV-C-1)”

For the separation of the iterative process, the criteria for dividing iteration is the priority of the requirements, which is based on the feedback of the business division. The new iterative process will also consider the modules which have defects in the last iteration. Generally, a project lasts between two and three months, less workload of which employs around three development groups, while more workload of which employs seven groups. An iteration usually lasts for five workdays.

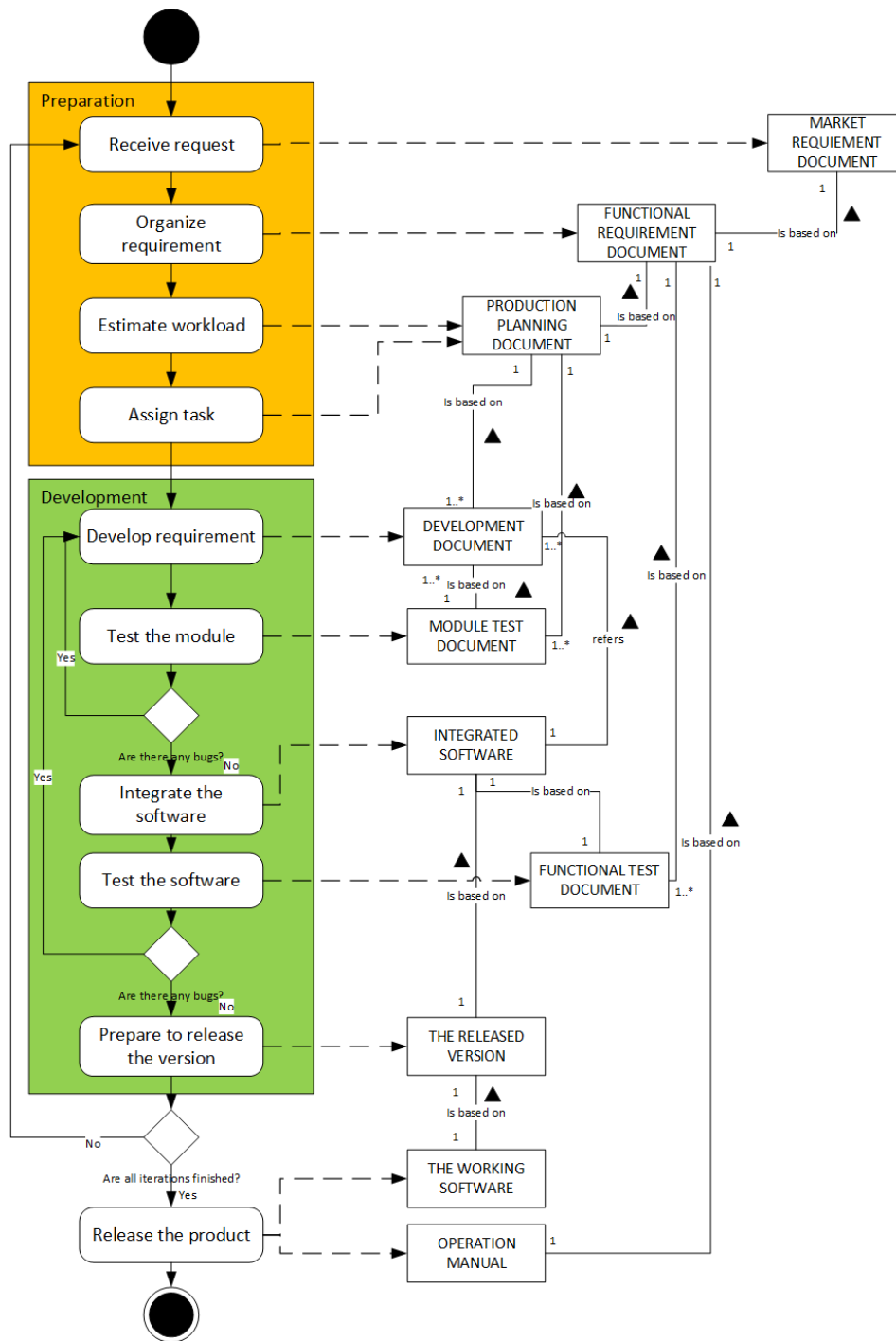


Figure 5-5: PDD of the software development method by GSC

In [Figure 5-5](#), the software development method of GSC is presented in the form of PDD. It is clear that its new iterations start from the beginning of the planning phase, as the department manager needs to obtain new requirements from the business division, which is different from the methods proposed by other companies participating in our research.

Supporting Infrastructure

For the tools supporting their development, the participant talked about their customised tools based on existing open source software except for SVN, which is used to maintain

versions of artefacts. For project management, they have developed a software which is similar to ZenTao⁷, on which they can manage the defects, test cases, and releases. For internal communication, a customised tool was also developed for the confidentiality and safety reason. As the developers simply need to deliver their code and documents before the deadline, there is no stand meeting and burndown chart:

“We don’t need to report our progress every day. Everyone in the department is self-disciplined and stressful. Deliver the code in time, or the bonus will be cut. When it comes to money, people in the department will be motivated, and the tasks will be finished without delay (INTV-C-1).”

Insights into Chinese Culture

The participant’s answer is organised based on the Hofstede’s cultural dimensions. For PDI, the interviewee’s company is no exception – stiff hierarchies are inevitable. The opinion of the interviewee towards hierarchies was dialectic. On the one hand, in his department, there was a lack of the obsession with hierarchy, which he thought it was helpful. On the other hand, it was hard for them to handle the work between the departments, as the managers of the business division often took advantage of their positions to raise the priority of their requirements without knowledge about software development:

“I am only an executor, and I know nothing about staff management. For our department, we value the hierarchy as our manager could give clear orders so that the projects can go smoothly. However, it can be troublesome when we have to associate with the business division. When a version is released, those guys from different divisions will scramble for increasing the priority of their requirements. You can hardly talk to them because they think you are their subordinates and you should follow their orders, although they are not your direct superiors (INTV-C-1).”

For UAI, the developer was honest. He disliked the changing requirements, but those changing requirements could not bother him directly. As a result, he was not inconsistent with the changes:

“I don’t like those changing requirements, and I think most of them are generated because of lack of domain knowledge and deep thinking of the product. However, it cannot bother me at all. I am a developer, and I have endless code to write. The changing requirements mean typing some other code for me (INTV-C-1).”

For MAS and LTO, the participant was ambivalent. The developers were mostly young men, and they attach great importance on their private time. However, when there is a product to release, they are required to work overtime. To ensure the product can pass the test and their KPIs qualified, they have to accept the reality.

When it comes to the dilemma between talented individuals and united teams, the interviewee’s answer was different from our existing data. Compared to a nice team, he thought talented individuals were more valuable to the development and the organisations:

“The labour is cheap nowadays. A development team can even be formed by some guys who only attend training classes and have no academic background knowledge. On the contrary, it

⁷ A Chinese project management software. It is a professional open source Scrum tool, which is the most popular development team collaborative tool in China with 1.4 million teams. See: <https://www.zentao.pm/>

is hard to find some talented persons, who have abundant knowledge reserve and can promote the development (INTV-C-1)."

5.4 Delta Software Company

Delta Software Company (DSC) participated in our research project in May 2018. DSC is a platform linking between Bed and Breakfast (BNB) owners and tourists, and its major target areas are Chinese mainland, Taiwan, Japan and Korea. DSC has around 200 employees and is a medium-sized enterprise. Located in Shanghai, the development team has 20 staffs, and the roles include product managers and developers. A senior developer and a product intern were designated to participate in our project.

Software Development Workflow

The basic workflow was described by the developer, as DSC just established the position of the product manager recently and the senior developer had worked in the company for three years. According to the developer, the requirements were given by the customers, which involves the BNB owners and the big clients from the tourists. Later the requirements were organised by the operation department and transferred to the technology department. The developers of the technology department would sit together and refine the requirements into concrete development tasks. Based on existing architecture, the work would be carried out iteratively. Due to the constraints in areas like personnel and funding, the developers also played the roles as testers. As a result, the developers were supposed to test their code. After the prerelease was validated successfully, the product was supposed to release:

"We write unit code tests while coding. Each developer is responsible for their modules. When testing is passed, the code is merged to the main branch, and system integration testing begins (INTV-D-1)."

As the position of the product manager was established recently, the major tasks are writing requirements document before the development and monitoring the overall progress of the projects:

"The development process is already mature. I only need to attend the meeting when our developers discuss the requirements of the operating departments and record what they should do respectively. In the future, I will also substantial involve in the requirement review process (INVT-D-2)."

The development was divided into different iterations regarding the priority of the requirements, while there was no product backlog and Scrum backlog. The projects in the past usually lasted between one and two months by the workload, and each iteration lasted for one week. By the way, both of the respondents referred to their "996" working arrangement:

"When there is an important project going online, we have to work six days per week, and each workday begins at 9 a.m. and ends at 9 p.m. (INTV-D-1)."

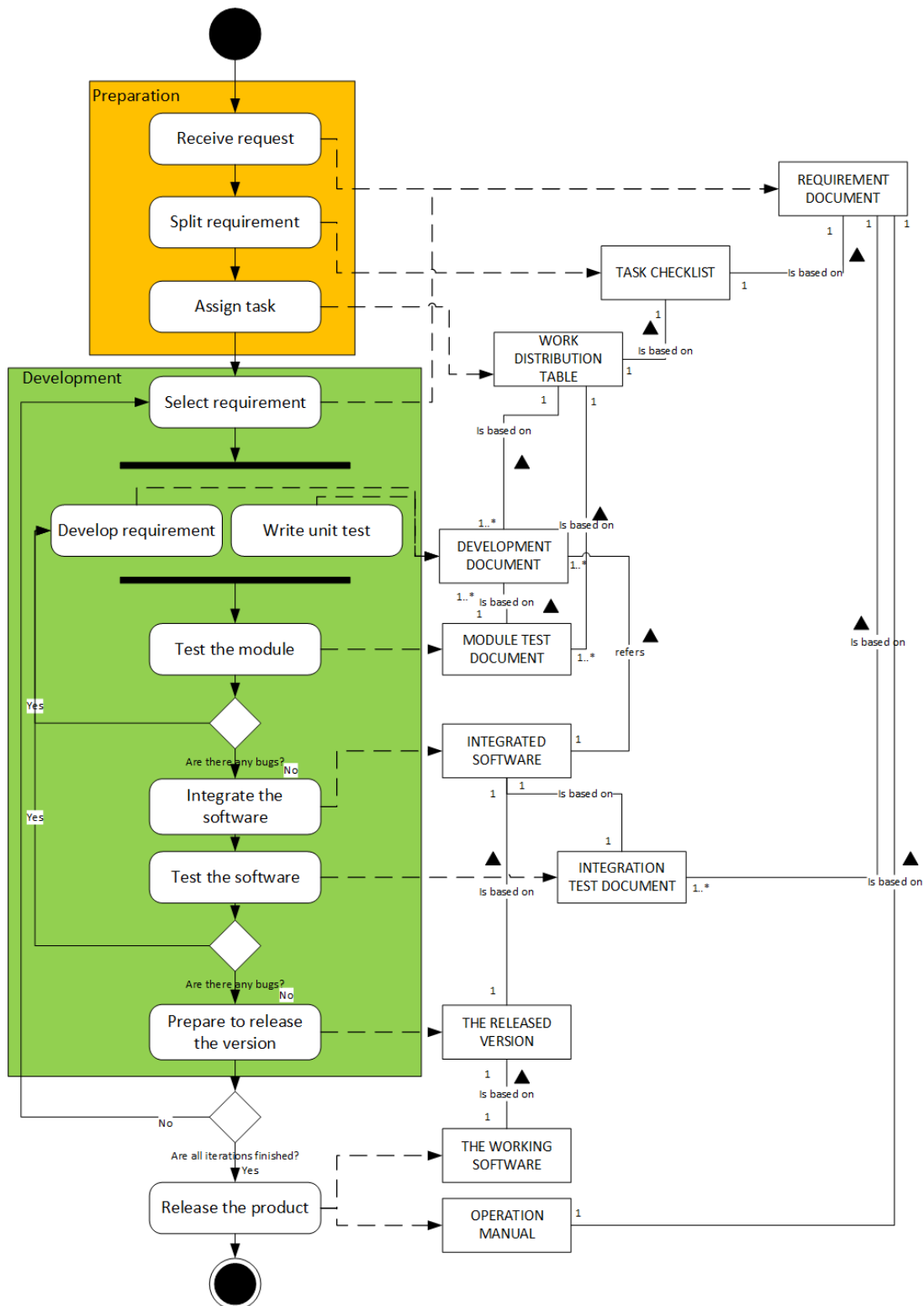


Figure 5-6: PDD of the software development method by DSC

In [Figure 5-6](#), the PDD of the software development in DSC is presented. The requirements are organised and settled in the beginning, while in each iterative process requirements are selected regarding their priorities. After precise and serious testing, the product is delivered.

Supporting Infrastructure

In GSC, face-to-face communication is favoured and promoted. When there is someone not coming to work, Wechat⁸ is also used to keep in communication. To manage the project, Tower⁹ is used to manage and monitor the to-do list. Compared to the companies aforementioned, GSC uses Git to control the versions of their products. Besides, Jenkins is applied to support continuous build process.

Insights into Chinese Culture

For the opinions towards Chinese culture, both of the respondents agreed that hierarchies do exist in their company. They thought it would be helpful as long as the leader can be objective and experienced:

“A good leader can make the team efficient. Our department manager is caring and skillful, and every recruit is assigned to a senior employee to help them adapt to the new environment quickly. We appreciate these privileges (INTV-D-2).”

As mentioned before, the “996” working arrangement was introduced by the interviewees. The reason why this arrangement generated was their desire and determination to build products with high quality, which corresponds to the sign of masculinity mentioned by Hofstede (2011). Besides, their attitudes towards the arrangement were relatively positive:

“Working under this horrible regulation, we have already gotten used to it. We can get more as we work overtime, which is acceptable. On the other hand, as other colleagues also accept this working arrangement, it would be abnormal to object (INTV-D-2).”

We can see from the intern’s feedback that they work overtime to seek positions in the company, which is the feature of the long-term orientation. Besides, the group psychology is also reflected, which corresponds to the Collectivism culture.

For changing requirements, both of the interviewees regarded it not that troubling. Same as the viewpoints of the respondents in GSC, they thought their duties were writing documents and coding constantly. Thus it doesn’t matter what they are writing or coding.

⁸ A Chinese multi-purpose messaging, social media and mobile application developed by Tencent. Available at: <https://www.wechat.com/en/>

⁹ A collaboration tool for software development teams. Users can manage their current projects, monitor the progress of each team member, and communicate easily on this platform. See: <https://hk.tower.im/>

6. SUMMARY OF THE FINDINGS

In this section, we will summarise the results of the literature study in [Chapter 3](#) and the findings from the interviews in [Chapter 5](#). By summarising the findings, we can have a deeper insight into the impact of Chinese culture on agile adoption, which helps in summarising the results of the research by the assumptions made in [Chapter 4](#).

The results of the four firms in the interviews are summarised in terms of their size, corporate property, a brief description of their methods, and the attitudes towards Chinese culture based on Hofstede's cultural dimensions.

For ASC, it is a large-scale privately-owned enterprise. Their basic development method can be described as (1) collecting and organising the requirements; (2) designing and implementing the architecture; (3) developing and testing the software iteratively. Their attitudes towards the impact of Chinese culture on agile implementation is organized by the cultural dimensions: (1) high PDI index of China benefits the agile adoption; (2) low UAI index of China benefits the agile adoption; (3) high MAS index of China benefits the agile adoption; (4) low IDV index of China benefits the agile adoption; (5) high LTO index of China benefits the agile adoption.

For BSC, it is a small-scale privately-owned enterprise. Their basic development method is: (1) managing the requirements; (2) designing the bottom layer of the product; (3) developing the product iteratively. The attitudes towards the impact of Chinese culture on agile implementation are the same as the respondents from ASC.

For GSC, it is a large-scale state-owned enterprise. Their basic development method is: (1) organising requirements for each iteration; (2) estimating the workload of the project and assigning tasks to corresponding persons; (3) developing and testing the desired fragments iteratively. The attitudes towards the impact of Chinese culture on agile implementation are slightly different from the previous two: (1) high PDI index of China plays a double-sided role in agile implementation; (2) low UAI index of China benefits the agile adoption as they accept changes, in spite of the indifferent attitude of the respondent; (3) high MAS index of China benefits the agile adoption; (4) low IDV index harms the agile adoption; (5) high LTO index of China benefits the agile adoption.

For DSC, it is a medium-scale privately-owned enterprise. Their basic development method is (1) organising the requirements from the operation department; (2) assigning the tasks to corresponding persons; (3) developing and testing the product iteratively. Their attitudes towards the impact of Chinese culture on agile implementation are the same as the feedback of ASC and BSC.

To better demonstrate the results, the answers of the respondents are also organised briefly in [Table 6-1](#) as below:

Table 6-1: the summary of the results of the interviews

Company	Size	Corporate Property	Attitudes towards the impact of Chinese culture on agile theories				
			High PDI	Low UAI	High MAS	Low IDV	High LTO
ASC	Large-scale	Privately-owned	Positive	Positive	Positive	Positive	Positive
BSC	Small-scale	Privately-owned	Positive	Positive	Positive	Positive	Positive
GSC	Large-scale	State-owned	Double-sided	Positive	Positive	Negative	Positive
DSC	Medium-scale	Privately-owned	Positive	Positive	Positive	Positive	Positive

7. FINDINGS AND ANALYSIS

In this section, the empirical findings will be presented in terms of the Hofstede’s cultural dimensions. Besides, we try to analyse why the hypotheses are accepted or rejected.

According to the hypotheses made in [Chapter 4](#) and the results of the interviews in [Chapter 6](#), the findings of the research can be concluded. In [Table 7-1](#), we can see the results of the empirical research. The assumptions of power distance and individualism were found inconsistent regarding the results of the interviews, while the rest were consistent. We will further explain and analyse the reasons in this section.

Table 7-1: Findings of the research regarding the hypotheses

Label	Hypothesis	Findings compared to the established hypotheses
H1	China’s high PDI index conflicts with agile theories.	Inconsistent
H2	China’s low UAI index conforms to agile theories.	Consistent
H3	China’s high MAS index conforms to agile theories.	Consistent
H4	China’s low IDV index conflicts with agile theories.	Inconsistent
H5	China’s high LTO index conforms to agile theories.	Consistent

7.1 PDI Hypothesis

We assumed that cultures with low PDI index could easily adopt agile methods. However, it turns out to be inconsistent regarding the empirical data gathered from the interviews.

As mentioned in [Chapter 4](#), nations with high PDI index prefer not involving their employees in decision-making phase, and the unequal relationship between the customers and suppliers through negotiation is not praised. In this way, we made our assumption about the high PDI index of China. However, according to the interviews, we found that our interviewees, both the leaders and the employees, were in favour of the clear regulations and hierarchies. According to Fan (2000), one basic human relation is described as *master and follower*, and its principles should be loyalty and duty. Besides, due to the thoughts of hierarchy and order proposed by Confucius, children are taught to be obedient, education is centred with teachers instead of students, and subordinates are used to be told what they should do. This is no wonder why our participants felt strange and unpractical when we talked about self-organising teams and the practices “no title” and “no leader” of Scrum (Schwaber & Beedle, 2001).

The leaders who participated in our interviews had experience in managing teams and delivering products timely. The employees accepting our interviews also emphasised the importance of a good leader. Restricted by the current system of Chinese companies, it is hard to remove the hierarchies. However, we can conclude the interviews that high power distance also helps to promote the development of the products, as long as the employees are content with the orders from their superiors, and the leaders can be experienced enough to manage

their teams and assign tasks legitimately. As the respondents were all satisfied with the performance of their teams, it can be understood why the findings of the interviews are inconsistent with the established hypothesis.

7.2 UAI Hypothesis

The low UAI index of China was assumed as a beneficial factor to boost the agile adoption, and it is found to be right by the results of the interviews.

Requirements are changing due to many reasons. There are three reasons which can be prevented: (1) the customers missed requirements; (2) the customers identified defects; (3) the customers found that they hadn't understood their practical demands. However, there are also changes which cannot be prevented, like the politics, marketing, and legislation. Faced with these changes, Chinese SPOs can handle the changing requirements by adding them into next iterations or readjusting priorities, which does not trouble their development. Based on the findings of the interviews, we found the statement of our interviewees conforms to our underlying assumptions.

7.3 MAS Hypothesis

The high MAS index of China was also assumed as a favourable factor for agile development, and it is proved right.

Working overtime means sacrificing leisure time to ensure the progress of the development. Generally, it often goes with deserved rewards. For the "996" working arrangement of DSC, it is not a single case, which also occurs in many major Internet companies like Huawei and Alibaba. Although there might never be a method to accurately measure the outcome and efficiency, yet everything real is rational. Employees accept the reality that working overtime can happen before the project delivery, not only because of the management's request, but also the better chance of promotion. On the other hand, for the companies, they also need to be ambitious and assertive to deliver products of good quality and maintain the reputation for efficiency and effectiveness. There is no wonder why masculine culture contributes to the agile development.

7.4 IDV Hypothesis

Similar with the PDI hypothesis, the hypothesis related to individualism was found inconsistent with the existing hypothesis in terms of the feedback of our interviewees (ASC, BSC, and DSC).

Influenced by the collectivist culture, employees are used to working together and listen to the orders from their superiors. Agile development focuses on the talents and skills of individuals, molding the process to specific people and teams (Society & Foresters, 2016). This is also why we made our hypothesis that agile theories prefer cultures with high IDV index. However, there is also a problem in striking a balance between individualism and teamwork. Regarding the interviews, the only objection was proposed by the developer in GSC, who was a talented individual and had career changes experience before. Combining the ideas of the interviewees, especially the leaders, we found the individualistic culture cannot adjust to the local conditions in China. Compared to individuals with talents and skills, teams with

obedience and harmony are preferable. Based on their working experience, maintaining skillful individuals would cost more than regular teams, and running off of valuable talents cannot be avoided as job-hopping has already become a normal phenomenon.

As a result, collectivism culture is preferred by agile development in China, and we can make out why the findings are inconsistent with the established hypothesis.

7.5 LTO Hypothesis

The high LTO index of China was also assumed as a favourable factor for agile development, and the hypothesis was consistent with the actual findings.

Long-term orientation was connected with customer collaboration regarding the agile values (Fowler & Highsmith, 2001). Regarding the results of the interviews, the interviewed companies have a number of strategies to strengthen the relationship with their customers. Guanxi network was stressed by the director of BSC, and activities like social drinking and gift-giving contributed to reaching the agreement. Except for the company-level, employees also stick to the long-term orientation, which was embodied in voluntarily working overtime and keeping a harmonious relationship with their colleagues. In this way, this hypothesis was supported.

8. DISCUSSION

In this section, we reflect on the findings in [Chapter 7](#) and spread out the topic concerning the cultural factors which impede agile implementation and the usage of the results. Besides, limitations are presented in the end.

8.1 Does Chinese Traditional Culture Really Impede Agile Implementation?

For the decades, ASD has been favoured by Chinese IT-related companies. In addition to the group psychology, there is one statement that companies are using pseudo ASD methods. In the blog of Shore (2008), he used “pseudo agile” for the first time. In China, practitioners are also aware of this issue. Regarding the questions posted on Zhihu¹⁰, we have collected many complaints about pseudo agile:

- Extreme Programming: 8 hours per workday is far from enough. In the view of the leaders, employees should get off work after 9 p.m.
- Welcome Changes: Once developers refuse unreasonable changes, product managers would complain that the developers do not know what agile is.
- Iterative Development: After the product is developed, it should be released as soon as possible. We can repair it and release a new version if our customers find problems.
- Test-Driven: *“I just clicked the button, and it can jump to another page. There should be no bug and let’s release it!”*
- End of the Sprint: *“The project is to be delivered tomorrow. Let’s work all night and finish this Sprint!”*
- Sprint Retrospective: *“There was a new release yesterday. Let’s keep the phones available in case of any emergency fix.”*
- Pair Programming: *“It must be a joke. Why do two developers take turns to program? Can I pay them two for one developer?”*

These answers might be strange, but it did happen in many companies. The phenomenon does exist: many companies exploit workforce euphemistically, in the name of implementing agile software development. The initial purpose of ASD was to satisfy the customer requirements in stages. However, it has evolved as a way to change requirements arbitrarily and persuade customers to agree on the project delay due to more iterations. In a word, many Chinese companies have misused agile software development, which only imitates its activities and practices but ignores its values and principles.

Besides the conflicts based on the Hofstede’s cultural dimensions, we also found that one key factor should be the will of the executive level. The agile theories can be a helping hand in development if the leaders understand its true essence and implement agile methods truly. On the contrary, agile can be a gimmick to deceive customers and exploit employees.

¹⁰ A Chinese Quora-like website. See: www.zhihu.com

8.2 How to Use the Results?

This project is built on a firm foundation, which is the theoretical framework, and validated by the four different scale Chinese companies. The results of the project confirm the relationship between agile theories (agile values and principles) Chinese cultural factors based on the Hofstede's cultural dimensions, and the research has provided empirical evidence. Here we can answer the SQ2, which explores the effects of the identified relationship in adopting ASD methods. Target audiences are classified into three groups, regarding their roles.

For the leaders of the Chinese IT-related companies, this project offers advice on managing development teams and actually using the "agile" concept. For example, avoiding stiff hierarchies helps better communication and empower execution of the teams.

For the software engineers of the Chinese IT-related companies, this project provides a reference when they have to handle the changing requirements. Not only readjusting the priorities of the requirements is necessary, but also conveying the changes to the developers in a tuneful way.

For the researchers who also have interests in this topic, the project links agile development theories with Chinese cultural factors, which is not further investigated. The empirical evidence on how agile theories interplay with Chinese culture can contribute to the design of a localised adaptive software development methodology.

8.3 Limitations

Due to the data of the research is collected via a subjective way, the findings may be biased. The interviewees' feedbacks were guided by their personal values. Personal values are abstract desirable goals. Personal values serve as guiding principles in people's lives, and also serve as standards or criteria that provide social justification for choices and behaviours (Rokeach, 1973). What makes value important is that the influence on people's behaviours. Roccas and Sagiv (2010) summarised the impact of personal values on behaviours: values affect the attractiveness of alternative courses of action; values affect the valence of a behaviour, which in turn makes it more likely that individuals will engage in that behaviour (Feather, 1995). In this way, people are likely to act in ways that promote the attainment of their important values and refrain from acting in ways that block it (Sagiv & Schwartz, 1995). For our research, we can guarantee that our interviewees are honest to a greater extent, as the companies we approached were glad to participate in the research, and we specified the results should be anonymous. Besides, we approached the companies by our relative network, which meant that the interviewees should try to offer their opinions in a justice way. However, the values of the interviewees cannot be measured, which leads to the biased problem. If their values are in accordance with the agile spirits, they definitely will offer positive feedback towards the impact of Chinese culture on agile implementation. On the contrary, if the interviewee is influenced by the waterfall approaches and is taught that waterfall approaches are appropriate, we may receive different feedback.

Another limitation is that the sample of the cases is relatively small. The results of four groups of semi-structured interviews can be representative, but overgeneralisations are also what we try to avoid. In further research, the bigger sample should be considered.

9. CONCLUSION AND FUTURE WORK

In this section, the conclusion of the thesis is presented, and the research questions will be answered. Besides, future work will be summarised.

9.1 Conclusion

In this thesis, the author investigated the relationship between agile theories and Chinese cultural factors. A literature review was conducted in the beginning to form a theoretical framework of the project. As the foundation of the research, the framework contains the agile knowledge and Chinese culture analysis based on Hofstede's cultural dimensions. Combining the agile theories and Chinese culture, we made several assumptions corresponding to the cultural dimensions of Hofstede. Next, semi-structured interviews were conducted in four Chinese IT-related companies with different scale. Their software development methods were analysed, and their attitudes towards the impact of Chinese culture on agile implementation were organised. By comparing the results with our established hypotheses, the results of the research were concluded.

For the SQ1, the beneficial/impeding cultural factors for adopting ASD approaches, we got our conclusions from the actual feedback of the interviewees. By categorising the cultural factors into Hofstede's five cultural dimensions, we found that Chinese cultural factors boost the implementation of ASD.

1. China's high PDI index conforms to agile theories.
2. China's low UAI index conforms to agile theories.
3. China's high MAS index conforms to agile theories.
4. China's low IDV index partially conforms to agile theories.
5. China's high LTO index conforms to agile theories.

For the second, third, and fifth conclusions, the results of the interviews corresponded to our predefined hypotheses, which were derived from the agile theories. Agile methodologies welcome changes, has a clear goal which is delivering working software frequently, and promote sustainable development. The conclusions conform to the UAI, MAS and LTO statement.

For the PDI and IDV statements, quite the opposite, Chinese practitioners believed collectivism culture could increase the ability of execution of the team. Far from the agile values and principles, hierarchies and collectivism help development teams work in an efficient and harmonious environment.

Besides the results we found from the interviews, we also got inspired by the discussion of the Internet forums. Leaders bear heavy responsibilities on their shoulders, who should truly know what agile is and prevent the happen of pseudo agile.

In response to our MRQ, "how Chinese cultural factors influence the adoption of ASD approaches within Chinese SPOs", we have identified the cultural factors which have an impact on agile adoption in China. The conclusions we drew were that the status quo of Chinese culture could be basically consistent with the requirements from agile theories. For the high PDI index and low IDV index, it is abnormal compared to our established hypotheses. However, it was found that both of the aforementioned cultural dimensions have no negative

effects on agile adoption. Based on the statement of the interviewees, we found hierarchies were strictly followed by the employees within the organisations, and hierarchies were regarded as a beneficial factor for the development. On the other hand, collectivist culture has deeply rooted in the hearts of Chinese. Compared to outstanding individuals, leaders participating in our research would favour harmonious and tight development teams.

9.2 Future Work

The impact of National culture on ASD has been studied by many researchers. When it comes to the specific country, the findings become less. This time we narrow our scope and focus on Chinese culture. Due to the complicated tendency of the requirements from the customers, agile software development can still be a popular topic. This paper leads to a new perspective to investigate the relationship between Chinese culture and agile theories. Regarding the customised development methods of the participants and the conclusions, there are several opportunities for future research:

1. The same study can be done in similar East Asian countries.
2. A localised software development method can be expected considering the PDDs of the participants. Method fragments can be modified based on existing agile methods like Scrum or XP, to better boost the development.

REFERENCES

- 2016年中国软件行业发展现状分析及行业竞争现状分析. (2017). Retrieved February 10, 2018, from <http://www.chyxx.com/industry/201701/489339.html>
- Adeoye, B. (2014). *Effects of information capitalism and globalization on teaching and learning*. Pennsylvania: Information Science Reference.
- Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., ... Thomas, D. (2001a). Manifesto for Agile Software Development, 2–3. Retrieved from <https://www.researchgate.net/file.PostFileLoader.html?id=57d055b593553b11467ddd59&assetKey=AS%3A403742915612673%401473271220194>
- Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., ... Thomas, D. (2001b). Manifesto for Agile Software Development, 2–3.
- Bhawuk, D. P. S. (2001). Evolution of culture assimilators: toward theory-based assimilators. *International Journal of Intercultural Relations*, 25(2), 141–163. [https://doi.org/10.1016/S0147-1767\(00\)00048-1](https://doi.org/10.1016/S0147-1767(00)00048-1)
- Bian, Y., & Ang, S. (1997). Guanxi networks and job mobility in China and Singapore. *Social Forces*, 75(3), 981–1005. <https://doi.org/10.1093/sf/75.3.981>
- Bloom, H., Calori, R., & De Woot, P. (1994). *Euromanagement: A new style for the global market*. Kogan Page Publishers.
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30–39. <https://doi.org/10.1109/MS.2005.129>
- Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, 38(4 SPEC. ISS.), 275–280. [https://doi.org/10.1016/0950-5849\(95\)01059-9](https://doi.org/10.1016/0950-5849(95)01059-9)
- Brown, J. D. (2000). What is construct validity? Retrieved May 9, 2018, from http://hosted.jalt.org/test/bro_8.htm
- Bryman, A., & Bell, E. (2015). Business Research Methods. *Business Research Methods*, 629–630. <https://doi.org/10.1016/B978-0-12-387000-1.01001-9>
- Campbell, M. A. (2016). Saying Goodbye to Bimodal IT. Retrieved from <https://www.cioinsight.com/it-management/innovation/saying-goodbye-to-bimodal-it.html>
- Changjian, X. (2017). *Personal communication*.
- Chen, Guodong; Luo, S. (2011). Improvement and Application of Scrum Agile Software Development Method in Practice.
- Chen, M. (2004). *Asian management systems: Chinese, Japanese and Korean styles of business*. Cengage Learning EMEA.
- Cragg, C. (1995). *The new taipans a vital source book on the people and business of the pacific rim*.
- Dong, X. (2017). *personal communication*.
- Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile Modeling, Agile Software Development, and Extreme Programming. *Journal of Database Management*. <https://doi.org/10.4018/jdm.2005100105>
- Eriksson, J. (2015). Analyzing potential barriers of Agile adoption in Chinese software development organizations.
- Fan, Y. (2000). A classification of Chinese culture. *Cross Cultural Management: An International Journal*, 7(2), 3–10. <https://doi.org/10.1108/13527600010797057>

- Feather, N. T. (1995). Values, valences, and choice: The influences of values on the perceived attractiveness and choice of alternatives. *Journal of Personality and Social Psychology*, 68(6), 1135.
- Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. *Software Development*, 9–14. https://doi.org/10.1007/978-3-319-10157-6_3
- Fraser, S., Abrahamsson, P., Robert, B., Eckstein, J., Kruchten, P., Mancl, D., & Wild, W. (2008). Culture and Agile: Challenges and Synergies. In *Agile Processes in Software Engineering and Extreme Programming* (pp. 251–255).
- Fredheim, L. E. (2013). *When is Agile development a bad idea? Do people switch to Agile then BACK to waterfall or some other method?* Retrieved from <https://www.quora.com/When-is-Agile-development-a-bad-idea-Do-people-switch-to-Agile-then-BACK-to-waterfall-or-some-other-method>
- Fukuyama, F. (1995). *Trust: The social virtues and the creation of prosperity*. Free Press Paperbacks.
- Fylan, F. (2005). Semi-structured interviewing. In *A handbook of research methods for clinical and health psychology* (pp. 65–78). Oxford University Press.
- Gallupe, R. B., & Tan, F. B. (1999). A research manifesto for global information management. *Journal of Global Information Management (JGIM)*, 7(3), 5–18.
- Gao, G. (1998). An initial analysis of the effects of face and concern for OTHER in chinese interpersonal communication. *International Journal of Intercultural Relations*, 22(4), 467–482. [https://doi.org/10.1016/S0147-1767\(98\)00019-4](https://doi.org/10.1016/S0147-1767(98)00019-4)
- Hall, E. T., & Hall, M. R. (1989). *Understanding cultural differences*. Intercultural press.
- Hampden-Turner, C., & Trompenaars, A. (1993). *The seven cultures of capitalism: Value systems for creating wealth in the United States, Japan, Germany, France, Britain, Sweden, and the Netherlands*. Doubleday.
- Hao, W., & Young, D. (2000). Drinking patterns and problems in China. *Journal of Substance Use*, 5(1), 71–78. <https://doi.org/10.3109/14659890009053065>
- Highsmith, J. (2002). What is Agile Software Development? *Crosstalk: Journal of Defense Software Engineering*, 4–9.
- Hofstede, G. (1991). Cultures and Organizations : Software of the Mind. *Reviews on Cultural Phenomena*, 38(1), 132–134.
- Hofstede, G. (2011). Dimensionalizing Cultures: The Hofstede Model in Context. *Online Readings in Psychology and Culture*, 2(1), 1–26. <https://doi.org/10.9707/2307-0919.1014>
- Horlach, B., & Drews, P. (2016). Bimodal IT : Business-IT alignment in the age of digital transformation Bimodal IT : Business-IT Alignment in the Age of Digital Transformation, (March).
- Hwang, K. (1997). Guanxi and Mientze : Conflict Resolution in Chinese Society. *Communication Studies*, (1978), 17–38.
- Ingmar, A. H., Bradley, K., & Benlian. (2017). Options for Transforming the IT Function Using Bimodal IT (2017). *MIS Quarterly Executive*, 16(2), 101–120. Retrieved from <http://misqe.org/ojs2/index.php/misqe/article/viewFile/766/463>
- Jiawei, Z. (2018). *Personal communication*.
- Jones, M. L. (2007). Hofstede - culturally questionable? *Oxford Business & Economics Conference*, 24–26. <https://doi.org/978-0-9742114-7-3>
- Kruchten, P. (2011). Agile’s Teenage Crisis? Retrieved April 5, 2018, from <https://www.infoq.com/articles/agile-teenage-crisis>

- Lee, Y. C., Lin, Y. C., Huang, C. L., & Fredrickson, B. L. (2013). The Construct and Measurement of Peace of Mind. *Journal of Happiness Studies*, 14(2), 571–590. <https://doi.org/10.1007/s10902-012-9343-5>
- Lessem, R., & Neubauer, F.-F. (1994). *European management systems: towards unity out of cultural diversity*. McGraw-Hill Book Co Ltd.
- Lewis, R. D. (1992). *Finland: Cultural lone wolf: Consequences in international business*. Transcreen Publications.
- Lewis, R. D. (2006). When cultures collide: leading across cultures: a major new edition of the global guide. *Nicholas Brealey International*.
- Lin, C. (2008). Organizational Studies Demystifying the Chameleonic Nature of Chinese Leadership, 303–321. <https://doi.org/10.1177/1548051808315552>
- Minkov, M. (2007). *What makes us different and similar: a new interpretation of the world values and other cross-cultural data*.
- Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*, 29(9), 972–980. <https://doi.org/10.1108/02656711211272863>
- Mitchell, S. M., & Seaman, C. B. (2009). A Comparison of Software Cost , Duration , and Quality for Waterfall vs . Iterative and Incremental Development : A Systematic Review, (September 2007), 511–515.
- Morden, T. (1999). Models of national culture—a management review. *Cross Cultural Management: An International Journal*, 6(1), 19–44.
- Muhammad, H., & Shaiq, A. (2011). Why not everybody loves Hofstede ? What are the alternative approaches to study of culture ?, 3(6), 101–112.
- Myers, M. D., & Tan, F. B. (2002). Beyond Models of National Culture in Information Systems Research, 10(2), 1–19.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72–78. <https://doi.org/10.1145/1060710.1060712>
- Newman, K. L., & Nollen, S. D. (1996). Culture and Congruence : The Fit between Management Practices and National Culture. *Palgrave Macmillan Journals*, 27(4), 753–779. <https://doi.org/10.1057/Tel>
- Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 20(5), 449–480. <https://doi.org/10.1111/j.1365-2575.2007.00259.x>
- Ramesh, B., Cao, L., Kim, J., Mohan, K., & James, T. L. (2017). Conflicts and complements between eastern cultures and agile methods: An empirical investigation. *European Journal of Information Systems*, 26(2), 206–235. <https://doi.org/10.1057/s41303-016-0023-0>
- Roccas, S., & Sagiv, L. (2010). Personal Values and Behavior: Taking the Cultural Context into Account. *Social and Personality Psychology Compass*, 4(1), 30–41. <https://doi.org/10.1111/j.1751-9004.2009.00234.x>
- Rokeach, M. (1973). *The nature of human values*. Free press.
- S1314857. (2014). Agile Software Development in China. Retrieved February 10, 2018, from <https://blog.inf.ed.ac.uk/sapm/2014/02/14/agile-software-development-in-china/>

- Sagiv, L., & Schwartz, S. H. (1995). Value Priorities and Readiness for Out-Group Social Contact. *Journal of Personality and Social Psychology*, 69(3), 437–448.
<https://doi.org/10.1037/0022-3514.69.3.437>
- Schwaber, K., & Beedle, M. (2001). Agile Software Development with Scrum.
<https://doi.org/10.1109/2.947100>
- Seagrave, S. (1995). *Lords of the rim: The invisible empire of the overseas Chinese*. Putnam's Sons New York.
- Shore, J. (2008). The Decline and Fall of Agile. Retrieved June 16, 2018, from
<http://www.jamesshore.com/Blog/The-Divide-and-Fall-of-Agile.html>
- Society, I., & Foresters, T. (2016). Agile Software Development : the people factor, (August), 131–133.
- Software Development in China. (2006). Retrieved February 10, 2018, from
<http://www.bcs.org/content/conWebDoc/3757>
- Søndergaard, M. (1994). Research Note: Hofstede's Consequences: A Study of Reviews, Citations and Replications. *International Journal of Cross Cultural Management*, (May).
- Stettina, C. J., & Heijstek, W. (2011). Necessary and neglected?: an empirical study of internal documentation in agile software development teams. *Proceedings of the 29th ACM International Conference on Design of Communication*, 159–166.
<https://doi.org/10.1145/2038476.2038509>
- Straub, D., Boudreau, M.-C., & Gefen, D. (2004). Validation guidelines for IS positivist research. *Communications of the Association for Information Systems*, 13(24), 380–427.
<https://doi.org/Article>
- Sutharsahn, A., & Maj, S. . (2011). Enhancing Agile Methods for Multi-cultural Software Project Teams, 5(1).
- Takeuchi, H., & Nonaka, I. (2007). The New New Product Development Game, 137–147.
- The Chinese Culture Connection. (1987). Chinese values and the search for culture-free dimensions of culture. *Chinese Values and the Search for Culture-Free Dimensions of Culture*, 18(2), 143–164.
- Thosmas W. Dunfee, D. E. W. (2001). Is Guanxi Ethical A Normative Analysis of Doing Business in China. *Journal of Business Ethics*, 32(3), 191–204. Retrieved from
<http://www.jstor.org/stable/25074568>
- Triandis, H. C. (1995). *Individualism & collectivism*. Boulder, CO, US: Westview Press. Retrieved from <http://psycnet.apa.org.proxy.library.uu.nl/record/1995-97791-000>
- Triandis, H. C., & Gelfand, M. J. (2012). A theory of individualism and collectivism. *P. A. M, Van Lange*, (Eds.), 498–520.
- van de Weerd, I., & Brinkkemper, S. (2008). Meta-Modeling for Situational Analysis and Design Methods. *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, 35–54. <https://doi.org/10.4018/978-1-59904-887-1>
- Wang, X. (2018). *Personal communication*.
- Watts, S. (2018). What is Bimodal IT? The State of Bimodal IT in 2018. Retrieved from
<http://www.bmc.com/blogs/bimodal-it/>
- Webster, J., & Watson, R. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *Management Information Systems Quarterly*, 26(2).
<https://doi.org/10.1.1.104.6570>
- What is Bimodal IT? (2014). Retrieved from <https://www.gartner.com/it-glossary/bimodal/>
- Wieringa, R. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer Berlin Heidelberg. <https://doi.org/10.1145/1810295.1810446>

- Williams, L., & Cockburn, A. (2003). Agile Software Development: It's about Feedback and Change. *Computer*, 36.
- Wilson, J. (2014). *Essentials of Business Research: A Guide to Doing Your Research Project - Jonathan Wilson - Google 图书*. SAGE Publications. Retrieved from [https://books.google.com.hk/books?hl=zh-CN&lr=&id=0h6VAgAAQBAJ&oi=fnd&pg=PP1&dq=Essentials+of+Business+Research:+A+Guide+to+Doing+Your+Research+Project&ots=N1lwyvRn67&sig=MccKDe7uFkwSKYtb_x8Gf9BcK1E&redir_esc=y#v=onepage&q=Essentials of Business Res](https://books.google.com.hk/books?hl=zh-CN&lr=&id=0h6VAgAAQBAJ&oi=fnd&pg=PP1&dq=Essentials+of+Business+Research:+A+Guide+to+Doing+Your+Research+Project&ots=N1lwyvRn67&sig=MccKDe7uFkwSKYtb_x8Gf9BcK1E&redir_esc=y#v=onepage&q=Essentials+of+Business+Res)
- Zhang, Y., & Shen, H. L. (2010). An Exploratory Study of Organisational Adaptation to Agile Project Management. *Umu.Diva-Portal.Org*, 101. Retrieved from <http://umu.diva-portal.org/smash/get/diva2:397150/FULLTEXT01>

APPENDICES

Appendix A. Interview Protocol

Appendix A-1: English Version



Universiteit Utrecht

My name is Xiuyu Qiao, a Master student majoring in Business Informatics at Utrecht University. Now I am doing my graduate project: *A Novel Solution to Improve Agile Software Engineering with Cultural Sensitive Aspects in a Chinese Context*. To identify potential barriers of agile adoption in China and generate an improved development method which better suits the state-of-art in China, the interview today is centered with agile adoption in China. With your permission, the interview will be recorded. Only researchers on the project will have access to the tapes which will eventually be destroyed after they are transcribed. Essentially, we state that: (1) all information will be held confidential, (2) your participation is voluntary and you may stop at any time if you feel uncomfortable, and (3) we do not intend to inflict any harm. Thank you for agreeing to participate.

We have planned this interview to last no longer than a half-hour. During this time, we have several questions that we would like to cover. If time begins to run short, it may be necessary to interrupt you in order to push ahead and complete this line of questioning.

Date: _____ Start time: _____ End time: _____

Interviewer: Xiuyu Qiao (x.qiao@students.uu.nl)

1st supervisor: Sietse Overbeek (s.j.overbeek@uu.nl)

2nd supervisor: Sergio España Cubillo (s.espana@uu.nl)

A. Interviewee background

1. How old are you?
2. What is your highest degree?
3. What was your field of study?
4. What the size of the company you current work is?
5. How long have you been working in this area?
6. Have you ever been involved in agile software development?
7. Which agile method(s) have you adopted?

B. Project information

1. Can you please describe the nature of your project?

Probe: What type of the customers of the product are?
(Domestic/international/mixed)

2. What was your role in the project?

3. How many people were involved in the project?

Probe: How many people can work efficiently in a team best?

4. What was the length of the project?

Probe: If the project has not ended, please estimate it.

5. How often does an iteration last in your project?

Probe: Which role is responsible for defining the length and the content of the iteration?

6. Have any tools been used in the project?

Probe: How do you like the tools?

7. Can you please introduce the key development activities (like reflection meeting) and work products produced (like product backlog)?

C. Agile value

1. Have your project covered any agile practices (i.e. easy access to expert users, frequent builds, time boxed development cycles, pair programming, publicly displayed information, reflective improvement, frequent & informal communication, self-organizing team)?

Probe: Can you please share your opinion towards agile practices?

2. To what extent do you agree with the agile values (i.e. individual & interaction over tools and processes, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan)?

D. Cultural sensitive aspects

1. Is there a strict hierarchy in your organization?

Probe: If yes, do you think it is suitable/harmful for the iterative processes?

2. Would you sacrifice your leisure time to produce the product?
3. Would you point out the mistakes of your colleagues in public?
4. Faced with conflicts, would you prefer avoiding them and solving them later, or finding the causes and the solutions of the problems in no time?
5. Would you decline unreasonable requests from your colleagues/leader if you cannot fulfill them on your own / on time?
6. What is your opinion towards changing requirements from customers?
7. Do you think it is easy to interpret the inner thought of the colleagues in China?
8. Do you think the Guanxi Network can boost the development of the product?

E. Reflective section

- a) In your opinion, what should be the core of agile software development?
- b) Do you think Chinese culture is correlated with agile adoption in China?

Probe: To what extent?

- c) How do you like agile software development methods?

Probe: Can you please list its pros and cons?

Thank you for your time!

Appendix A-2: Chinese Version



Universiteit Utrecht

您好，我是乔秀宇，一名来自荷兰乌特勒支大学商务信息学的硕士研究生。现在我正在进行我的硕士论文：敏捷软件工程在中国文化因素影响下的新实行方法。为了发现敏捷开发在中国实施的潜在障碍和设计更好适应中国现状的开发方法，今天的采访围绕敏捷开发在中国实施来展开。在您的允许下，采访将会被录音。只有参与本项目的研究者有权接触本录音，且录音将会在听译整理后被销毁。在此我们声明：（1）所有的信息将会保密；（2）您的参与为自愿，且若您对采访感到不适，您可以在采访过程中任何时间停止；（3）我们不会造成任何伤害。感谢您的参与。

我们计划本次采访持续半小时左右。在这期间，我们有规定的问题数目。如果时间预算紧张，请理解您的回答将会被打断，以保证采访能够在规定时间内完成。

日期：_____ 开始时间：_____ 结束时间：_____

采访者：乔秀宇 (x.qiao@students.uu.nl)

第一导师：Sietse Overbeek (s.j.overbeek@uu.nl)

第二导师：Sergio España Cubillo (s.espana@uu.nl)

A. 受访者背景

1. 请问您的年龄为？
2. 请问您的学历是？
3. 请问您的专业是？
4. 请问您现在工作单位的员工数量为？
5. 您从事现在这个行业的年数为？
6. 您是否参与/了解过敏捷开发？
7. 您所参与的敏捷开发方法为？

B. 项目信息

1. 能请您描述下您参与的敏捷开发项目情况吗？

讨论：您工作开发的产品面向的客户类型为？（本国/国际/混合）

2. 您在团队中的角色为？

3. 请问您开发团队的人数为？

讨论：您认为团队最佳人数多少为宜？

4. 您项目的时间持续了多久？

讨论：如果项目还没结束，请预估。

5. 在您的项目里，一个迭代通常周期为多久？

讨论：团队中哪个角色负责定义迭代周期和迭代内容？

6. 您在敏捷开发中有使用过哪些工具？

讨论：您对这些工具有何评价？

7. 能请您介绍下您参与的敏捷开发中重要的开发活动（例如反思会议）和工作交付品（例如产品 backlog）吗？

C. 敏捷价值观

1. 您参与的敏捷开发项目有覆盖相关的敏捷业务惯例吗（即：与顾客方便的接触；频繁的产品发布；固定周期的开发循环；结对编程；公开研发信息；反思提升；频繁且非正式的交流；自组织团队）？

讨论：能请您对这些敏捷业务惯例发表自己的看法吗？

2. 您对敏捷价值观的认可度如何（个体和互动高于流程和工具；工作的软件高于详尽的文档；客户合作高于合同谈判；响应变化高于遵循计划）？

D. 文化敏感因素

1. 您所在的公司有严格的等级制度吗？

讨论：如果有的话，您认为它对敏捷开发的迭代流程有好处/坏处？

2. 您愿意牺牲自己的业余时间来完成产品的开发吗？
3. 您会当众指出同事在工作中的错误吗？
4. 面对冲突，您通常偏向于回避冲突再在之后进行解决，还是立刻找到冲突的原因所在并且给出解决方案？
5. 如果您不能完成被分配的任务，您会拒绝来自同事/领导的不合理需求吗？
6. 您对客户不断变化的需求有何看法？

讨论：您对不断变更的需求持接纳/抵触心理？

7. 您认为在中国是否容易去读懂同事内心的想法？
8. 您认为关系网络是否能够促进产品的开发进度？

E. 回顾部分

- a) 您认为敏捷开发的核心是什么？
- b) 您认为中国的文化（例如：尊卑有序/集体主义/中庸）是否会影响敏捷开发在中国的实施？
- c) 您对敏捷开发有何见解？

讨论：相对于瀑布模型，您能简单列举敏捷开发的优劣吗？

感谢您的参与！

Appendix B. Company Profiles

B.1 Alpha Software Company

Located in Shanghai, one of the first-tier cities in China, ASC has at around 1000 FTEs, which means that ASC is a large enterprise in China. ASC specialises in enterprise management software since 2000, and it has focused on omnichannel retailing for five years. Its product portfolio includes ERP system, supply-chain management system, CRM system, WMS system, and online shopping system.

The department which accepted our interviews was the shopping mall department. For the product, online-shopping system, it was designed for enterprises which desire to develop their online business in the boom era of net purchase. As a fast-developing domain, Chinese E-commerce has been tied to Taobao due to Taobao's vast contribution to Chinese E-commerce. However, with the growth of settled merchants in Taobao, the merchants desire to develop their brands and bring in more user traffic to their official websites. Thus, the pain points have motivated the need for establishing a customised official shopping website.

The first interviewee was the director of the department, who also played a role as a product manager. He had been working in the domain for 13 years. He had a college degree, and his major was E-commerce.

The second interviewee was the tester. She had been working in the area for 3 years. She had a bachelor degree and her major was electronic information and technology.

B.2 Beta Software Company

BSC was established in 2000 and is still a small enterprise which has less than 20 employees. It is located in a provincial capital, which is a second-tier city. In the beginning, BSC was only focused on developing ERP systems for machine factories; later it also involved in the environmental protection business offered by local government.

The director of BSC accepted our requests for interviews in May 2018. The interviews were conducted offline. For the director, he had a college degree, and had been working in this industry for 20 years. The major he had was environmental engineering.

The developer also had a college degree and his major was software engineering. He had been working in this domain for two years.

B.3 Gemma Software Company

Gamma Software Company (GSC) is a state-owned enterprise which focuses on the power sector. GSC has more than 20000 employees, and its business area has covered most of the provinces in China. Compared to private companies, its secrecy rules are stricter. We tried to invite the employees from GSC to accept an interview, and a developer participated in our project. The participant belongs to the IT department, which is located in Beijing, one of the first-tier cities in China. This department is responsible for developing software used by internal employees and cooperative partners.

The senior developer had been working in this area for four years. He had a bachelor degree and his major was computer science and technology. By that time, it was his fourth job.

B.4 Delta Software Company




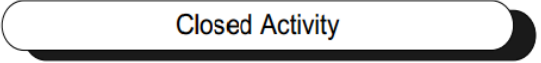
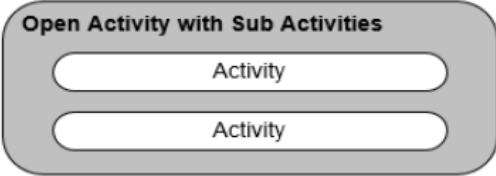





Delta Software Company (DSC) participated in our research project in May 2018. DSC is a platform linking between Bed and Breakfast (BNB) owners and tourists, and its major target areas are Chinese mainland, Taiwan, Japan and Korea. DSC has around 200 employees and is a medium-sized enterprise. Located in Shanghai, the development team has 20 staffs, and the roles include product managers and developers. A senior developer and a product intern were designated to participate in our project.



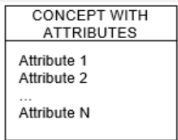
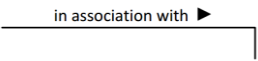

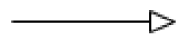

The developer had been working in this area for three years, and he had not changed his job. His major was computer science and technology and he had a bachelor degree.

The product intern just started working by that time. His major was software engineering and had not graduated from his university.

Appendix C. PDD Notation

Table A-1: the notations and descriptions of PDD (van de Weerd & Brinkkemper, 2008)

Notations	Descriptions
Process View	
	Initial state
	Standard activity: An activity that contains no further activities.
	Open activity: An activity that consists of a collection of sub-activities which are expanded in the same diagram or another diagram.
	Closed activity: A complex activity where its sub-activities are not expanded since it is known or not relevant in the specific context.
	Open activity with sub-activities. An activity that consists of a collection of sub-activities which are depicted inside it. This notation also is used to describe a set of unordered activities.
	Branch: A state where the process is split into two or more routes based on specific criteria.
	Transition: A notation that explains the flow of the process.
	Forking and Joining. Forking and Joining use the same notation. Forking is used to start a set of concurrent activities and Joining is used as the end state of parallelism.
	End state
Deliverable View	
	Standard concept: A concept that contains no further concepts.

	<p>Open concept: A concept that consists of an aggregate of other concepts which are shown in the same or another diagram.</p>
	<p>Closed concept: A complex concept where its sub-concepts are not expanded in the same diagram since it is not relevant in the specific context.</p>
	<p>Concept with attributes: A concept (can be a form of standard, open, or closed concept) which describes its attributes.</p>
	<p>Relationship: A structural relationship that connects two concepts and specifies how concepts are linked to another.</p>
	<p>Aggregation: A specific type of relationship that represents the relation between a concept containing other concepts.</p>
	<p>Generalization: A relationship between a general concept and more specific concepts.</p>
	<p>Connection: Connecting process to delivered or utilized concepts.</p>

Appendix D. PDD Documentation

Table A-2: table of processes for the PDD of ASC

Activity	Sub-activity	Description
Waterfall process	Define requirement	The product managers are responsible for scoping the requirements for the entire system, with the data gathered from the operating department.
	Design software architecture	The developers are required to design the architecture, based on existing architecture or from scratch.
	Implement software framework	The developers design and implement the basic framework.
Agile process	Review product backlog	The product managers select requirements from the product backlog to prepare for the Sprints.
	Make planning	Sprint backlogs are organized by the product managers and team members can verify the backlogs.
	Assign task	Tasks are assigned to corresponding developers by the director of development and product managers.
	Develop requirement	The developers are required to develop the assigned fragments in required time.
	Check developed module	The testers are supposed to test the deliverables.
	Deliver working software	Working software is to be delivered when the testing is passed.
Release external version		The external version is released. The product managers should write the release document and the testers validate the working software.
Release to partner		Working software is delivered to the partners of ASC.

Table A-3: table of concepts for the PDD of ASC

Concept	Description
PRODUCT MANUAL	A general document which contains the requirements, architecture, testing, release, and operation document.
REQUIREMENT DOCUMENT	A document to describe all the requirements of the software.

ARCHITECTURE DOCUMENT	A document specifying the architecture of the software is written by the developers.
PRODUCT BACKLOG	A backlog which is transferred from the REQUIREMENT DOCUMENT, containing all the requirements of the software.
SPRINT BACKLOG	A backlog for a Sprint, which is the requirements extracted from the PRODUCT BACKLOG.
DEVELOPED MODULE	Modules delivered by the developers.
TEST CASE	The document is written based on the REQUIREMENT DOCUMENT. A guideline when testers verify the DEVELOPED MODULES.
DEMO	Working software integrated by the DEVELOPED MODULES in a Sprint.
RELEASE DOCUMENT	A document is written by the product manager, specifying the requirements added in the Sprint and the information of the bug fixes.
DELIVERABLE PRODUCT	A verified and quality-assured DEMO to be released to the partners.
OPERATION MANUAL	A document teaching users to operate the software.

Table A-4: table of processes for the PDD of BSC

Activity	Sub-activity	Description
Preparation	Define requirement	The product manager is responsible for organising the requirements based on the business of the customers.
	Design the database structure	The product manager and the developers design the database together.
	Design software architecture	The developers design the architecture.
	Divide the workflow	The product manager divides the development into several stages based on the actual workflow of the customers.
Development	Select the requirement	Requirements of the software is selected iteratively in each iteration.
	Assign task	The product manager assigns the modules to the developers.
	Develop requirement	The developers develop the assigned modules.
	Test the module	Team members test the delivered modules.
	Deliver working software	The working software is delivered once the software is tested successfully.
Release the product		The product is released to the customers.

Table A-5: table of concepts for the PDD of BSC

Concept	Description
----------------	--------------------

REQUIREMENT DOCUMENT	A general description which specifies the requirements, the design of the database, the architecture, work distribution, and the product explanation.
DATABASE SPECIFICATION	The design of the database.
SOFTWARE ARCHITECTURE DOCUMENT	The architecture of the software.
WORK DISTRIBUTION TABLE	The document specifying the assignment distribution.
DEVELOPED MODULE	The modules developed by the developers.
TEST REPORT	The document specifying the bugs found during testing.
WORKING SOFTWARE	Working software integrated by the DEVELOPED MODULEs in an iteration.
DELIVERABLE PRODUCT	A verified and quality-assured product to be released to the customers.
PRODUCT DESCRIPTION	A document introducing the product to the users.

Table A-6: the table of processes for the PDD of GSC

Activity	Sub-activity	Description
Preparation	Receive request	The leader of the department receives the requirements organised by the operating department.
	Organise requirement	The leader of the department re-organise the requirements and make them clear for the development.
	Estimate workload	The leader of the department estimates the workload of each requirement.
	Assign task	The leader of the department assigns tasks to developers at one time, while the developers develop the modules in different iterations as planned.
Development	Develop requirement	The developers develop the desired modules.
	Test the module	The testers test the modules.
	Integrate the software	The developers integrate the modules together.
	Test the software	The integrated software is tested by the testers again.
Release the product	Prepare to release the version	The developers package a release when the working software is tested successfully.
		The product is released to the customers.

Table A-7: the table of concepts for the PDD of GSC

Concept	Description
MARKET REQUIREMENT DOCUMENT	The requirements collected from the operating department.

FUNCTIONAL REQUIREMENT DOCUMENT		The requirements re-organised by the leader of the department.
PRODUCTION DOCUMENT	PLANNING	A document specifying the estimation of the workload for each requirement and the corresponding employees to solve the tasks.
DEVELOPMENT DOCUMENT		The document describing how the module was developed, which is written by the developers.
MODULE DOCUMENT	TEST	The document specifying the bugs found during module testing.
INTEGRATED SOFTWARE		Working software integrated by the developed modules in an iteration.
FUNCTIONAL DOCUMENT	TEST	The testing report written in the functional testing phase.
THE RELEASED VERSION		A verified and quality-assured product to be released in the end of an iteration.
THE SOFTWARE	WORKING	A verified and quality-assured product to be released to the customers.
OPERATION MANUAL		A document teaching the customers how to use the software.

Table A-8: the table of processes for the PDD of DSC

Activity	Sub-activity	Description
Preparation	Receive request	The product managers receive the requirements from their operating department.
	Split requirement	The product managers and the developers split the requirements into tasks.
	Assign task	The product managers and the developers assign the tasks to corresponding developers.
Development	Select requirement	The product managers select requirements from the REQUIREMENT DOCUMENT as the requirements to be solved in an iteration.
	Develop requirement	The developers develop the modules.
	Write unit test	The developers write the unit test while coding.
	Test the module	The developers test their module by themselves after coding.
	Integrate the software	When the self-testing is passed, the modules are integrated.
	Test the software	The working software is integrated and tested again.
	Prepare to release the version	The software is packaged to be released.
Release the product		The product is released to the customers.

Table A-9: the table of concepts for the PDD of GSC

Concept	Description
----------------	--------------------

REQUIREMENT DOCUMENT	The requirements re-organised by the product managers.
TASK CHECKLIST	A document interpreting the requirements into specific tasks.
WORK DISTRIBUTION TABLE	The document describing the working distribution.
DEVELOPMENT DOCUMENT	The document recording the ideas while the developers are coding.
MODULE TEST DOCUMENT	The document specifying the bugs found during module testing.
INTEGRATED SOFTWARE	Working software integrated by the developed modules in an iteration.
INTEGRATION TEST DOCUMENT	The testing report written in the integrated testing phase.
THE RELEASED VERSION	A verified and quality-assured version to be released in the end of an iteration.
THE WORKING SOFTWARE	A verified and quality-assured product to be released to the customers.
OPERATION MANUAL	A document teaching the customers how to use the software.