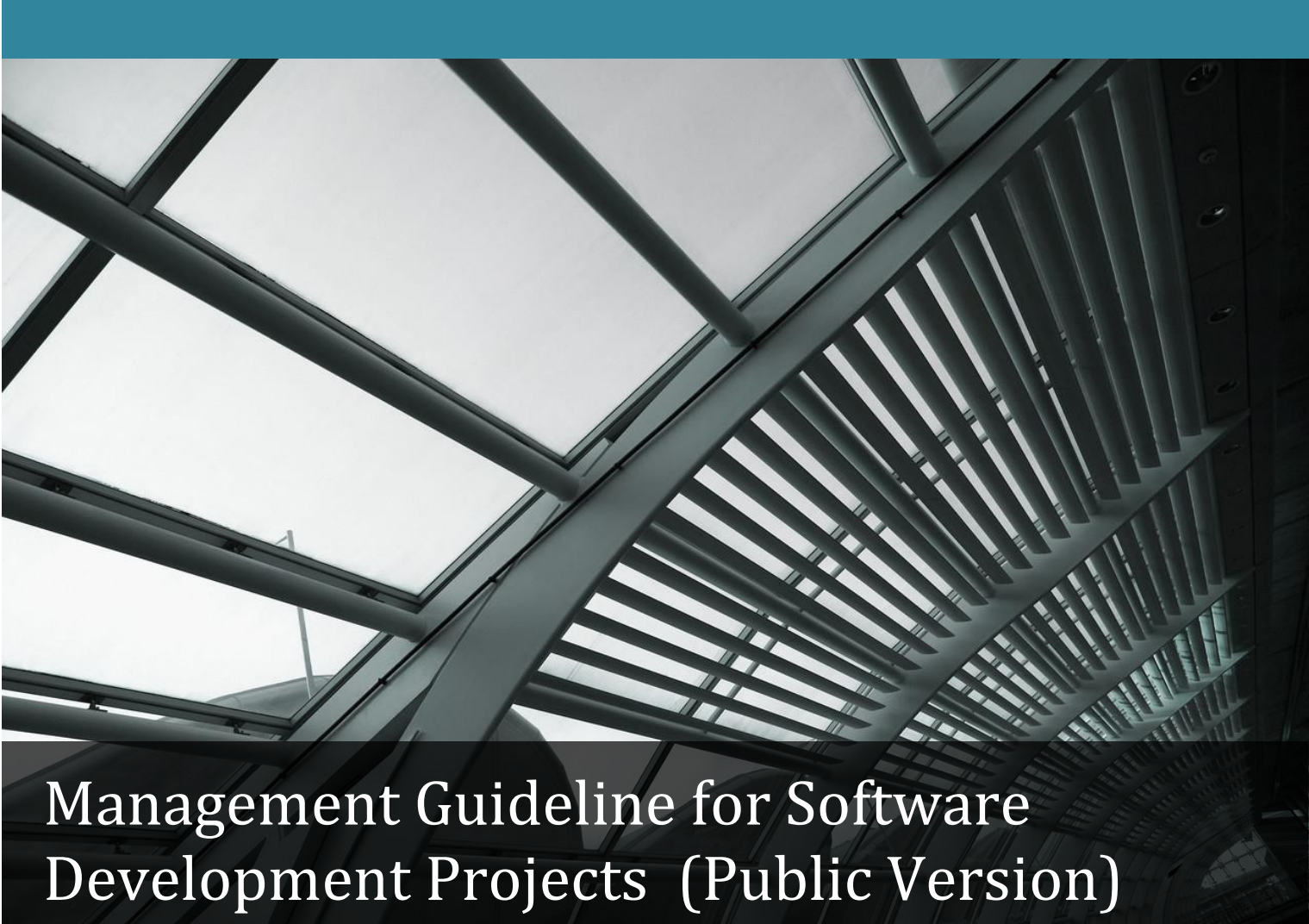




Management Guideline for Software Development Projects (Public Version)

SELIN SENGUR

Management Guideline for Software Development Projects having unclear requirement definition and stakeholders with different sense of urgencies: Lessons learned from a Case Study

Systems Engineering, Policy Analysis and Management

Management Guideline for Software Development Projects

(Public Version)

M.Sc. Thesis
Selin Sengur
4209370

Faculty of Technology, Policy and Management
Delft University of Technology
Jaffalaan 5, 2628 BX Delft, The Netherlands
S.Sengur@student.tudelft.nl

August 2015

Graduation Committee:

Prof.dr.ir. A. Alexander Verbraeck – Delft University of Technology
Dr. M.E. Martijn Warnier – Delft University of Technology
Dr. H.G. Haiko van der Voort – Delft University of Technology
Dr. Ivo Wenzler – Accenture

Contents

Executive Summary.....	ix
Acknowledgements.....	xi
1 Introduction	2
1.1 Research Problem	2
1.2 The Research Gap.....	4
1.3 The Research Goal.....	4
1.4 Summary of the Research Problem and Goal	4
1.5 The Research Question and Sub-questions:	5
1.5.1 Research Question:	5
1.5.2 Research Sub questions:	5
Methodology.....	6
1.5.3 Action Research	6
1.5.4 Literature Survey.....	6
1.5.5 Case Study	7
1.5.6 Interview	8
1.6 The Structure of the Thesis	9
2 Literature Survey.....	12
2.1 LITERATURE SURVEY: Software Development Methodologies	14
2.1.1 What is a Project?	14
2.1.2 Project Management: A systems approach to management.....	14
2.1.3 Critical Success Factors of a project.....	15
2.1.4 Project Management Methodology	15
2.1.5 History of the Software Development Models.....	16
2.1.6 Waterfall model	16
2.1.7 Iterative and Incremental Development (IID) Approach: Agile Model	17
2.1.8 Software Development with Scrum.....	19
2.1.9 Current States of the Art of the Agile Software Development.....	20
2.1.10 Application of Scrum in Large-Scale Global Software Development Projects.....	21
2.1.11 Conclusion.....	23
2.2 Literature II: Stakeholder Analysis and Management.....	24
2.2.1 Introduction	25

2.2.2	Actors and Actor Analyses	25
2.2.3	Multi-issue decision-making	29
2.2.4	Dilemmas faced in a multi-actor decision making environment.....	31
2.2.5	Conclusion.....	32
3	Case Study: PART I - Stakeholder Analysis and Management	34
3.1	Introduction.....	34
3.2	Project Stakeholders	35
3.2.1	The Project Owner: Accenture.....	35
3.2.2	The Software Development Company: MACOMI.....	35
3.2.3	The Client / Contractor	36
3.2.4	Stakeholder Analysis.....	37
3.2.5	Power Relations between the Stakeholders.....	40
3.2.6	Engaging the Stakeholders.....	42
3.2.7	Conclusion.....	43
3.3	Stakeholder Management Methodology I – Two Poles Approach	43
3.3.1	Stakeholder involvement Strategy of Accenture and Scenario Analysis....	43
3.3.2	Results of the Two-poles approach	46
3.4	Stakeholder Management Methodology II – Multi-issue Decision-making	46
3.5	Results and Conclusions	47
4	Case Study: PART II - Software Design and Development.....	50
4.1	Problem Definition of Client and several observations from the organization:	50
4.2	Offer from Accenture/Planning Phase of the Project.....	51
4.3	Capabilities and the Limitations of the Final Delivery:	52
4.4	The Simulation Tool.....	52
4.5	Project Planning and Requirement Analysis	52
4.5.1	Project Planning	53
4.5.2	Requirement Analysis:	53
4.6	Methodology used in the Project.....	54
4.6.1	Iterative and Incremental Development - Agile Approach.....	54
4.6.2	The role of Agile Approach on managing the different sense of urgencies and changing requirements in the case study.....	56
4.7	Mediator:.....	59

4.8	Dilemmas: Negotiated nonsense vs design nonsense and openness vs closeness	59
4.9	Conclusion	60
4.10	Reflection of the Case Study.....	60
4.10.1	Reflection by the software consultant.....	60
4.10.2	Reflection by the Researcher	61
5	Project Management Guideline.....	64
5.1.1	Stakeholder Mapping Technique:.....	66
5.1.2	Which stakeholders to involve: Two-poles approach:	67
5.1.3	Priming strategy:.....	69
5.1.4	Applying multi-issue game in software development project environment having several stakeholders with different sense of urgencies	69
5.1.5	Iterative and incremental approach – Agile Software Development Approach.....	71
5.1.6	Mediator:	72
5.1.7	Dilemmas faced in a software development project with multi-actor environment, having different sense of urgencies.....	73
5.2	Conclusion	76
6	Validation of the Guideline	78
6.1.1	Stakeholder Mapping Technique.....	79
6.1.2	Two-poles Approach	79
6.1.3	Priming Strategy.....	79
6.1.4	Multi-issue decision-making	80
6.1.5	Iterative and incremental approach:	80
6.1.6	Mediator:	80
6.1.7	Dilemmas:	80
6.2	Conclusion	80
7	Conclusion.....	82
7.1	Answers to Research Question:	82
7.2	Main Conclusion	83
7.3	Reflections.....	84
7.4	Further Study.....	85
7.5	Suggestion to Accenture	85
8	Bibliography	87

Executive Summary

The needs of companies are evolving and becoming more and more demanding each year, which result in more complex stakeholder relations. The complexity of the projects increases as well. For instance, stakeholders do not always have the same sense of urgencies and their expectations from the same project might differ. Such variety in demand urges project managers to opt for a new; more broaden management approach that includes more human type of issues as well as hard skills. One area this situation is observed in contemporary projects is software development projects with complex stakeholder environment, where the project managers have responsibility to deliver high quality end delivery at the same time to meet the demands of stakeholders with different sense of urgencies. Main characteristics of a software development project in a complex stakeholder environment are unclear requirement definition and the different sense of urgencies arises from stakeholder complexity. However, current methodologies in the literature started to remain inadequate to meet the contemporary demands of the companies. Literature provides methodologies on how to adapt changing requirement to the software development process such as agile software development approach. Although the literature guides managers on how to deal with unclear requirement definition in software development projects, they do not put a focus on managing different sense of urgencies in such project. On the other hand, there is literature on stakeholder management, particularly different sense of urgencies but those are not applied in a software development environment in the literature. Proposed methodologies are mostly used in political environment or non-software projects. There is no unique methodology that bridges these two fields and applies in a software development project environment.

This study aims for answering the research question *“How can software projects having unclear requirement definitions and stakeholders with different sense of urgencies can be managed successfully?”*

The research searches an answer to this question by analyzing the literature and lessons learned from a case study. The literature is reviewed from two perspectives:

- Methodologies used for managing the software development projects with unclear requirement definition
- The literature used for the stakeholder management techniques, particularly on dealing with different sense of urgencies.

The case study used in this research is about a software development project having unclear requirement definition and stakeholders with different sense of urgencies. As the characteristics of the case study is matching the definition of the project in the research question, the insights taken from the case study become a valuable input for the management guideline. Insights both from the literature and the case study are combined and following methodologies are proposed as a guideline in the light of these insights:

1. **Stakeholder mapping technique:** To identify the power relations between the stakeholders and their characteristics.
2. **Two-poles approach:** Involving the stakeholders that have the most divergent requirements.
3. **Priming strategy:** To make the project more attractive for the reluctant stakeholders by changing the problem perception of a key stakeholder.

4. **Multi-issue decision-making approach:** To involve stakeholders into the decision-making process in software development project.
5. **Iterative and incremental approach:** For the management of the software development tool because of its characteristics of adaptability to the continuously changing requirements.
6. **Mediator:** Find a mediator for an effective communication between the programmers and the client.
7. Make a good balance between negotiation and high quality end product to deal with the dilemmas: “**openness vs closeness**” and “**negotiated nonsense vs design nonsense**”

The research question is answered by this management guideline, which is validated by expert opinion. They confirmed that the proposed methodologies as effective to deal with unclear requirement definition and different sense of urgencies in a software development project.

Acknowledgements

This study has been accomplished under irregular conditions that required long-term persistence and dedication. The motivation was the key factor for me to keep working on the thesis. Therefore, I owe my deep appreciation to many people, who contributed to keep my motivation up and supported me.

My first thanks go to my committee members. They became more than just being supervisors. Their motivation and support is substantial on the success of this thesis. A supportive e-mail that suddenly came from Alexander Verbraeck at a time that I was about to lose my hope on the progress of the thesis was a big milestone to turn back to the studies and to believe in myself on finishing the thesis in the planned time. I appreciate his supportive and encouraging attitude, understanding on long and late meetings because of my job. Additionally, his substantial input on towards which direction the topic of the thesis should go and what are his expectations to achieve a high quality report and how to reach there was invaluable for the progress of my thesis. I appreciate he invested a lot of time and effort on me although he has a busy schedule.

Martijn Wannier, my first supervisor, did not give up believing me to finish my thesis, always gave his support, and was answering my questions even though when he was on holiday. He never demotivated me when I could not make the deadlines, was patient and tried to find solution to overcome the reasons behind it.

Ivo Wenzler, my supervisor from Accenture and the manager of the project that I participated in as an intern last year and used the insights from this project in substantial part of the thesis. I owe most of the knowledge that I have on this thesis to him. First of all, I appreciate that he gave me the chance to work in such a big project, which turn out to be as a unique experience and also rooted my current employment in Accenture. He involved me all the meetings with the client together with the internal ones, which brought me invaluable insights that I could not learn anywhere else. Having the chance of following his way of thinking and approach against problems/situations was a great opportunity. Additionally, his understanding and positive attitude towards my life-changing decision helped me to lower the burdens on my shoulder. Also, he always made time for me for any thesis related questions, provided his input to finish my work even though he has tight schedule. I appreciate them all.

Haiko van der Voort, my second supervisor, thank you very much for your understanding and effort to give me green light just before your vacation. Even though I involved you to the loop very late, you did your best to provide your input.

My thanks are not limited with TU Delft. As I wrote this thesis together with a full time job, I have many understanding managers to support me to get the thesis done in Accenture. First of all, my career counselor, Bert Savelkouls, thank you very much for supporting and encouraging me from the beginning of the thesis process. Your trust and support was a great impulse for me to believe in myself. Joerg Langhorst, managing director of the client project that I am currently working on, thank you for your respect on my thesis work alongside of my job and giving me a not-planned, last minute week off when I needed time to work on my thesis. Mashad Bani-Aman and Amir Majlesi, my managers from my current and previous role in the client project, thank you very much your motivation and flexibility on giving me day off when I needed to work on my thesis.

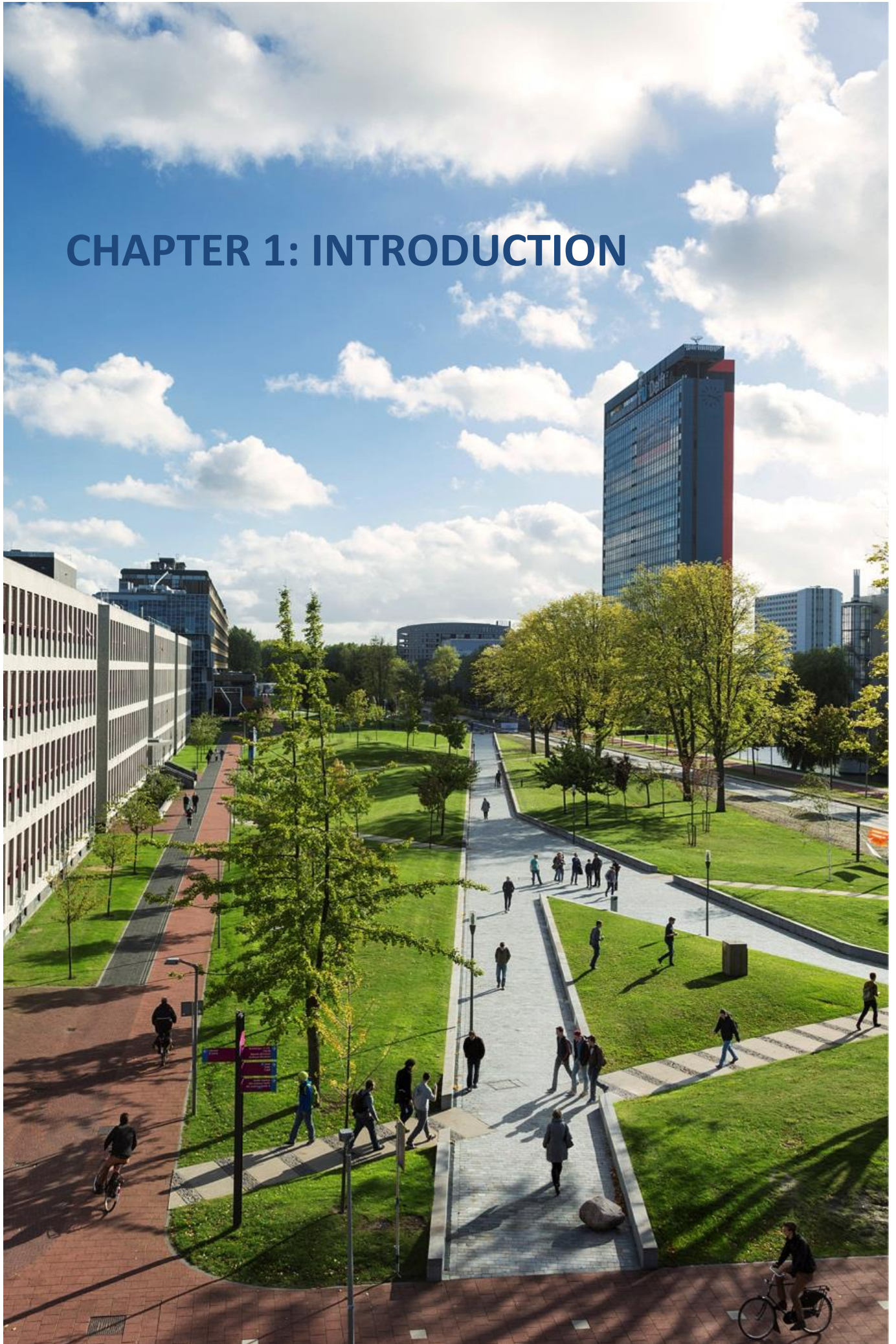
Natasja Ensberg, software development manager, thank you for making time for me and providing your valuable insight to validate my approach in the thesis.

My last special thanks go to all my friends and the family, who have been patiently waiting me to finish my thesis, for their understanding during this difficult period of time. I certainly hindered my responsibilities to them by not calling and not joining their invites or even sometimes not responding to their texts at all. They all motivated me to make me keep the hard work. Thank you all!

Selin Sengur

August 2015

CHAPTER 1: INTRODUCTION



1 Introduction

More and more organizations have been realizing the importance of the human factor in the projects as they are the ones undertaking the project work and have the influence (negative or positive) on the end result. Although traditional project management techniques are significant, they are not always adequate to accomplish a project successfully. Communication between the team members and the entire network has significantly important for the success of the project as it is crucial for the shared understanding of the project and its goals (Ruuska & Vartiainen, 2003) (Strang, 2003). Therefore, understanding the demands of the stakeholders and their power relations as well as the skills/methods to manage them effectively is a vital topic that should not be underestimated (Cowie, 2003). Kliem et. al., (Kliem & Ludin, 1997) support this view in their studies as well. Based on several researches on the literature, it can be concluded that successful project management requires interdisciplinary skills such as interpersonal ability, technical competencies, cognitive aptitude together with the capability of understanding the situation and the people in the project and dynamic integration of the appropriate leadership behavior accordingly (Strang, 2003).

Kloppenborg et. al., (Kloppenborg, 1999) claims that project leadership is the combination of the technical skills and the ability of managing the team. Successful management of the relationships is significant for the stakeholder satisfaction in every phase of the project. The project outcomes are achieved not only by the use of hardware and techniques but also through the people by acknowledging their knowledge and creativity. For this reason, relationship skills (soft skills) should be considered as complementary for the development of the hard skills (Bourne & Walker, 2004). Engaging effective relationships among the team members and the stakeholders is one of the tricky challenges that a project manager encounters as it requires advanced capability of pursuing a balanced approach between the hard and soft skills (Bucero, 2004) (Thamhain, 2004).

Strang et. al., (Strang, 2003), Mantel et. al., (Mantel, JR. & Meredith, 2011), Loo (Loo, 2003) and Belzer (Belzer, 2001) all claims that a more broaden approach that includes human types of issues together with the traditional project management methods is crucial for a successful project management.

This chapter starts with introducing the research problem, research gap and the research goal. Then, the research questions and sub-questions are discussed. The methodology used in the thesis to answer the research question is explained. Finally, the structure of the report is given.

1.1 Research Problem

Traditional project management methods are important to accomplish the elements of project triangle (Figure 1) successfully, which results in performing high quality deliverables. However, many projects result in failure, as there is no available tailor made management method for each project. Every project has its own unique problems and needs. Therefore, the methods that work in one project may not give effective results in one another. Looking at the literature, many researchers pointed out the importance of accomplishing the projects within time, budget and with satisfied stakeholders.

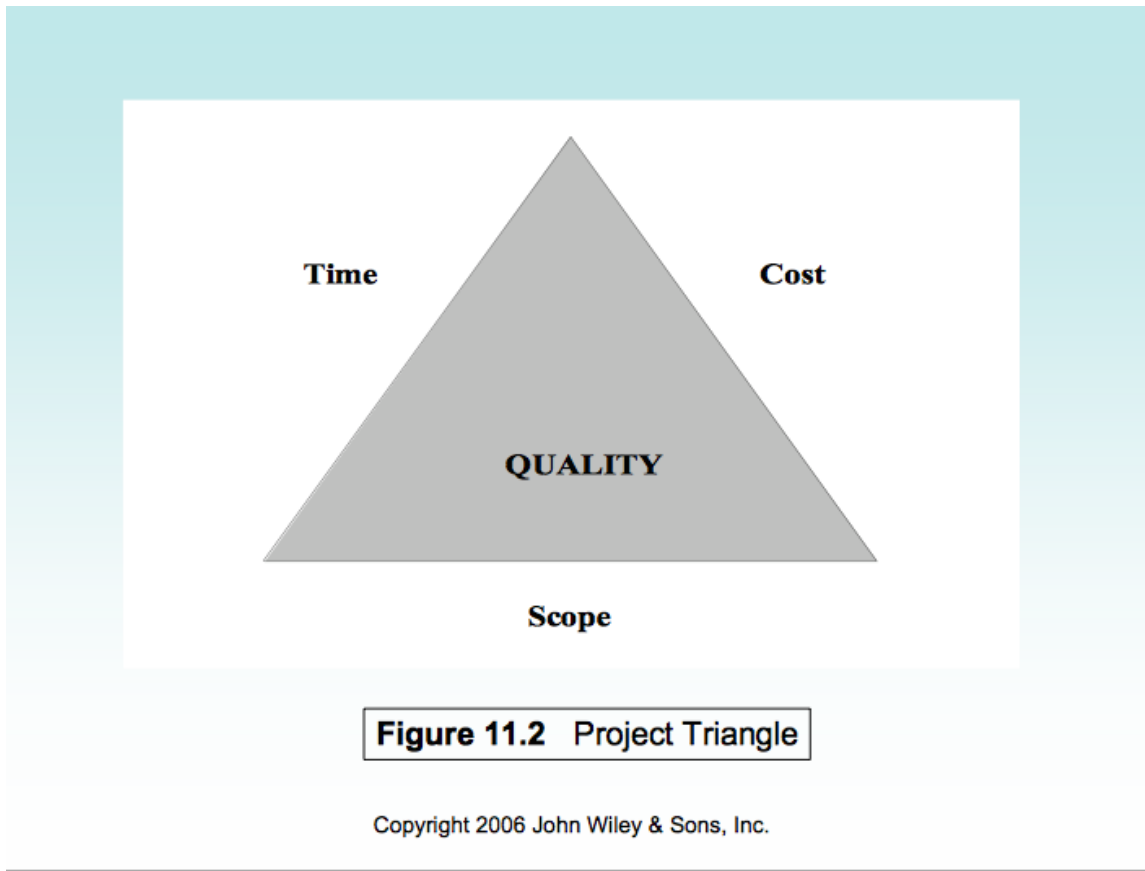


Figure 1: Project Triangle (Pearlson & Saunders, 2006)

Waterfall approach comes up as a unique management approach that is consulted by the professionals and the researchers. With the introduction of this structured approach, the percentage of the projects that are notified successful increased significantly. However, it was not successful as such to give the desired result for the projects, where the requirements are hardly defined and the end delivery is difficult to comprehend by the stakeholders. The most common example for those types of projects is software development projects, where the requirements are difficult to define in advance and exposed to a continuous change. For these types of projects, a new approach was developed. The agile/scrum technique was a unique approach for those projects.

The Problem: However, the needs of the companies are evolving and becoming more and more demanding each year. So that the complexity of the projects as well. Stakeholders in the projects do not always have the same sense of urgencies and their expectations from the same end product might differ. This challenges the project managers on meeting the demands of each stakeholder and forces to opt for a new, more broaden project management approach that includes more human types of issues as well as the hard skills. The current methodologies started to remain inadequate to meet the contemporary demands of the companies to manage the software projects with vague requirement definition and stakeholders with different sense of urgencies. In this situation, two problems arises the software development project managers encounter:

1. Developing and implementing high quality end delivery.

2. Effective management of the complex stakeholder environment in order to satisfy all stakeholders.

1.2 The Research Gap

There is abundance of literature that guides how to manage the software development projects that have unclear end product and vague requirements: Agile software development approach/scrum etc. There is also literature on how to manage the stakeholders that have different sense of urgencies: Stakeholder management methodologies. However, there is no structured methodology in the literature that combines these two fields: stakeholder management and agile approach.

As most of the published articles are about introducing the methodology and providing a guidance on adapting the agile software development approach, the variations (different focuses) on the practical implementation of the methodology is still very scarce. As an exception, Drury et. al., (Meghann, Conboy, & Power, 2012) exploited decision-making, which is an important aspect of software development projects. They identify the obstacles on decision-making by investigating six different case studies with agile software development. Then, they relate their results with the Descriptive Decision-making principles and elaborate on the effects of those obstacles.

Besides the decision-making, stakeholder involvement is another important aspect of the software development since stakeholders' contribution significantly affects the progress of the end product. Although the importance of the stakeholder involvement from the initial stages of the process is highlighted in the Scrum methodology, there is no clear focus on the effects of dynamic stakeholder interaction on the process and managing the different sense of urgencies.

1.3 The Research Goal

This study aims to propose a guideline to the professionals and the researchers with a more structured methodology on how to manage the software development projects with unclear requirement definition and stakeholders with different sense of urgencies.

How?: By combining the insights from the literature and a case study:

- Making a literature review about the techniques and methodologies used for implementing the hard and soft skills (Agile software development approach and the stakeholder management techniques)
- Analyzing the methodologies used in the case study.
- Proposing a more structured methodology in type of guideline utilizing and combining the insights from the literature and the case study.

The target group of this study: The professionals and researchers, which undertake software development projects that have unclear requirement definition, inadequate understanding of stakeholders about the end product of the project and complex stakeholder environment with different sense of urgencies.

1.4 Summary of the Research Problem and Goal

- Stakeholder relations are getting more complex because of the evolving needs of companies and the different sense of urgencies of stakeholders.

- In the literature, most of the software development projects with unclear requirement definition have one client with a unique sense of urgency. Therefore, current project management methodologies mostly focus on the development of the end-product and arranging regular meetings with the stakeholders to get their feedbacks and increasing their understanding on the end delivery. They do not have stakeholder engagement and planning how to involve them into the project and deal with the different sense of urgencies among the stakeholders.
- Agile approach is a contemporary and effective methodology that increases the success rates of the software development projects that have unclear requirement definition because of its inherent characteristic of requirement adjustability/adaptability.
- Stakeholder management methodologies are effective to manage the stakeholders with different sense of urgencies.
- There is no combined unique methodology, which manages the complex stakeholder environment, and also making sure that the project is on target despite the ambiguity of the definition of the requirements.
- There is a methodology needed that combines the agile software development and the stakeholder management methods to effectively manage the projects with unclear requirements and complex stakeholder environment.
- Therefore, the purpose of this study is proposing a guideline for professionals and academicians to manage the software development projects that have unclear end product definition and vague requirements with complex stakeholder environment.

1.5 The Research Question and Sub-questions:

Thesis Topic: Agile Approach and Stakeholder Management Techniques on a Software Development Project: Lessons Learned from Use Cases

1.5.1 Research Question:

“How can software projects having unclear requirement definitions and stakeholders with different sense of urgencies can be managed successfully?”

To have a detailed answer to the research question, the following three sub-research questions are answered.

1.5.2 Research Sub questions:

In order to give a detailed answer to the research question, following sub-questions should be answered:

R1: “What type of approach is needed to deal with different sense of urgencies in a software development project?

R2: What type of approach is needed to deal with unclear requirement definition in a software development project?

R3: What kind of methodologies could be suggested in order to successfully manage a software development project, having unclear requirement definition and stakeholders with different sense of urgencies and which stakeholders would be the action owner of these methodologies?

Methodology

This chapter discusses what kinds of methodologies are used in this study to answer the research question together with their reasoning. Following methodologies are used in general that will be explained further in this chapter:

- Action Research
- Literature Survey
- Case study

1.5.3 Action Research

Action research has been used in the social and medical sciences since the mid-twentieth century. Its popularity for use in scholarly investigations of information systems started to grow around the end of 1990s. Because of its characteristic grounding in practical action and aiming at solving immediate problem situation while carefully contributing to the theory, the methodology produces highly relevant research results (Baskerville, 1999).

In the action research, the researcher involves and works in the case study that he is going to use in his research himself and evaluate the project that he involves. The downside of this technique is that it is very likely to have biased evaluation if the author evaluates the process he actively involved and had an influence. With this technique, the researcher should be very careful on drawing conclusion. He should select the research area, where he can take out his influence (Verbraeck, 2015).

Beside its downsides, the action research is the best way of observing and drawing insights from a project because the researcher receives the input from the first person. Researcher is one of the stakeholders have impact on the outcome of the project, and part of the evolution of the project cycle (Verbraeck, 2015).

Action research is chosen for this study because it gives a unique opportunity to have clear observation on the case study. In order to avoid a biased evaluation, the researcher chose her research focus at the area that she had least influence. Her influence in the project was at the conceptual and the interface design of the project. However, the research focus is at the planning phase of the project, where she has no influence on. At the time when she involved in the project, the planning phase was already finalized. The project owners had already decided their approach and methodology to be used in the project. Therefore, the researcher did not influence this part, so in this way she could avoid the biased evaluation risk of the action research. On the other hand, she was the first witness of the outcome and effects of the project owner's choices by participating and being part of the implementation part of the project.

To conclude, by using the action research methodology, the researcher got the advantage of having the most effective way of observing the case study. On the other hand, selecting the research focus in the area that she had no or the least influence on, I could avoid the biased evaluation risk of the action research technique.

1.5.4 Literature Survey

The first step of developing a body of knowledge is searching and analyzing the previous works in the area of interest to understand how far the literature has gone through the studied issue (Kumar & Phrommathed, 2005). In order to develop a new approach or improving the existing methodologies, the recent and earlier works should be explored in a subject area. Only after a thorough understanding of the existing knowledge, it is possible to identify the areas, where the

further research is beneficial. This is possible with a careful and focused literature survey because it summarizes the state of the art in a specific subject (Rowley & Slack, 2004).

The literature in this research is used as an intake for the proposed management guideline. Literature survey will help to learn the current state of the art for the techniques and skills required for managing the software development projects with unclear requirement definition and stakeholders with different sense of urgencies. As the analyses from the literature show that although the literature provides proven methodologies for the software development projects with changing requirements, the answer of how to manage the stakeholders in case different sense of urgencies exist in such projects is unstudied. That means current literature remains insufficient to have a direct answer to the research question of this thesis. Therefore, the literature survey is not limited to the used techniques in software development projects. It also searches the stakeholder analysis and management techniques that are used in non-software development projects and/or in political environment, which could be implementable, but have not been practiced in the literature yet for the software development projects with complex stakeholder environment.

For this reason, the literature will be reviewed from two aspects in this research:

1. What are the methodologies in the literature used in software development projects that have unclear requirement definition?
2. What are the stakeholder analysis and management methods to deal with the different sense of urgencies?

The insights from the literature and the case study will be harmonized and a structured methodology will be developed to guide professionals and academics on how to approach and manage a software development project with continuously changing requirements and complex stakeholder environment. Utilizing the insights from a real-life example and combining them with the insights from the literature will bring a sensible solution to our research problem. Finally, consulting the professional experts will validate the proposed methodology. Expert view will be gathered by the interview technique.

Literature survey is done by using the academic sites such as Google Scholar, Scopus and Web of Science as well as the lecture books that are used in the Technology, Policy and Management Faculty in Delft University of Technology. For the first part of the literature survey, mainly project management books and articles are reviewed. For the second part of the literature survey, mainly the book of de Bruijn et. al., (de Bruijn & ten Heuvelhof, 2008) is consulted together with the other articles where their area of research is being used.

1.5.5 Case Study

Case study is an ideal approach when there is a need of holistic and in-depth investigation (Feagin, Orum, & Sjoberg, 1991). Case studies are used in many investigations, particularly in sociological studies, but more in instruction. When the procedures of case study methodology, which have been widely experienced and developed by Yin (Yin, 1984) (Yin, 1989) and Stake (Stake, 1995), are followed, the researcher will be counted to use a method as well developed and tested as any in the scientific field. Unlike experimental or quasi-experimental studies, case studies bring out the details from the viewpoint of participants by using multiple sources of data. They are not a sampling research; however, the case should be selected according to its capability of bringing the maximum learning for the research area in the period of time available for the study (Tellis, 1997).

This study is using case study methodology to bring practical insights from a real example of software development project, which has unclear requirement definition and complex stakeholder environment with different sense of urgencies. The case study used in this research exemplifies the research problem and brings a solution for a specific case.

The findings from the case study will be generalized by answering the question:

- What type of techniques could be used to improve the involvement of the stakeholders during the planning and the development phase of a software development project with unclear requirement definition and how could their level of trust to the project be increased and different sense of urgencies be managed?

For this specific research, it is analyzed a case study, where agile software development approach is applied. In this specific case, the end product is unclear to the stakeholders as the requirements are volatile and adjusted continuously during the development of the product. This characteristic of the project serves our first purpose: Unclear end product definition. Moreover, the client of the project is composed of several stakeholders with different sense of urgencies and dynamic power relations. This characteristic of the project serves our second purpose: Complex stakeholder environment. Regarding the mentioned characteristics of the project, it can be concluded that the selected case study has sufficient content for our research problem.

Another important reason selecting this case study is the methodology that is used for the management of the project. Regarding the complexity of the project, project manager used Agile Software Development Approach as the management solution of the project. In the literature, agile approach is considered an effective and a contemporary method to deal with the changing requirements in a software project, where the end product is rather unclear to the stakeholders. Project manager took the advantage of this characteristic of the agile method to cope with the continuously evolving requirements in the project. Additionally, they aimed to solve the problems coming along by the complex stakeholder environment by using soft skills in addition to the agile approach. This perspective was new to the literature as there is no written structured methodology on how agile approach manages the dynamic stakeholder environment with different sense of urgencies in a software development project. In that sense, this case study provides invaluable insights to the researcher.

In conclusion, as the case study provides sufficient content to our research problem and the proposed solution has an added value to the literature, we decided to analyze this case for our research topic.

1.5.6 Interview

Interview is one of the most popular qualitative data collection strategies, which is used across many disciplines (DiCicco-Bloom & Crabtree, 2006). In this research interview technique is used for two purposes:

1. **Interview to get insight for the case study:** It takes place with the project manager and the consultant to bring insights about the used methodologies and techniques in the case study to deal with the continuously changing requirements and stakeholders with different sense of urgencies.
2. **Interview to validate the proposed management guideline:** Two interviews have been conducted with software development experts to validate the proposed methodologies in the guideline. First interviewee is a software development expert, who managed

different software development projects for seven years. The second interviewee is the project manager of the case study issued in this research. He has extensive experience in large-scale global software development projects with unclear requirement definition and stakeholders with different sense of urgencies. Interviewees' feedbacks about the guideline are given in Validation of the Guideline.

1.6 The Structure of the Thesis

In the table below, the structure of the thesis is given:

Chapter 1	Introduction: Introducing the research problem and methodology
Chapter 2	Literature Survey: Software Development & Stakeholder Management
Chapter 3	Case Study: Part I - Stakeholder Analysis and Management
Chapter 4	Case Study: Part II - Software Design & Development
Chapter 5	Project Management Guideline
Chapter 6	Validation of the guideline: Expert opinion
Chapter 7	Conclusion
Chapter 8	Bibliography

- The purpose of the first chapter is introducing the problem, goal of the research, defining the research questions, discussing the methodologies used for the research and explaining the structure of the thesis.
- Second chapter is a literature survey consists of two parts:
 - Part I - Software Development Methodologies: In this part, used techniques and methodologies are given for software development projects, specifically agile approach, together with general literature about project management.
 - Part II - Stakeholder Analysis and Management: Second part of the literature survey is about the used techniques and methodologies for stakeholder analysis and management that could be applicable in a software development project.
- Third chapter is about used methodologies in the case study from a stakeholder analysis and management perspective. The insights gathered from this chapter become a substantial input for the guideline (chapter 5).
- The fourth chapter explains the software development process and the used methodologies in the case study.
- The fifth chapter is the core of the thesis, where the added value of the thesis takes place. It proposes a management guideline to the professionals and the researchers with a more structured methodology on managing the software development projects with unclear requirement definition and stakeholders with different sense of urgencies.
- The sixth chapter aims to validate the proposed methodologies in chapter five with expert opinion. These experts are manager of the project used in the case study, who has extensive experience in similar projects and an expert, who has been managed software development projects for seven years.
- The seventh chapter is the conclusion and the further study areas.

CHAPTER 2: LITERATURE SURVEY



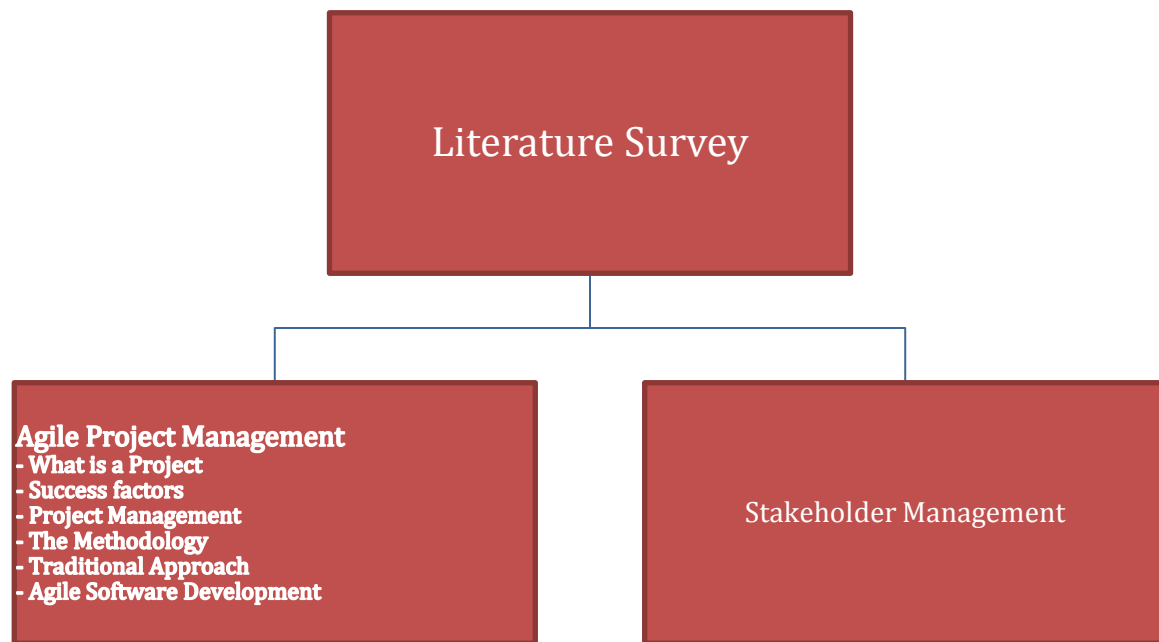
2 Literature Survey

The purpose of the literature survey in this study is having insights about the used methodologies for the management of software development projects, particularly having changing requirements and the methodologies used for stakeholder analysis and management.

The literature survey of this study will consist of two main parts: Agile Project Management, particularly agile software development approach and Stakeholder Management. The insights gathered from these two fields (together with the learning from the case study) will be used as an intake to develop a more structured methodology for the software development projects that have unclear requirement definition and complex stakeholder environment.

The first part of the literature survey will start with the definition of a project, the reason of the necessity of project management (PM), its role on implementing a project and the approach used for the PM. The literature survey will also mention about the necessity and the importance of a methodology for PM as the core of this study is developing a project management methodology. Although there are enormous amount of different methodology in the literature, the focus of the literature survey will be agile project management methodology, specifically agile software development approach (SCRUM) as it is the most common and effective PM methodology used for the projects that have unclear requirement definition. Additionally, the methodology used in the case study is also agile software development approach. Insights from the literature are necessary to make a comparison with its implementation in the case study.

The second part of the literature survey will focus on the stakeholder management. As it is explained in the introduction part, the classical agile project management approach remains inadequate to manage the stakeholders when there is divergent / different sense of urgencies. This part of the literature survey will search for the methodologies in the literature used for the management of a complex stakeholder environment. The insights gained from this part will be compared with the case study and it will be figured out to what extend these methodologies were used in the case study and which technique(s) from the literature that was/were not used in the case study would bring more effective results to the management of the project (if available).



2.1 LITERATURE SURVEY: Software Development Methodologies

First part of the literature survey will focus on the software development project management methodologies applied in the literature and their implementation. As the scope of this study is limited by the projects that have unclear/changing requirements and complex stakeholder environment with different sense of urgencies, the research will focus on the project management methodologies that are applied for those kinds of projects and their historical development, specifically agile methodology. However, the literature survey will start with a general overview about project, project management, importance of methodologies on project management etc. to give the reader more understanding why such kind of research topic is chosen for this study. And then, the search will narrow down to the methodologies used in software development projects.

2.1.1 What is a Project?

A project is a temporary undertaken endeavor that has a definite beginning and end, aims to perform a unique service or product (Guide, 2001). Beside this basic definition of it, the characteristics of a project can be detailed as follows (Nicholas & Steyn, 2012):

1. A project has a definable goal or purpose, well-defined deliverables and usually specified in terms of cost, schedule and performance requirements.
2. Every project must be *unique*, which means the product or the service performed has distinguishing characteristics that differentiates it from all its simulants.
3. Projects are *temporary* activities: they are ad hoc organizations of personnel, material and facilities aiming to accomplish a certain goal within a pre-defined time frame.
4. Projects bring skills and talents from different functions, professions and organizations together.
5. The uniqueness characteristic of projects causes uncertainty about the outcome because it aims to perform something new, which inheritably brings projects *unfamiliarity* and *risk*.
6. The projects are usually undertaken for the purpose of improving/healing something at stake. In case the projects are not carried out with special effort, they might end up with a failure that may jeopardize the organization or its goals.
7. A project is a *process* of working towards a certain goal. It passes through different phases in the project life cycle within the process that utilizes different tasks, people, organizations and resources in each phase.

Nicholas et. al., (Nicholas & Steyn, 2012) finds the second, third and the forth characteristics of a project more significant distinguishing. Every project is unique, unfamiliar, and involves multi-organizational and multi-functional interference that causes uncertainty and risk. It decreases the chance of achieving a successful outcome, which requires a special management type: *Project Management*.

2.1.2 Project Management: A systems approach to management

Pearlson et. al., (Pearlson & Saunders, Managing and Using Information Systems: A Strategic Approach, 2004) defines the project management as the application of knowledge, skills, tools and techniques to project activities for the purpose of meeting or exceeding the stakeholders' needs and expectations from a project. They state that PM is full of continual trade-offs that are supposed to be managed by the project managers.

Nicholas et. al., (Nicholas & Steyn, 2012) on the other hand, brings a systems approach to the project management: The role of management is planning, organizing and integrating the resources and the tasks in the way to achieve the organization's goals. The purpose of project management is managing a system of tasks, resources, people and organizations in order to achieve the goal of the project. This requires a systems approach to management (Nicholas & Steyn, 2012):

- A system is a collection of interrelated components or elements, working to serve the same final goal when being merged.
- The systems approach to management conceives the goal or solution proposed to a problem as the outcome of a system. The focus of the approach is optimizing the performance of the overall system rather than the individual components. It is done by defining the goal, identifying the elements or components of the system and then managing these elements to achieve the goal with the highest performance.
- Project management is a systems approach to management. A project is a system of interrelated components including the tasks, resources, schedules, budgets, stakeholders and so on. The function of project management is integrating all these components effectively to achieve the project goal.

2.1.3 Critical Success Factors of a project

In order to achieve the project goal in the most productive way, the projects should be completed successfully. Although the traditional definition of a project success is considered as the completion of an activity within the constraints of time, cost and performance, Harold R. Kerzner brings a contemporary approach to the success factors of a project (Kerzner, 2013):

Successful project is the completion of a project

- Within the allocated time period,
- Within the budget
- At the proper performance and the specification level
- With the approval/acceptance of the client/user
- With minimum scope changes (they should be agreed by the user/client)
- With no negative influence on the main work flow of the organization
- Without changing the corporate culture

2.1.4 Project Management Methodology

In previous sections the characteristics of a project and the importance of project management is explained. In order to manage such a complicated system (project) with many interrelated components effectively and successfully, project management needs a methodology. Methodology acts like a guideline for the team and the project manager to perform collaboratively (Nicholas & Steyn, 2012).

According to Archibald (Archibald, 2003), PM methodology enables bringing all important elements of information regarding (1) time, (2) cost in funds manpower or other key resources, (3) products or results of the project together. This information is gathered in every phases of the project. Then, all elements of information are exposed to an interrelated evaluation, which enables continual revision of future plans and projection of completion time and cost of the project. This provides integrated planning and the control of the project (Baker, Murphy, & Fisher, 2008).

Moreover, the project management methodology helps project manager to (1) identify the required project tasks, (2) identify the required resources and the costs, (3) establish priorities, (4) plan and update schedules, (5) monitor and control end-item quality and performance, and (6) measure project performance while proceeding from one phase of the project to another (Mathew).

As it is such important to have a methodology on managing a project, the following sections will focus on the several methodologies have been used for the software development projects, their main characteristics, advantages and the areas they remain inadequate on managing software development project will be discussed.

2.1.5 History of the Software Development Models

Software development activities such as requirement analysis, design, implementation and validation are carried out within a software development life cycle (Sommerville I. , 2010). The organizations tend to adapt the life cycles from the software development models that have been used and succeeded before. However, organizations adapting different development models to the life cycle of their software projects resulted in failure on achieving their business targets. The analysis performed by The Standish Group in 2009 (The Standish Group, 2009) shows that 24% of the software projects were failed with the reason of cancelation, 44% of them were reported as challenged with a reason of overdue, over budget and/or inadequate delivered functions/features, while solely 32% of the projects reported as successful. The high failure rates of the software projects became a motivation for the Traditional (plan driven) software development models (Tarhan & Yilmaz, 2014).

2.1.6 Waterfall model

The software development methodologies that have been used by the software engineering community since 1960s are subjected to the continuous improvement and refinement. These serious improvement efforts resulted in reaching mature and stable levels for the majority of the methods over years and referred as 'traditional software development methods'. Although there are several traditional software development models, which serve to different development issues and implementation conditions, the waterfall approach is the oldest and a widely used one. The waterfall approach is considered as a successful method especially for the large and complex engineering projects (Sommerville I. , Software Engineering, 2000).

The Waterfall approach life cycle is composed of five distinct and linear stages: Requirement Analysis and definition, System and Software design, Implementation and unit testing, Integration and system testing, Operation and maintenance (Huo, Verner, Zhu, & Babar, 2004). Likewise other traditional models, waterfall approach are processed with fully specified problems, rigorous planning, pre-defined processes and regular documentation (Sommerville I. , Software Engineering, 2010).

Although it is reported as successful with large and complex projects, waterfall method does not show the same success with the systems where the requirements are not pre-defined and continuously adjusted in each stage of the project. (Sommerville I. , Software Engineering, 2000). As a result of its cascaded structure, each step is executed when the previous step is finished.

Waterfall approach is an efficient approach as long as every requirement of the project is precisely identified based on a requirement analysis. This requires an effective planning of the requirements of the software before the implementation. Furthermore, the requirements of the

client may change after the introduction of the end product. The modifications on the initial steps may require other steps to be executed again, which increases the overall cost and time of the project substantially.

2.1.7 Iterative and Incremental Development (IID) Approach: Agile Model

As an alternative to the traditional approaches, agile methods were developed to address the number of the drawbacks issued in the waterfall model (Huo, Verner, Zhu, & Babar, 2004). The agile software development has drawn considerable attention of the research community since the creation of the agile manifesto in 2001. The number of scientific publications and participating (63) countries in the development of agility shows the significant increase in the popularity of agility among the research community after the announcement of agile manifesto. Abrahamsson et. al. (Abrahamsson, 2002) grounds the popularity of the Agile Software Development paradigm to its flexible characteristic on managing the volatile requirements and focus on the extensive collaboration between the clients and the developers. (Abrahamsson, 2002) In 2001, a group of agile software development experts described the agile manifesto as the expression of four comparative values of agility. The comparative values of the agility expressed in Agile manifesto are (Fowler & Highsmith, 2001):

- Individuals and interactions over processes and tools,
- Working software over contract negotiation,
- Customer collaboration over contract negotiation,
- Responding to change over following a plan.

The techniques that agile method takes advantage are mainly simple planning, short iteration, earlier release and customer feedback (Huo, Verner, Zhu, & Babar, 2004). The following paragraph explains the logic of the agile approach further.

To have a flexible project management approach that enable modifications on the requirements, Iterative and Incremental Development (IID) approach is created. In IID, the whole implementation is broken into several subsets of requirements. Fundamentally, IID consists of several Waterfall approach and at the end of each Waterfall, a feedback is taken from the client about the requirements. The expanded requirements are used in the next Waterfall cycle. The process continues iteratively. The main advantage of IID is that it is flexible through the steps so any changes on requirements do not have substantial effect at the end of the project.

Since software development may require rapid changes because of its nonlinear, unpredictable nature, it is important to use more flexible management rather than traditional approach (Dingsøyr, Dybå, & Moe, 2010). One of the IID's main benefits incremental approaches in software development is its rapid progress from the first prototype to the final product. This rapid development prevents the risks of problems that may occur at the final integration of different parts for the final product (Greer & Ruhe, 2004). Incremental software development is defined by Greer et. al., (Greer & Ruhe, 2004) in the following way:

Incremental software development is designed to deliver the system in smaller releases that are implemented sequentially rather than developing a monolithic system that is launched after the whole development process (Greer & Ruhe, 2004). Greer et. al., finds incremental approach more advantageous over the traditional waterfall approach due to its capability to changes. They point out the advantages of the incremental approach in the following way: First, Greer et. al., believe that the incremental approach enables gaining the benefits of the new system in earlier phases of the project if the requirements are prioritized: Implementing the most

important ones in the initial iterations and leaving the least important ones to the end. This approach guarantees acquisition of the substantial part of the end product even if the final requirements are omitted due to the insufficient time and budget. Secondly, as the client receives the part of the system at an early stage of the process, it is more likely for them to involve into the project actively, so support the development by giving feedbacks. Third, as the product is delivered in small releases, the team can make more accurate estimations about the time and the budget of the project. Forth, incremental approach eases the adjustment of the plans according to the feedbacks of the client. The last but not the least important advantages of the incremental approach is its capability of reaction to the adjustments or additions to the requirements, which is much better than the traditional waterfall approach.

Agile methodology utilizes IID in a way to simplify the project process by reducing the planning complexity, emphasizing the customer value and encouraging the stakeholder involvement and collaboration. The involvement of the stakeholder in the whole process of the development especially from the first prototype decreases the risk of unsatisfactory products at the end of the project. Since a feedback can be obtained from the stakeholders initially after the first prototype, it is still early enough to make necessary improvements until the final product. Furthermore, contact with the development team and the stakeholders by means of prototypes prevent the team to focus too much on the technological innovation rather than the real requirements (Highsmith, 2002).

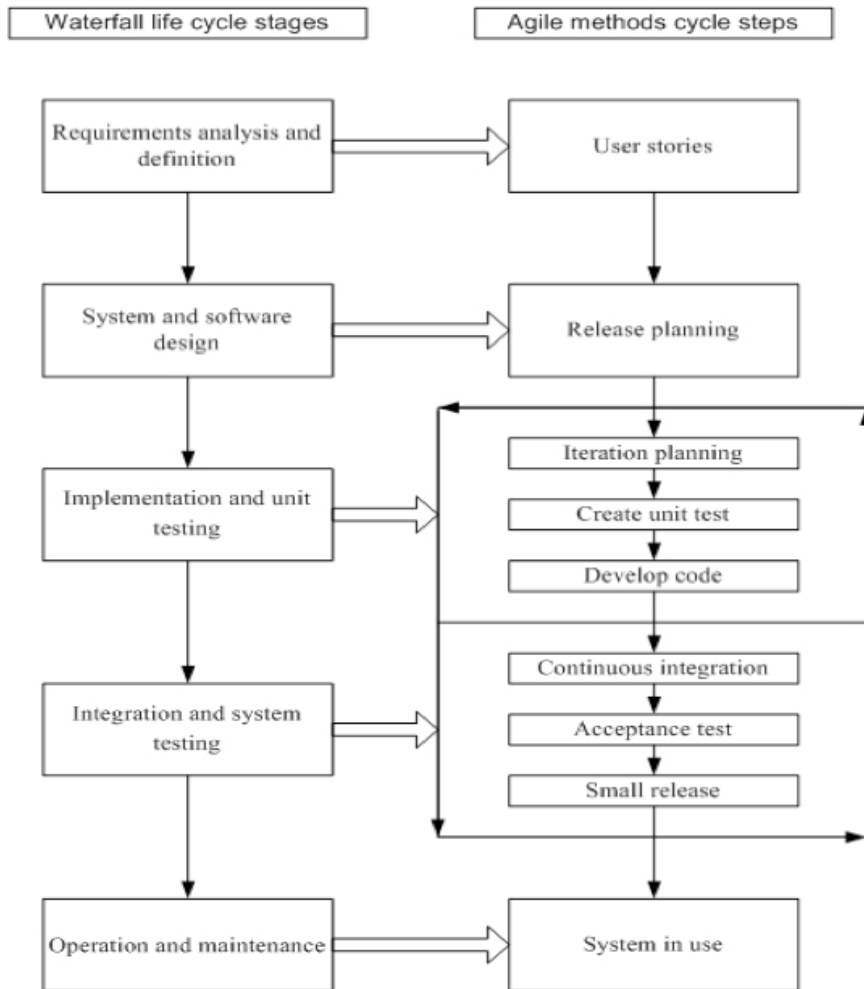


Figure 2: Waterfall model vs. Agile method life cycle (Huo, Verner, Zhu, & Babar, 2004)

2.1.8 Software Development with Scrum

Recently, a popular approach that is used for the software engineering is called Scrum (Schwaber, 1997). Scrum was introduced by Takeuchi et. al. (Takeuchi & Nonaka, 1986). The scrum methodology aims to have a drastic simplification on the project management approach and adopts the “inspect and adapt” ideology. In scrum, the software is delivered in **Sprints**, which is an incremental approach that consists of 2 to 4 weeks of iteration. The software continuously evolves based on the experiences gained during the process and the feedback from the client (Schwaber & Beedle, 2002).

The Scrum team starts the sprint with a detailed plan that lasts up to four hours. It ends with the sprint review meeting, where the stakeholders take part; review the state of the business, market and technology. Another type of meeting in scrum is retrospective meeting. In retrospective meetings, the effectiveness of the team is evaluated; plans are made to enhance the quality of the teamwork. Lastly, daily scrum meetings takes place 15 minutes each day by the scrum team. The following questions are addressed in daily scrum meetings (Schwaber & Beedle, 2002):

- What did I do yesterday?
- What will I do today?

- What difficulties (obstacles) do I have?

Scrum has three main deliverables: Product backlogs, sprint backlogs and burn-down charts. The backlog is for the client's requirements and the burn-down charts show the cumulative remaining work (Schwaber & Beedle, 2002).

In terms of administration, Scrum methodology consists of three roles namely the team, the product owner and the scrum master. The team is the developer of the product for the product owner. Through the sprints, there is a close collaboration between these two groups until the final product. At the end of each iteration, the additive requirements are given as a feedback to the team for the next iteration. One of the unique properties of the scrum methodology among the others is the existence of the scrum master. The scrum master is the main link between the development team and the product owner. The scrum master also controls the progress of the development team in the progress (Schwaber & Beedle, 2002).

Software development is a nonlinear process, which requires continuous feedback and control over the progress. Scrum has been specifically designed for the software development process because of its flexibility and adaptability. Scrum enables the stakeholders interfere to the development process as well as it gives the development team the freedom of choosing any desired technique and method to be utilized. Detecting any deficiencies early enough to make necessary modifications is possible with the frequent meetings through the process and intermediate prototypes help the development team to stay focused on the main objectives. These properties make Scrum to be one of the most effective management approaches to be utilized in software development projects (Schwaber, 1997) (Schwaber & Beedle, 2002).

2.1.9 Current States of the Art of the Agile Software Development

According to the research of Dingsøyr (Dingsøyr, Dybå, & Moe, 2010), the articles published between 2003 and 2011 related to the agile software development are mostly about improving the understanding of the agile concepts. Among these published articles the most common agility methods were XP (Extreme Programming) and Scrum. Other prominent topics about agility are adoption and/or adaptation of agility, the evaluation of the adaptation for the organizations that are not inherently prone to agility and comparison between the agile and traditional software development methods.

The recent studies concentrate on improving the coordination (Strode, Huff, Hope, & Link, 2012) and decision-making (Drurya, Conboy, & Power, 2012) in agile software development projects. The applicability of agile approach on post-adaptation stages has been investigated as a case study by Senapathi et. al. (Senapathi & Srinivasan, 2012). Furthermore, Adolph et. al. (Adolph, Hall, & Kruchten, 2011) has showed that improvements on agile approach is achievable based on considering social factors in software development.

Moreover, Begel et. al., made an empirical study by conducting an anonymous online web-based survey with 2,821 people composed of software developers, test developers and managers to investigate Agile development and its perception by people in development, testing and management (Begel & Nagappan, 2007). Their finding showed that scrum is the most popular Agile Software Development methodology among the participants. Their study indicated the following benefits of Agile Development Methodologies out of the results of the survey. The highest number of respondent approves the first one on the list, while fewer respondents towards the bottom of the list cite them.

Table 1: Benefits of Agile Software Development Methodology (Begel & Nagappan, 2007)

1. Improved Communication and Coordination
2. Quick Releases
3. Flexibility of Design Quicker Response to Changes
4. More Reasonable Process
5. Increased Quality
6. Better Customer Focus
7. Improved Focus - Better Prioritization
8. Increased Productivity
9. Better Morale
10. Testing First

The participants of the survey gave the highest benefit of the Agile approach to improved communication and coordination among the team members. They indicated that daily scrums were useful to bring the testers and developers together, which also increased the awareness on the activities of the team members. They found sprints useful for early discovery of the development issues/problems, which result in early solution creation. The second most cited benefit of agile practice is quick releases, which causes an ongoing tracking of the progress and monitoring the software quality. In the third place, the flexibility of the design comes. Developers noted that short sprints with a focus on the customer feedback enabled better agility and efficiency to respond the continuously changing requirements. Another benefit is more reasonable process. Participants noted that developers waste less time on the tasks they perceive irrelevant such as “large specs that are out of date before they are finished”. Improved quality from design and architecture to performance of the software is another important benefit noted by the respondents. They find out that improved communication thanks to agile approach is important for faster turnaround time on blocking the bugs. The rest of the benefits of the Agile methodology according to Begel et. al., (Begel & Nagappan, 2007) is better focus on customers, better prioritization of development and focus on the product, improved productivity, increased morale-motivation and more reliance on test-driven development. All these benefits are further explained in the study of Begel et. al., (Begel & Nagappan, 2007).

However, while these benefits are indicated by the respondents in Begel et. al.,’s study most of the participants were questioning whether it is possible to scale agile methodology in larger projects, especially with multi-cultural environment (Begel & Nagappan, 2007). Begel et. al.,’s study do not cover this variation.

2.1.10 Application of Scrum in Large-Scale Global Software Development Projects

Paasivaara et. al., (Paasivaara, Durasiewicz, & Lassenius, 2008) noted in their research that combining large-scale global software development and agile practices is a perceived as a challenge by many companies. Although successful practices of agile methodology already exists in small distributed projects, Paasivaara et. al., find out that how these practices could be applied in large scale projects is unstudied in the literature. Therefore, they conducted a single-case study of a large software development project with a 40-person development organization distributed between Norway to Malaysia. They apply agile practices from scrum. Based on several interviews on the development organization, they found out how Scrum practices are successfully applied in a multi-national environment, such as using teleconference and web cameras for daily scrum meetings, synchronized 4-week sprints and weekly scrums. Additionally,

they identified some agile practices, for instance frequent site visits, unofficial distributed meetings and annual gatherings etc.

Paasivaara et. al., (Paasivaara, Durasiewicz, & Lassenius, 2008) suggests the following agile practices for the large-scale software development projects:

1. **Daily Scrum meetings:** These meetings can be arranged by using telephone conferencing and web-cameras. It is possible to share applications as well though it is not needed.
2. **Weekly scrum-of-scrums:** They are arranged once a week, which takes half an hour. One team member from each team and all scrum masters attend this meeting. Each team discusses what have been done last week and what is the plan for the next week.
3. **Synchronized 4-week sprints:** If there are several teams in a project they come together each month (4-week sprints), at the end of each sprint.
4. **Distributed sprint planning meetings:** they are divided into several meeting. Their number depends on how many local market is included in the process. In these meeting, how scrum is applied in each team is discussed. Paasivaara et. al., (Paasivaara, Durasiewicz, & Lassenius, 2008) used Microsoft NetMeeting used for their application sharing. They arranged virtual meetings with teleconferencing.
5. **Sprint demos:** Meetings are for sharing the developed demos. For this purpose, teleconference and application sharing tools are used.
6. **Retrospective Meetings:** it is a one-hour meeting directly after the demos. The team, product owner and scrum master participate the meeting to discuss the done improvements. Questions to ask in retrospective: "What has been good during this sprint?", "What has not been that good?" and "What kind of improvements could we do?".

Agility supporting distributed practices that Paasivaara et. al., (**Paasivaara, Durasiewicz, & Lassenius, 2008**) suggest are unofficial distributed meetings through chat, email, teleconferences over the internet using headsets; centralized version control, which enables team members abroad to connect the server via VPN to check the code; having site visits during first iteration to discuss how to apply scrum; onsite system expert; frequent visits to the site especially at the critical moments to collocate the team and annual gatherings, which is a social event brings all the teams together to discuss the future of the product and for team building.

Paasivaara et. al., concludes that all these agile and non-agile activities in a large-scale software development project that uses scrum provided better quality of work, better and more quality of communication within the stakeholders and improved motivation within the team members.

Moreover, Sutherland et. al., (Sutherland, Viktorov, Blount, & Puntikov, 2007) analyze and recommend the best practices for globally distributed Agile teams. Likewise Paasivaara et. al., (Paasivaara, Durasiewicz, & Lassenius, 2008), Sutherland et. al., (Sutherland, Viktorov, Blount, & Puntikov, 2007) also suggested Scrum-of-Scrums as the best practice for the large scale agile projects, where scrum masters meet regularly across locations by linking the scrum teams in the project that are working globally. Sutherland et. al., (Sutherland, Viktorov, Blount, & Puntikov, 2007) supports scrum-of-scrum practice because of its characteristic of effective

communication, cooperation, cross-fertilization and appropriateness to inexperienced joiners to Agile development. In their study they show that daily scrum meetings prevents disparities in work styles and help overcoming cultural barriers.

2.1.11 Conclusion

In the first part of the literature survey, the used techniques and methods for the software projects in the literature are evaluated. Application of the agile approach in large-scale global software development projects is reviewed. Regarding the current state of the art although agile / iterative and incremental approach is successful on the management of continuously evolving requirements, which is the basic characteristic of the software projects, there is no sufficient focus on the literature for the stakeholder management of software development projects in multi-actor environment.

In order to fill this gap, the following part of the literature survey will investigate the stakeholder management techniques that can be implementable in such projects.

2.2 Literature II: Stakeholder Analysis and Management

In the first part of the literature survey, benefits of the agile/scrum applications on a software development project with changing requirements are indicated. It shows implications from small, dedicated projects to large-scale global projects. However, one significant point of the scope of this study is still missing in the literature: Involving stakeholders with different sense of urgencies to the software development process. As there is no practice in the literature on how to deal with different sense of urgencies in a software development project, this chapter will survey stakeholder management applications in other fields, where different sense of urgencies exist. In order to manage the different sense of urgencies, the project owner needs to prioritize the stakeholders and determine the strategies to apply on these stakeholders, which needs stakeholder analysis. As the literature does not have practices on stakeholder analysis techniques specifically adapted in software development projects, this chapter will also survey the stakeholder analysis techniques that are applied in non-software development projects. Guideline chapter will present the adaption and application of these practices on software development projects.

The survey will mainly bases insights from the book of de Bruijn et. al., (de Bruijn & ten Heuvelhof, 2008). Hans de Bruijn and Ernst ten Heuvelhof provides invaluable insights in their book with depth of understanding and wealth of advice on a multi-stakeholder decision-making process.

The key point of this chapter is that the lessons learned from Bruijn et. al., is elaborated in a way, which can be implemented in a project management environment. Bruijn et. al.,’s insights are designed for the political environment, where there is a lot of negotiation takes place but no end-product is developed. However, unlike politicians, project managers have responsibility to deliver a successful end product with several restrictions such as cost, time and resource. This fact urges project managers to keep their focus on the end delivery rather than ongoing and endless negotiations. Still, negotiation is an inevitable and crucial part of a project with multi-stakeholder environment especially when different sense of urgencies takes place. Therefore, it is their responsibility to find the balance on putting the right amount of effort between the product development and negotiation. The dilemmas related to this issue are further discussed in Section: 2.2.4.

Bruijn et. al. suggests many techniques for the stakeholder management but only the ones that are relevant and applicable in a project management environment are discussed in this study. This chapter takes the insights that can be implemented on a project based environment.

Apart from the insights used in the political environment, this chapter also includes literature that is already being used in the project management environment such as stakeholder mapping.

Insights taken from literature, whether they have been implemented only in a political environment or project environment will be harmonized with the insights gathered from the case study and then will be elaborated in the guideline chapter in the way that can be implemented in a software project with multi-actor environment..

The structure of this chapter is as following:

3. Introduction to multi-actor decision-making: General characteristic of a multi-actor environment
4. Actors and Actor Analyses

5. Multi-issue decision-making
6. Possible dilemmas can be faced in a software project with multi-actor environment
7. Conclusion

2.2.1 Introduction

When a decision-making takes place in a network, there is always an involvement of several actors. They usually have different interests and are dependent on each other. That means, there is interdependency between the actors. When this is the case, they cannot solve their problems independently as they are tied to each other, which require joint decision-making in order to realize their own goals. However, this is not a straightforward process because the actors have many mutual differences (variety), which hampers the cooperation between actors and joint decision-making. In some situations, actors may not be interested in cooperation at all (closedness). This may result in actors joining or withdrawing from the network, which is called (dynamic) stakeholder environment. In this case, the number of actors in decision-making process continuously changes, which makes the process capricious and unstructured (Bosch & Postma, 1995).

Characteristics of decision-making in networks (de Bruijn & ten Heuvelhof, 2008):

- Irregular and no clear sequence of activities
- Rounds
- Actors join and withdraw and behave strategically
- Several arenas, no isolated starting and end point
- Content of the problem shifts
- Incentive to regard problems as unstructured
- Flexibility and unpredictability

Capriciousness is inevitable in decision-making processes in networks and this provides each actor to offer his/her own interest. However, this characteristics of multi-actor decision-making requires completely different strategies than those recommended in many decision-making models.

Following sections discuss multi-issue decision-making strategy that an actor, facing with capriciousness in network-like decision-making, can use to realize his goals. The decisions are made in a process of interaction between the interdependent players in a network. The actor should inquiry how to influence the process of interaction. He should know which other actors may contribute on serving his interests and how could the actors be committed to those interests. Therefore, the actor should give the primary focus on the actors, whose support is needed, rather than the content of the problem (de Bruijn & ten Heuvelhof, 2008).

In order to manipulate other actors to serve on a particular interest, it is useful to know the types of actors and their power positions in a network. Therefore, this section will start with the analyses of actors.

2.2.2 Actors and Actor Analyses

Decision-making in networks requires high attention on the actors, whose interests might be harmed by the decision-making. The actors, who want to take initiative for a decision and

dependent on other actors for this decision, should at least know the following characteristics of other actors (Madeleine, 2004):

- The stances of actors: What view do they take?
- The underlying **interests** of actors: Why do they take for that particular view?
- The **resources** of actors: What means do they have to block or promote a decision?
- The **relations** of these actors: With what other actors do they maintain relations?
- The **repetitive** character of the relation with the actor: How often and in connection with what subjects will the initiator meet the other actors again?

2.2.2.1 Opinions and interests:

Actors have opinions about an initiative that relies on their interests. If the actor that want to take the initiative on decision-making knows not only the opinion of other parties but also their underlying interests, he can create room for the decision-making. If the discussion is conducted on the level of opinions, it is hardly any room for negotiation. On the other hand, if the initiator knows the interests of other actors behind their opinions, it becomes more open to negotiation, so room for decision-making can be created.

2.2.2.2 Resources, relations and the repetitive character of relations:

It is important to know what resources other actors have because initiator can use those resources during the process to realize his own interests. Some examples to those resources are funds, authority, knowledge or reputation. In fact, there is one special resource, whose impact causes it to stand out among others: an actor's network of relations. When an initiator is confronted with an actor, he should always know how extensive that actor's network relations is. The greater his network of relations, the more important to gain his support is. In case there is a conflict between the initiator and such powerful actor may cause that actor to activate his relations and set others against the initiator. Therefore, the initiator should put special attention to those actors. All these features of those actors leads to three types of power positions:

- Production power: Actor might have a positive contribution towards the realization of initiator's interest.
- Blocking power: The actor has blocking power can only halt the progress.
- A diffuse power position: His power and effectiveness is unclear and dynamic for the initiator.

After the initiator decides who are the main players, what are their power positions in decision-making, he can design his strategy accordingly. Muray et. al., introduces the stakeholder mapping technique in their studies in order to understand the interest and power relations of each stakeholder (Murray-Webster & Simon, 2006). Murray et. al., state that considering and understanding the stakeholders is the first step of any managed change initiative.

There are three main dimensions important to know initially for the stakeholder analysis (Murray-Webster & Simon, 2006):

- The **power** or the ability to influence of the organization: The power might be derived either from the stakeholder's positional or resource power in the organization or from their credibility as a leader or expert.

- The **interest** of the organization: it is measured by understanding whether they will be active or passive in the project.
- The **attitude** of the organization: it is the representation of whether the stakeholder will support (back) or resist (block) to the project.

These dimensions are useful to gain insight about the stakeholders. The commonly used two-dimensional grid technique is composed of two axes that are labeled with the stakeholder status/behavior. The area between the axes represents the name of the each group. Commonly used grids are “power vs interest” and “interest vs Attribute”. However, Murray et. al., experienced that this technique is insufficient to have a thorough understanding about the stakeholders. They support their claim in this way:

Power vs Interest: It helps the analyzer to understand the power of the stakeholder but also whether the stakeholder will be active (interested) or passive in the project. However, it is useless without knowing whether the stakeholder is “against” or “for” to the project.

Interest vs Attribute: It is important to understand whether the stakeholder is active backer or active blocker. However, before this analysis it is important to know the power of the stakeholder, so it will be clear whether the stakeholder is influential or not.

In order to overcome this ambiguity, Murray et. al., proposed a three dimensional grid (shown in Figure X) that maps all the criteria needed to be considered. It gives a more descriptive and useful guide that can be checked during the overall process of the stakeholder analysis. This technique stimulates the thoughts better and gives a truly meaningful information to the project manager to elaborate the stakeholders.

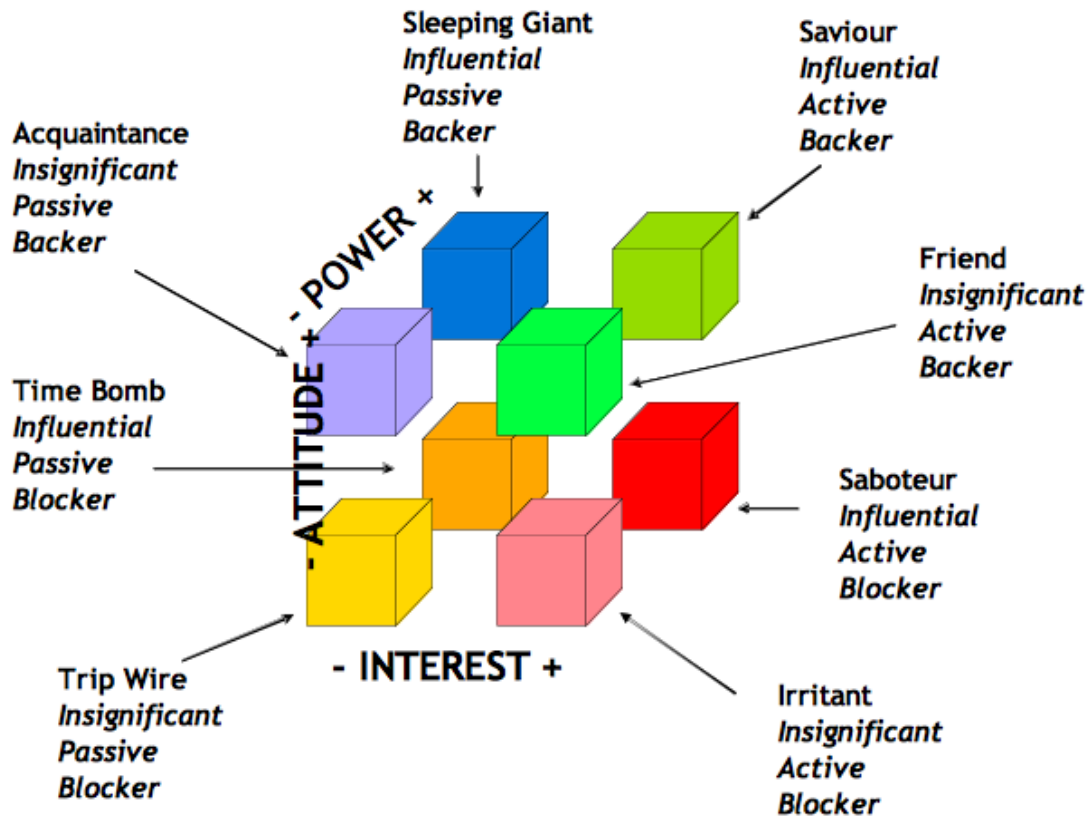


Figure 3: Three-dimensional grid for the stakeholder analysis

Using this approach each of the eight labels can be summarized as shown (Murray-Webster & Simon, 2006):

- **Saviour** – powerful, high interest, positive attitude or alternatively influential, active, backer. They need to be paid attention to; you should do whatever necessary to keep them on your side – pander to their needs.
- **Friend** – low power, high interest, positive attitude or alternatively insignificant, active, backer. They should be used as a confidant or sounding board.
- **Saboteur** - powerful, high interest, negative attitude or alternatively influential, active, blocker. They need to be engaged in order to disengage. You should be prepared to 'clean-up after them'.
- **Irritant** – low power, high interest, negative attitude or alternatively insignificant, active, blocker. They need to be engaged so that they stop 'eating away' and then be 'put back in their box'.
- **Sleeping Giant** - powerful, low interest, positive attitude or alternatively influential, passive, backer. They need to be engaged in order to awaken them.
- **Acquaintance** –low power, low interest, positive attitude or alternatively insignificant, passive, backer. They need to be kept informed and communicated with on a 'transmit only' basis.
- **Time Bomb** - powerful, low interest, negative attitude or alternatively influential, passive, blocker. They need to be understood so they can be 'defused before the bomb goes off'.

- **Trip Wire** – low power, low interest, negative attitude or alternatively insignificant, passive, blocker. They need to be understood so you can ‘watch your step’ and avoid ‘tripping up’.

Any kind of stakeholder grid, including the three-dimensional one can be useful when it is used sensibly. Establishing the positions/behaviors of the stakeholders can only give awareness about their status. The value of this analysis would be scarce if it is not followed up. There might be stakeholders, which are important to the project. Those actors could be engaged by negotiation and dialogue after checking out their real drivers and concerns. This kind of engagement turns the stakeholder mapping into a stakeholder analysis. Stakeholder mapping is a simple technique but it is a starting point for one of the most crucial activities when managing change (Murray-Webster & Simon, 2006).

2.2.2.3 Related Works:

Newcombe (Newcombe, 2010) argues in his book that stakeholder-mapping technique is an effective methodology for project managers on conducting the analysis of power, predictability and interest of key project stakeholders. He supports his arguments by exemplifying a large construction project. Furthermore, Bryson et. al., (Bryson, 2007) give insight in his book on how and why managers should use stakeholder identification and analysis techniques to help their organizations to fulfill their targets and creating a public figure. His research argues that wise use of stakeholder analyses can help framing the solvable issues that are technically feasible and politically acceptable. Moreover, Crosby et. al., (Crosby, 1991) explains the purpose of stakeholder analysis as having an indication of whose interests should be taken into account when making a decision. He argues that stakeholder analysis is a vital tool for strategic managers on the recognition of the key roles of the stakeholders on determining the policies, their implementations and outcomes.

2.2.2.4 Conclusion:

In this section, the techniques to identify the stakeholders, their importance, power relations etc. is discussed. Following section suggests multi-issue decision-making strategy to involve stakeholders into the decision-making procedure.

2.2.3 Multi-issue decision-making

Once the initiator decides which actors to involve into the decision-making process or which actors to focus on, they should find a strategy to involve them into the decision-making. Bruijn et. al., (de Bruijn & ten Heuvelhof, 2008) suggest multi-issue decision-making strategy to involve the stakeholders into the loop and manipulate them in the way to realize initiator’s own goals. The reason is that if the decision-making in network is limited to one subject (one-issue decision-making), then there is a risk of deadlock. When different actors take different opinions on the issue, it is little room for compromise. Therefore, the initiator should turn the decision-making into a multi-issue game by:

1. Creating several issues on the agenda
2. Making sure that there are sufficient interesting issues for each actor on the agenda
3. In this way making the participation to the decision-making process attractive to those actors.

2.2.3.1 The advantages of a multi-issue decision-making:

One important advantage of multi issue decision-making is that initiator can create a room for a process of giving and taking. Other actors can be persuaded to the decision of the initiator by offering other issues that are attractive to them. Giving and taking process is necessary for the smooth progress of decision-making in networks.

Second advantage of a multi-issue decision-making is providing incentives for cooperative behavior thanks to its characteristic of changing coalition. In one-issue decision game, actors can easily position against each other when their interests are not in common on the issue at hand. On the other hand, in multi-issue game, coalitions are changing continuously. Actors that have conflict in one issue may have coalition on another issue. In Figure 4: Multi-issue game and changing coalitionsFigure 4 below, actors P1 and P2 are positioned against P3, P4 and P5 on a one-issue coalition. On the other hand on multi-issue coalition, actors having conflict in one issue, are having coalition in another issue. Therefore, they do not position against each other.

According to Bruijn et.al., (de Bruijn & ten Heuvelhof, 2008) one risk of multi-issue gaming is the development of a counter-strategy by other actors once they realize the initiator's strategy. They may end up with long wish lists if they realize the initiator always solve the problems with the multi-issue game. The trick here is that the initiator should put whatever they wish to the agenda because trying to reduce the wishes before starting the game may cause resistance of other parties and so, the process may not start at all. The initiator's first interest should be involving the actors to the process. If the initiator places all the wishes to the agenda, an incentive is created for the parties to join the decision-making process. It becomes difficult not to enter the process. Once they involve into the process, it is difficult to leave. If they decide to leave, they will not only jeopardize their own prospect of gain but also other actors' as well. In the future, they will need other parties for different issues, so they will not want to harm their relations with others. When the process starts, they will start exchanging information and possibly become sensitive on what is possible and feasible and what is not. They will also become aware that trade-offs have to be made. They do not only become sensitive to the fact that those trade-offs have to be made but also they actually make them happen. They will be willing to abandon some of their wishes because they will learn during the process that they have to trade-off their wishes in order to make a successful decision-making. In the end, the actors in the process reach a package deal, which is attractive to all members. There will be number of decisions that are mutually connected. The actors will know that the only way to secure their own gain is to support others on securing their own gains. Therefore, multi-issue game is a powerful tool on a multi-actor decision-making.

2.2.3.2 Related Works:

The literature is quite limited on how to apply the multi-decision-making technique but the importance of multi-issue decision-making is being discussed and several studies exist on how to improve the efficiency of the multi-issue decision-making can be improved. One of these studies on multi-actor decision-making is done by Coehorn et. al., (Robert M. Coehoorn & Jennings, 2004). They highlight the importance of learning opponent's preferences to make effective multi-issue decision-making. For this, they suggest kernel density estimation to make negotiations more efficient regarding the utility and the time of agreement. Moreover, Raymond Y.K. Lau (Lau, 2005) emphasizes the characteristic of real-world negotiations such as complex negotiation spaces, tough negotiation deadline, limited information about the opponents and volatile negotiator preferences. They propose GA-based adaptive negotiation

mechanism because it supports multi-party multi-issue negotiations based on a distributive decision-making model. According to the study of Lau, his mechanism guarantees theoretically optimal negotiation model. Lastly, Luo et. al., (Luo , Jennings, Shadbolt, Leung, & Lee, 2003) develops a fuzzy constraint based model for bilateral multi-issue negotiation in trading environments. They seek for a fair deal for both parties while maximizing their own pay-offs.

2.2.3.3 Conclusion:

Until now, the importance of multi-issue decision-making strategy in a multi-actor environment and its implementation is discussed. However, all these given insights are created considering the political environment, where there is a lot of negotiation takes place but no end product (delivery) is produced. As it is already mentioned before, project based environment has different incentives than political environment such as the quality of the end product, time and cost. Therefore, the insights taken from Bruijn et. al., may not be directly implementable in a project based environment. The guideline chapter will explain how the given insights by Bruijn et. al., can be elaborated in a software development project with stakeholders having different sense of urgencies and in which phase of the project can be used.

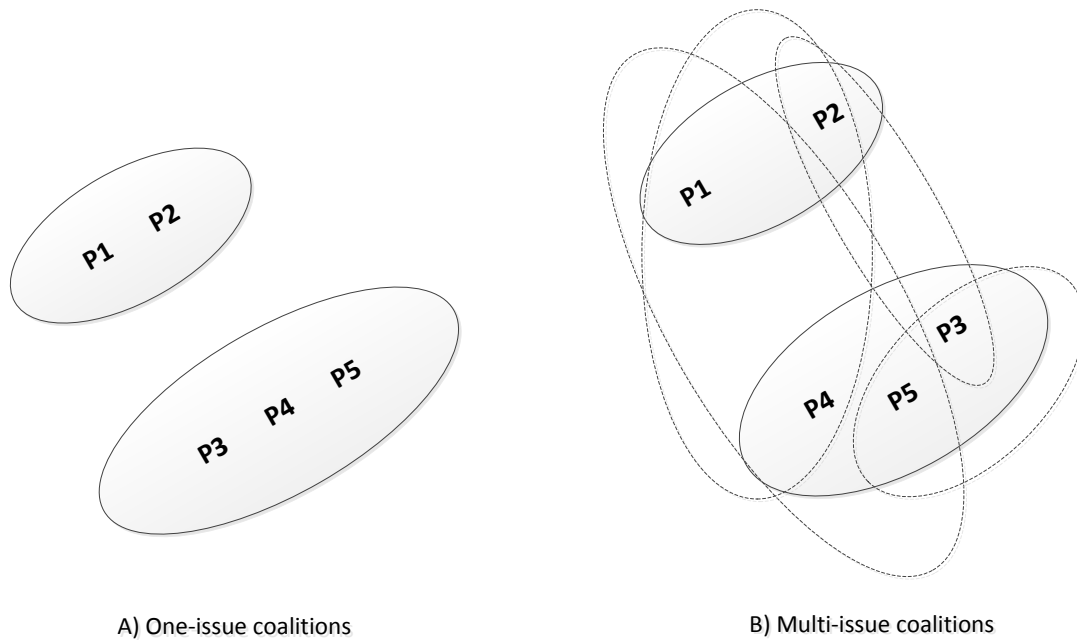


Figure 4: Multi-issue game and changing coalitions

2.2.4 Dilemmas faced in a multi-actor decision making environment

De Bruijn et. al., (De Bruijn, Ten Heuvelhof, & In't Veld, 2010) discussed the dilemmas faced when designing a process in a multi-actor environment. They explain process management as not taking unilateral decisions but adopting an open attitude. The process is made appealing to other parties by offering the opportunity to affect the decision-making and highlighting the issues they are interested in. Other parties feel that their interest might be put into the negotiation agenda so they have a prospect of gain. De Bruijn et.al., (De Bruijn, Ten Heuvelhof, & In't Veld, 2010) explain this strategy as “Openness”. However, even the open decision-making is opted for and there is discussion and negotiation, there is no guarantee to come up with a decision. There is a possibility to end up with many sluggish processes with no clear end result.

On the other hand, if there are not enough agenda points in the negotiation (closeness) there is a risk of not attracting the key stakeholders. One of the design principles that De Bruijn et. al., mention is that the initiator should make sure that there is sufficient momentum and progress in the process. Therefore, the scope of the negotiation should be balanced well that there is enough topic in the agenda that is appeal to all key stakeholders; on the other hand scope is not too broad that may halt the progress.

Another dilemma in a decision-making is negotiated nonsense vs design nonsense. De Bruijn et. al., (De Bruijn, Ten Heuvelhof, & In't Veld, 2010) propose stimulating 'early participation' to prevent slow start, which is the characteristic of processes that require wide participation. They explain the reason of this slow participation with possible disagreements with the starting conditions of the process or unsuitability of the chosen moments to start date of the process to some actors. As slow start may cause difficulty to organize the negotiations the initiator should stimulate the early participation by making the waiting time less appealing option to the participants and introducing the arrangements. If initiator can manage early participation, he can have more organized negotiations, so the result would not be negotiated nonsense. Also, as there will be enough input from the stakeholders from the initial stages of the process, there will be enough effort to focus on delivering high quality product, which prevents having design nonsense result. Another strategy to avoid this dilemma proposed by De Bruijn et. al., (De Bruijn, Ten Heuvelhof, & In't Veld, 2010) is creating 'Quick Wins'. Processes always involve conflicting demands. When this is the case, initiator should make sure to give sufficient gain to all parties with the right balance at the beginning of the process to keep all stakeholders in the loop. In this way, the initiator can keep the involvement of the stakeholders to the process active. When there is active stakeholder involvement, the product is designed according to the needs of the stakeholders, which prevents negotiated nonsense delivery. On the other hand, as creating quick wins will make the interest of the stakeholders active, they will be willing to provide more input to the process, which will cause more organized negotiations. Therefore, there will be enough time to put effort to develop the product. In other words, initiator will not be lost in negotiation and delivering a design nonsense product (De Bruijn, Ten Heuvelhof, & In't Veld, 2010).

2.2.5 Conclusion

This chapter elaborated the stakeholder management methods and techniques used in the political environment together with the stakeholder analysis techniques to identify the power relations in between. Project managers have different drivers than the politicians. They have to deliver a high quality end product within time, cost and resource limitations. Therefore, the techniques offered in literature may not be directly implementable on a software project. This chapter provided insights to software development project managers that are dealing with multi-actor environment on how to apply stakeholder management techniques in the negotiation session, also what are the points they should avoid or consider by mentioning several dilemmas with their solutions.

Next chapter will investigate to what extend the suggested literature is used in a given software development case study, whether they were successful or not. If not, how these methods could be applied in the case study to deliver better results. If the suggested approaches are used and worked out in the case study, the suggested methodology will be justified. If they are not experienced in the case study but suggested to be used in a software development project with multi-actor environment (specifically different sense of urgencies take place), then the approach will be validated with an expert opinion through several interviews.

CHAPTER 3: CASE STUDY PART I

Stakeholder Analysis and Management



3 Case Study: PART I - Stakeholder Analysis and Management

The results of the literature survey made in Chapter 2 shows that although there is significant amount of research about the methodologies used for the development phase of the software in the literature, there is no focus on the stakeholder analysis and management of the software development projects and no case study with a client/stakeholders with different sense of urgencies.

In chapter 3 and 4, a case study about a simulation software development project, where there is continuously changing requirements and a client, inherits more than one organization (stakeholder) with different sense of urgencies and power relations take place, will be analyzed. Therefore, this case study is a valuable resource to observe and gain insights not only the software development phase of a project but also stakeholder analysis and management phase of a project. Mentioned (above) characteristic of this project, makes the insights from the case study a good contributor on answering the research question: *“How can software projects with unclear requirement definitions and stakeholders with different sense of urgencies be managed successfully?”*

Chapter 3 will focus on the stakeholder analysis and management aspect of the project. After introducing and describing the project from the stakeholder management perspective the analysis will focus on what kind of methodologies and approaches are used on managing the complex stakeholder environment and to what extent the chosen methodologies succeeded. The analysis will be an input for Project Management Guideline.

The insights from the case study will be gathered via interviews with the project team members such as project manager and consultant. The research also includes the researcher’s own insights from the project as she was the member of the team and actively worked on the development of the software (Action Research: 1.5.3).

3.1 Introduction

The main actors in the case study are between the client (product owner), contractor (manager) and the programmers (team). The client of the project in this case study is a high-tech company, implementing medical equipment. The client needs solution for the capacity planning of their workforce planning and management. Contractor of the project is the consultancy company, which proposes and implements a solution for the problem of Client. The main deliveries of the project is a simulation software to support Client’s decision-making on their Workforce Planning and Management strategies and a process design that ensures the integration of the model into Client’s business planning process.

The purpose of this project is to find out how to use the capacity of the engineers for the corrective and preventive maintenance of the Client machines that are located at several places (hospitals). Client was inquiring how to forecast the machine breakdowns. There are so many factors impacting the machine breakdowns such as the type of machine, how many engineers are trained on that specific type of machine, the distance of hospital, where the breakdown occurs, when the breakdown takes place, and so on. Therefore, it is complicated to estimate how to schedule the breakdowns. Dealing with all those uncertainties and designing their workforce capacity planning was the main issue of Client (Fumarola, 2015). This case study is about a capacity-planning project to test out different scheduling scenarios and finding which approach is more robust. The solution proposal of the contractor is developing simulation

software that gives insights to Client on their decision-making for their workforce capacity planning and finding a balance between their workforce supply and demand.

The insights given on this chapter are based on the documents prepared by the consultant of the project and the interview made by the project manager, project consultant and the software development consultant. Additionally, the researcher's own insights also takes place in this chapter as she actively involved and took part in the project. All the interview reports and the researcher's own insights can be found in the Appendix: (Chapter **Error! Reference source not found.**).

This chapter will start with introducing the project stakeholders, and then stakeholder analysis will be made. Then, the power relations between the stakeholders and the challenges encountered related to power dynamics will be analyzed and stakeholder engagement plan will be explained. Finally, the stakeholder management techniques used in the case study will be described.

3.2 Project Stakeholders

This section introduces the project stakeholders that are project developers (software development company, project manager (consultancy company) and the client of the project with several organizations and different sense of urgencies underneath. Figure 5: Stakeholder Map gives the stakeholder map and the relations in between.

3.2.1 The Project Owner: Accenture

Accenture is a global management, consulting, technology services and outsourcing company. It opts for delivering high-performance businesses and governments to its clients with more than 305,000 employees in around 200 cities and 56 countries all over the world. The company generated \$30.0 billion net revenues for the fiscal year 2014 (About Accenture, 2014).

In this Project, Accenture is in charge of consulting Client on developing a capacity planning simulation software tool to support their decision-making on their Workforce Planning and Management processes. As it is the stakeholder, who is responsible for the delivery, Accenture is the project owner in this case.

The consultancy agency in this project composes of the following stakeholders (Sengur, 2015):

- *Project manager:* He is the direct contact between the client and the software development company. He creates a solution proposal to the client together with the software development company.
- *Project consultant:* He is the key contact of the project as he arranges and supervises all the meetings with the client and the project team (software developers). He conveys the information between the client and the software developers; he has a mediator role in this project.
- *Analysts:* They are the resources that are performing the tasks needed to achieve the end delivery under the supervision of project consultant.
- *Researcher:* She is both observer and the part of the team. She actively involved in the development of the software. Her observations from the project are used as insights in this chapter.

3.2.2 The Software Development Company: MACOMI

Macomi is a start-up company working on simulation software development projects with a team of 6 people. The role of Macomi in this project is developing the software itself with three programmers and one simulation software consultant. Macomi collaborated directly with Accenture to provide the delivery to the client (Sengur, 2015).

3.2.3 The Client / Contractor

Client is a diversified technology company, producing appliances in the field of Healthcare, Consumer Lifestyle and Lighting. The company is specialized in cardiac care, acute care and home healthcare, energy-efficient lighting solutions and new lighting applications (Company Profile, 2014).

Client required a capacity planning solution for their Workforce Planning and Management processes in order to reach their business plans. As they do not have the capability to build simulation software, they asked Accenture to give a consulting service to enhance their Workforce Planning and Management solutions to reach their business goals.

There are three critical stakeholders within the contractor of this project:

1. The Central Market: Eindhoven office of Client, where the central decisions are given. Central market is the initiator of this project, which creates the sense of urgency for the project.
2. US Local Market: US is the most powerful and the biggest local market because they are generating the highest value within the Client complex. The distinctive characteristic of this local market is its decentralized management system. In US, **engineers** decide when and where to go for the maintenance. Engineers make the workforces planning themselves.
3. Germany Local Market: Germany is another important stakeholder in this project as they have the most diversified workforce scheduling system from US, which is the most powerful local market. Unlike US, Germany has central planning for their workforce scheduling. It has **planners**, who decide which engineers are assigned to which locations and when.

Stakeholder Map of the Case Study

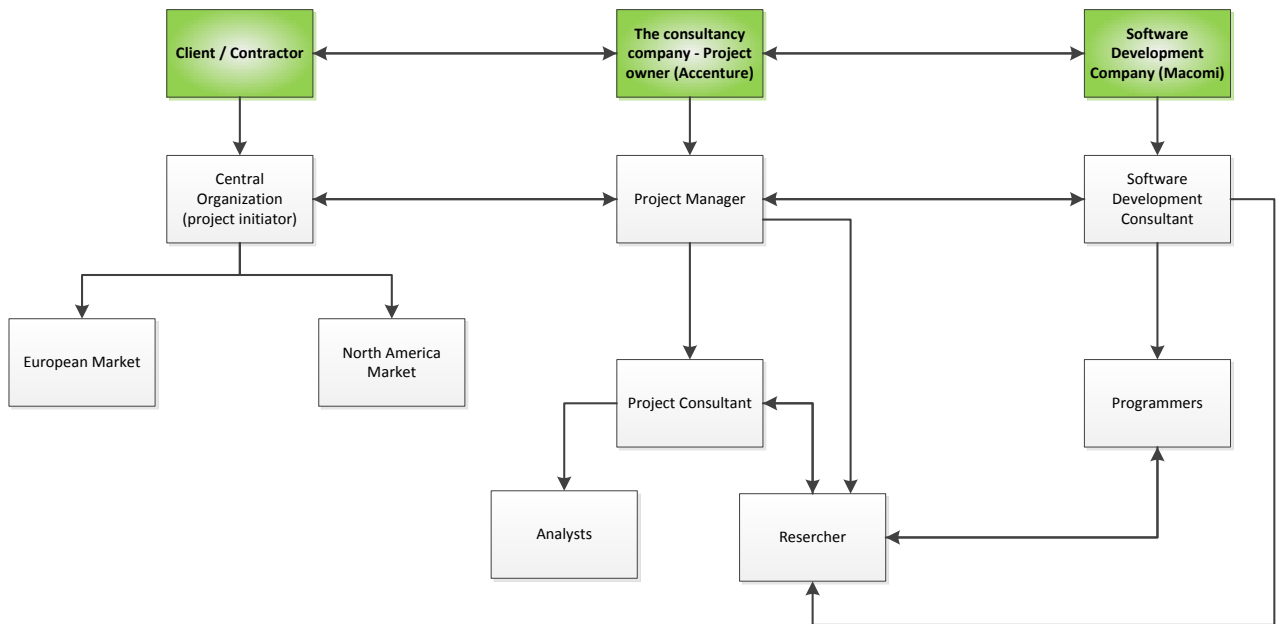


Figure 5: Stakeholder Map

The information written in the following sections (rest of the chapter) is based on the researcher's her own insights that she received from the project manager while working in the project. She interprets all the insights she has about the stakeholder analysis and engagement according to the terminology used in the literature.

While planning the project, project owners did not call the terms used by Muray et. al., (Murray-Webster & Simon, 2006) but they used the same approach to identify the power relations between the stakeholders and determine which one to involve into the software development process. Therefore, it can be concluded that stakeholder-mapping technique is used in the case study. The purpose of the decision makers in the case was involving the *sleeping giants* into the design and development process of the software as actively as possible.

3.2.4 Stakeholder Analysis

There are four main stakeholders in the project:

Accenture, who is the project owner and supports Client on reaching their business target on Workforce Planning and Management. Its attitude was positive to the project because they aimed to help Client (Central Market). Their interest was also positive as they were making business. If they succeed the project, it is possible to gain the trust of the client, results in more business opportunities with the same company, so increase in their market share and reputation. Good reputation means higher market share all around the world. However, their power was quite limited in this project. Accenture used their expertise to show the right way, convince other stakeholders and have their consent about the designed/developed solutions but they were not the decision makers for the implementation and the adaptation of the solution. Therefore, Accenture can be counted as a '**Friend**' type of stakeholder in this case

because they have positive attitude and interest to the project but did not have the power to execute it.

Central Market is the client of the project, who asked help from Accenture on supporting their Workforce Planning and Management decisions. In order to reach their business target, they wanted to decrease the costs resulting from Workforce Planning and Management within their organization. Therefore, they opted for having more optimal solutions in that field. Central Office is the head of the company, where the administrative decisions regarding the all offices around the world are taken. In that perspective, they are counted to be powerful on making decisions. However, as they are bound to the local markets on generating the revenue, Central Market is not the monopole on implementing the given decision. They need the support of the other local markets, especially the ones that has the biggest market share. This power balance between the Central market and the Local Markets was the tricky part of this project because they did not always have the similar sense of urgencies. Hence, it needed several strategic behaviors and managerial expertise to draw the attention and having the consent of the sleeping giants to execute the given decisions by the Central Market. That makes this case an attractive one for our research.

The attitude of the Central Office is positive, as they are the initiator and also the client of the project. Beside their power and the attitude, their interest is also positive because if they succeed this project, they will be able to shrink the cost in Workforce Planning and Management area within their organization and reach their business target. Regarding this information, it can be concluded that Central Office is a '**Savior**' type of stakeholder in this project.

North America is one of the local markets of Client, which has the biggest market share and the one generating the highest revenue. This gives North America the highest sanction power upon the administrative decisions made by the Central Office. Above all, North America was one of the pilot markets in this Capacity Planning Simulation Development (CSPD) project. Therefore, it would be nearly impossible to implement the project without the active involvement (such as data sharing and regular feedbacks) of North America. Therefore, it can be concluded that North America has a significant power on the implementation and adaptation of the project.

The attitude of the North America can be counted as positive because they were not objected to a new Workforce Planning and Management tool but their interest was different than the Central Market. While Central Market required a tool that supports decisions on decreasing the costs, North America was not interested in so much about decreasing the costs. They were giving more priority to increase the efficiency, so the customer satisfaction. Therefore, the expectation and the requirements of these two markets from the capability of the software tool were different.

The interest of North America was negative at the beginning of the project because they had different priorities than the Central Market and were unsure about to what extent the developed tool would satisfy their needs. Additionally, they already had their backup tool that they used internally on their Workforce Planning and Management solutions and were satisfied with it. They did not know whether the new tool, proposed and developed by Accenture would surpass the capability of their backup tool. Moreover, as they had their own system that they accustomed to and satisfied with, they were reluctant to change and adapt to a new system.

Considering the given attitude, power and the interest of North America, it can be confidently asserted that North America was a '**Sleeping Giant**' for this project and the one that should be invested hard.

Germany was the other pilot market in this project that required having active involvement. Their power was positive, as their involvement was significant for the development of the project. Additionally, they were a local market, so were the one generating revenue. Although their market share is not as big as North America, they still have a power upon Central Market. Considering their market power, they could refuse to involve into the project or adapt it. Therefore, it was another market that was supposed to be invested significantly.

The attitude of Germany to the project was similar to the North America: positive but different interests. They also had their back-up solution for their Workforce Planning and Management decisions, so were reluctant to the new tool at the beginning of the project. Therefore, their interest was also negative likewise North America. The tricky part with the involvement of Germany was their expectations and the requirements were different that both Central market and the North America. Although they had similar interest with North America (increasing the efficiency and the customer satisfaction), their workforce and scheduling policies were different. Hence, they had different requirements and expectations from the tool. With the involvement and the importance of Germany, there become three poles (decision makers) with different interests and sense of urgencies in the project that needed to be managed collaboratively and turned out a single solution that serves the needs of all stakeholders.

Considering the positive attitude and power but negative interest of Germany market, it can be said that it was a **'Sleeping Giant'** in this project.

Beside the four important stakeholders, including Accenture, there were other stakeholders that could be considered as **'Acquaintance'** type. One of them was United Kingdom (UK) that passively involved into the project. Although the Central Office was eager to involve UK office more actively into the development of the software tool, UK was very reluctant to the project at the initial iterations. As UK was not one of the pilot countries in this project, the project owner and the client did not wait the input of UK to start the development of the project. UK started to show interest to the project at the final iterations, as they understood the capability of it better but their contribution and also benefit from the tool remained shallower compared to the pilot markets. The adaptation of the tool to their needs took longer time compared to the others as they gave their input at the late phase of the project.

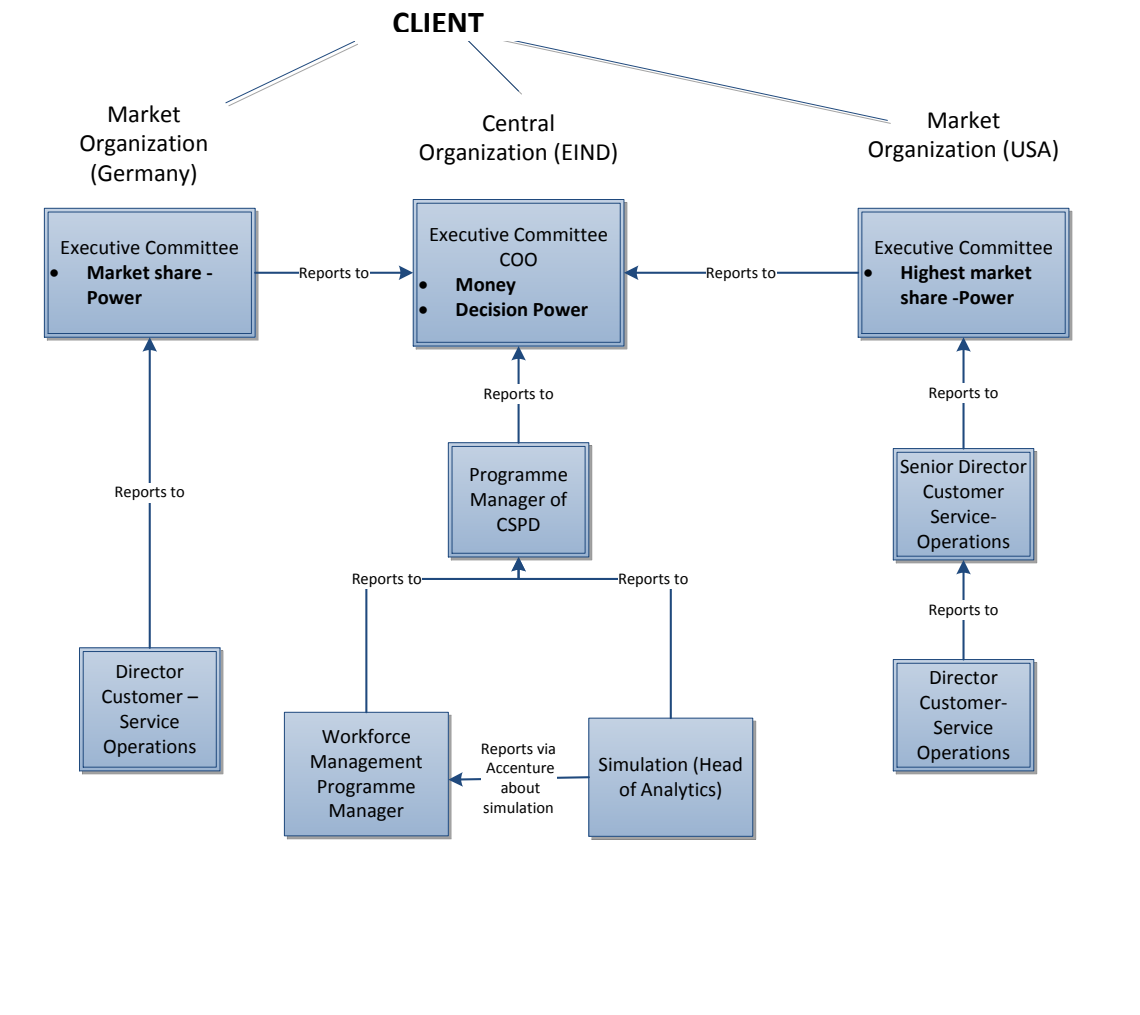
Another **'Acquaintance'** type of stakeholders was the local markets of Client that the tool is aimed to be adapted to in the long run. Their interests are still negative to the software tool because they do not have urgency and they are reluctant to change. Their power for the development of the tool was insignificant. On the other hand, they were not opposed to this project, so their attitude can be counted as positive. The software was developed in a flexible way considering the possible additional needs of other local markets may come across in the future. However, the adaption of the tool to these markets is not in the scope of this project.

	Area of Interest	Attitude (+/-)	Power (+/-)	Interest (+/-)	Stakeholder Type
Accenture	Aims for supporting Client on making more optimal decisions on their Workforce Planning and Management process	+	-	+	Friend
Central Market	Aims for decreasing the costs	+	+	+	Savior

– Client (The Netherlands)					
North America - Client (Local market)	Aims for increased customer satisfaction	+	++	-	Sleeping Giant
Germany - Client (Local market)	Aims for increased customer satisfaction	+	+	-	Sleeping Giant
UK - Client (Local market)	Aims for increased customer satisfaction	+	-	-	Acquaintance
Other Local markets - Client	Aims for increased customer satisfaction	+	-	-	Acquaintance

3.2.5 Power Relations between the Stakeholders

Besides analyzing the type of the stakeholder, the power relations between the actors are substantial information to conduct the right strategy against each actor. The scheme below, Figure 6 gives the information of detailed relations between the active stakeholders in this project and the basic difference between the central market and the local markets. The project was conducted and managed based on these differences and the power relations in between (Wenzler, 2015).



Differences Between the Central and Local Markets

	Central Market (Eindhoven)	Local Market (Germany + US)
SPEED	FAST (6 Months of Deadline)	SLOW TO MEDIUM (No Deadline-Substitution)
KNOWLEDGE (Requirements)	HIGH Knowledge	VERY LOW Knowledge

Figure 6: Relations between the stakeholders in the client side

3.2.6 Engaging the Stakeholders

As could be seen previously, stakeholders have different goals and interests and do not need to be involved in the process in the same way. As such, it is important to consider different approaches and roles for the stakeholders (Wenzler, 2015). In the following Table 7 an overview of the strategy recommended per stakeholder is given concerning the level of cooperation necessary for the initiative to successfully be implemented.

Attitude	Power	Interest	Type	Strategy
+	+	+	Savior	Empower
+	-	+	Friend	Collaborate, Consult
+	+	-	Sleeping Giant	Involve, Inform
+	-	-	Acquaintance	Inform

Table 7: Stakeholder Engagement Strategies

The table given below describes what kind of actions can be taken with different kind of strategies that will be applied to the stakeholders. It also gives information about who are the responsible actors for the given actions:

Strategy	Responsible Actor	Action
Inform	Accenture	We will keep you informed
Consult	Central Office	We will keep you informed, listen to you, and provide feedback on how your input influenced the decision
Involve	Central Office	We will work with you, ensure your concerns are considered and reflected in the alternatives considered, and provide feedback on how your input influenced the decision
Collaborate	Accenture	We will incorporate your advice and recommendations to the maximum extent possible
Empower	Accenture	We will implement what you decide

Table 8: Stakeholder Engagement Actions

The project manager noted that they used the following steps in their stakeholder engagement by finding an answer to each question below (Wenzler, 2015):

1. Who are the stakeholders involved in this project? – They make a list of the stakeholders
2. What type of person are they? – They use social style techniques for that (expressive, driving, amiable, and analytical) See the journal of Snavelly et. al., (Snavelly, 1981) for further insights.
3. What are their expectations?
4. What is needed to approach them?

Project manager noted that they engage their stakeholders according to the answers of those questions together with all the other analyses done in this section.

3.2.7 Conclusion

In this section, stakeholder analysis and engagement strategy of the case study is described. The interests of different stakeholders and their power relations between each other are described. This information is used to develop a robust stakeholder management methodology in the project (Wenzler, 2015). The next section, the stakeholder involvement and management strategies of Accenture on the planning phase of the software development will be evaluated. They mainly used two general approaches for the stakeholder management of the project:

- Two-poles approach
- Multi-issue decision-making

These approaches are selected based on the stakeholder analysis done in this section. They will be explained thoroughly in the next section.

3.3 Stakeholder Management Methodology I – Two Poles Approach

In this section, the stakeholder involvement and management strategy of Accenture on the development of the software will be evaluated and compared with other possible scenarios.

Before this evaluation, stakeholder involvement strategy of the project will be discussed. Then, the approach of Accenture on involving the stakeholders and the strategy they followed will be evaluated. Finally, their approach will be compared with other possible scenarios that could be implemented but was not chosen.

3.3.1 Stakeholder involvement Strategy of Accenture and Scenario Analysis

All the insights in this section (3.3.1) is taken from the software development consultant of the case study (Fumarola, 2015). The workforce scheduling approach of all other local markets within Client complex are somewhere in between these two stakeholders: US and Germany. As only one software is aimed to be developed for the capacity planning of all local markets that all have different scheduling system, Accenture needed to find a smart way to develop the tool (Fumarola, 2015).

Different possible approaches to develop the software could be (Fumarola, 2015):

- *Selecting one pilot country:* designing and implementing the software according to the requirements of this pilot country. Involving other countries after pilot country starts using the tool.
- *Selecting one giant country and few other countries that have similar scheduling pattern:* Involve and adapt the software to the requirement of the countries that have opposite pattern after the software is already started to be used in other countries.
- *Selecting small local markets, developing the software based on these countries:* After implementing the tool in these countries, involve giant actors latest and adapt the software gradually for the needs of big stakeholders.
- *Two-poles approach:* Selecting two giant pilot countries that has the most diversified scheduling system and requirements. Designing the software according to the requirements of the most diversified countries and adapting others after the tool is being implemented in the pilot countries.

Accenture selected the last (forth) scenario to develop the simulation software for Client although in most cases selecting one pilot stakeholder, implementing the solution in one party and then spreading out to other places is a more common approach. The central office of

Client's sense of urgency impacted to giving this decision. Central office created a sense of urgency by raising the issue of their ineffective workforce planning. They wanted to find a solution to their workforce planning system as an earliest possible time and to be implemented in all Client offices in the world. This initiation of Client urged Accenture to find a different and more creative approach. Although starting with one pilot country is a safer solution, it was going to take a lot of time to adapt other countries. Once the tool is developed, it is difficult and takes time to adjust new and diversified requirements (Fumarola, 2015).

On the other hand, although managing the stakeholders from two countries is more difficult than one pilot country in the development phase, once the tool is developed, it is a smooth process to adapt the solution to other countries.

The two pilot markets are selected for the development of the software: Germany and North America. The reason of selecting these two countries is the divergence on their scheduling characteristics and culture. It is expected that if the software is designed based on these two pilot markets, it would be easier to adapt other markets to the tool as their scheduling character fall somewhere in between.

The United Kingdom market was also attempted to be involved into the project in the initial phases. However, their lack of interest in the initial iterations of the project caused them to be involved in later stages. Therefore, the contribution of the UK market to the product design was minor.

The complexity of this project was the power relations between the stakeholders. Client is headquartered in The Netherlands, so it has the managerial power at their hand. However, other markets such as US and Germany are the ones who are generating the revenue. Although the central office (Eindhoven) seems to be the most powerful market in principle, US is practically the most powerful market as it is the one generating the highest revenue.

The aim of the central office is increasing the efficiency and decreasing the costs of the workforce planning and management, so they decided to receive consultancy to support their business plan. The purpose of the Central Office was to ensure that all Client offices around the world are using the same program in order to reach the highest efficiency on the capacity planning of their workforce. However, the senses of urgency of other offices (local markets) were different than the central market, which caused them to show reluctance on involving in the project. Also, it would difficult to manage all countries in six months of time period. Therefore, Accenture decided to select two pilot markets to start the implementation: US and Germany. Although this strategy eased the scoping the project significantly, it was still difficult to involve these two markets actively into the project because of the characteristics of those countries:

- US and Germany both had their own back-up solutions for workforce planning and management and were not open to change.
- Moreover, they did not have adequate insight on the capabilities of the Capacity Planning Model, resulted in inadequate attention to the project.
- Also, the Central Office did not have enough enforcement on other markets because US and Germany are the biggest markets generating revenue and do not accept all the decisions given by the Central Office.
- US and Germany has the most divergent scheduling system. That means, they are two stakeholders that have the most divergent requirements. Therefore, requirement

setting sessions were chaotic and difficult to align compromise on the prioritization of the requirements.

- This approach starts implementing the project with the most powerful and the biggest stakeholders in one go. The first implementation of the tool is always the most difficult part of the project because the benefits of the software is still not visible, requirements are not set, so the uncertainty is high at the beginning. Involving the biggest stakeholders at the same time and at the phase, where there is high uncertainty requires robust management capability. Therefore, it carries higher risk of failure than implementing the software on one pilot country and adapting other countries gradually.

Regarding these challenges, it seems like a risky approach to choose the most powerful and divergent countries first to the project. It might be thought that big players are the most difficult to get inline within a six month of project duration. Therefore, the failure chance of the project seems more possible. However, after analyzing other possible scenarios, it is found out that the chosen strategy is the most feasible solution for this kind of project, where many stakeholders with different sense of urgencies take place. The comparison of possible scenarios will be done in the next section.

On the other hand, the advantages of selecting the most divergent and biggest players can be ranked as below:

- Client needed a rapid change on their workforce planning as their existing system was causing a lot of cost to the company. There was no time to execute the project in one pilot country first and then gradually adapt it to other countries. The approach of Accenture speeded up the process of spreading the tool to all stakeholders. The software is developed based on the countries that have the most divergent requirements. As the requirements that come from other countries are always in between the most divergent countries, adapting the tool to other countries went very smooth. The software did not require big adjustments or change. That caused a robust and fast adaption process of the tool to all stakeholders.
- Once the bigger players apply the solution, other countries saw the benefits of the software already before they adapted the solution. They saw how much cost is saved thanks to the tool. Their sense of urgency changed rapidly and other stakeholders are urged to adapt the software within their organization.
- As the process of adapting the tool was fast, Client started to save bigger amount of cost in early phases of the project than a possible one-country pilot project scenario.
- The approach of focusing on not more than two countries at the software implementation phase speeded up the implementation phase. If there were more stakeholders involved in early phase of the project, they were going to cause more chaos and conflict that might have caused delay Accenture having more delays and conflict that might have been come through other stakeholders.
- As the most powerful countries are involved to the integration process first, the cost saving effect is higher than a scenario that adopts less powerful countries first.
- Once the bigger parties start to use the software, they can enforce less powerful countries to use the tool as well.

- The regret cost is lower if either Germany or US do not accept the project when they are involved first in the project.
- The ambition of Client was implementing the solution in all countries as quick as possible. By selecting the most divergent two countries approach the tool is implemented according to the requirements of the most divergent countries, therefore software did not need further adjustments because of undefined requirements. Only, different capacity planning approaches are tested in different locations based on their scheduling approaches. This speeded up the adaption of other countries to the developed software.

3.3.2 Results of the Two-poles approach

This section discussed different possible approaches on the stakeholder involvement. Based on the comparison of all possible scenarios and the insights from the project manager and the software consultant, it can be concluded that Accenture chose the best strategy regarding the given conditions: The solution is required to be implemented in all local markets in a shortest possible amount of time, under certain budget and with a six months of software planning and development duration.

Additionally, according to the interview that was done with the software consultant of the project (Fumarola, 2015), the adaptation and execution of the software in other locations went very smooth. After implementing the simulation software according to the requirements of the most diversified countries, adaptation of other countries was not very difficult because all other countries' requirements were somewhere in between the requirements of US and Germany.

After US and Germany start using the tool and implementing the workforce capacity planning based on the outcomes of the software, the benefits became visible. They reduced significant amount of cost thanks to the tool developed by Accenture. As the other countries saw the benefits of the software from US and Germany, they become much eager to use the tool. In fact, the demand to the tool from other countries is such that the capacity of the existing machines (laptops) are having difficulty to meet the demand. In eight months five more countries adapted and started using the software. The rest is still in progress on adapting the software.

Lastly, Client central office is happy with the speed of the process and the results of the project based on the saved cost.

To conclude, these results gathered from the project consultant justified the success of the chosen approach. That means, the real results supports the conclusion that selecting the most diversified and most powerful stakeholders is the best approach based on the given conditions.

3.4 Stakeholder Management Methodology II – Multi-issue Decision-making

Multi-issue decision-making is applied to involve the most critical players into the decision-making loop. Based on the stakeholder analysis and the decision to use "two-poles approach", North America and European markets of the client is identified as the most critical stakeholders of the project. However, their sense of urgencies were different than the central market, which have been described in the previous section 3.3. Therefore, the project manager used multi-issue decision-making method to be able to involve these two countries into the software development process and negotiation sessions for the requirement set up. Making an alignment between these two countries were quite challenging because they had the most divergent scheduling system, therefore the most divergent requirements within all other stakeholders. On the other hand, as the client did not have a good understanding about the capabilities of the

end product, their expectations were not always realistic. When realizing all these challenges, project manager decided to use multi-issue decision-making methodology. He talked to each market that are critical for the project and made a list of expectations. He put all the wishes and expectations on the agenda. Stakeholders started to have a rather more interest to the project as their wishes were taken into account, so they accepted to sit on the same table with the consultant firm and the process started. As it can be remembered from the literature, the most important priority in the multi-issue decision-making method is starting the process. Once the actors are involved into the process it is difficult to leave, as they would jeopardize the purpose of gain of other stakeholders. They will not want to harm their relations with other stakeholders because they are committed to each other and have coalition in other projects even if they are in conflict in this one. The situation was exactly the same in the case study. The sense of urgencies and the interests of the US and European market were in conflict and different from both each other and the central market. However, they could not leave the project once they are involved, as they did not want to harm their relations with the central market (Wenzler, 2014).

Also, after they involved to the process of the software development, they started to have a better understanding about the end product, so they realized that some of their wishes were unrealistic and abandoned their unrealistic wishes in later stages of the project. Additionally, they started a trade-off between each other for the prioritization of the requirement implementation. Out of those negotiations, it is decided which requirements to implement in the next iteration. Project manager supervised all these sessions and directed the negotiations (Wenzler, 2014).

3.5 Results and Conclusions

Multi-issue decision-making methodology used in case study was effectively implemented (Wenzler, 2014). By not refusing the wishes/expectations of the stakeholders, instead by putting their all wishes into the agenda at the beginning of the project and then discarding unrealistic ones during the implementation phase of the project via negotiations and trade-offs resulted in involving the most critical stakeholders into the loop. In that sense the multi-issue decision-making strategy worked successfully in the case study.

Moreover, two-poles approach: *“Selecting two giant pilot countries that has the most diversified scheduling system and requirements”* is a new technique that has not been applied in the reviewed literature. However, this approach composes the basis of the management methodology used in the case study and both project manager and the software consultant confirmed that this approach is the inevitable part of the success of the project (Wenzler, 2015) (Fumarola, 2015). Therefore, the insights gathered about this methodology from the case study will be the part of software project management guideline. The software development experts will validate the approach at the Validation of the Guideline.

As the two-poles approach is new to the literature, there is no generalized information about the benefit of this approach. Based on the insights from the project manager (Wenzler, 2015) (Fumarola, 2015) the benefits of the two-poles approach can be generalized in the following way:

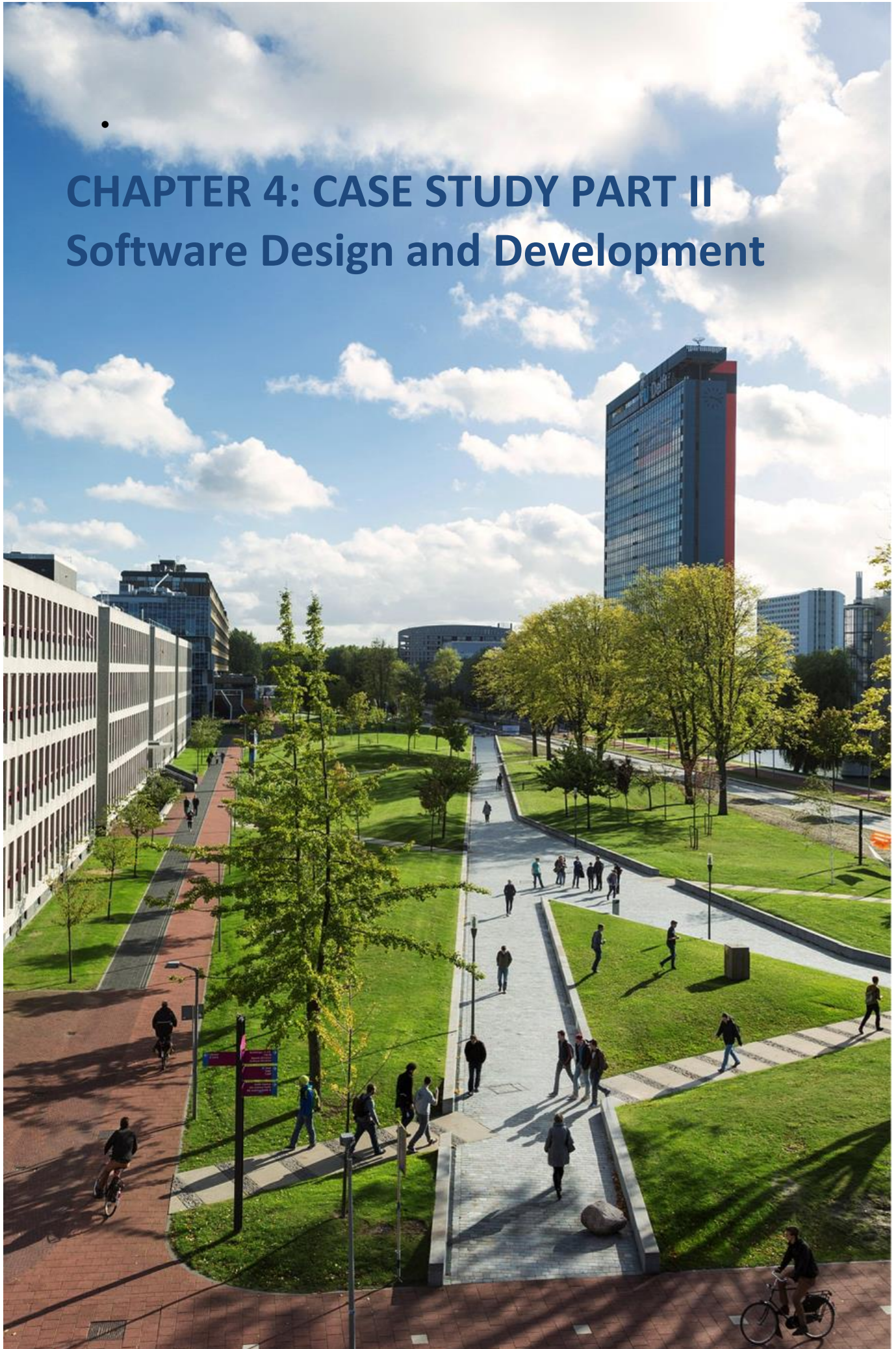
The Benefits of two-poles approach: Involving the biggest fighters into the loop first:

If the biggest fighters are managed to compromise, it is easier to align compromise with the less powerful stakeholders because of the following reasons:

- If the proposed solution has already implemented by the biggest and the most diversified players and worked, it will be easier to convince the other parties to involve because the benefits of the new system will be visible. Their perception about the proposed solution will be like: "If this solution works with the biggest players, it will work with rather smaller parties too!"
- The smaller players will want to follow the pattern that bigger players adopt. People have a tendency to follow the successful and powerful examples (bandwagon effect)
- If the biggest players are already committed to the new system, small players will feel the pressure to adapt it as well or big players can use forceful power on the smaller once. Smaller or less powerful players will not resist so much to accept the new system because they will not want to harm their relations with the bigger players. It is likely for them to need the support of big players on other issues in the future.
- Earlier the bigger players are involved into the project, less costly to give up the project. If the biggest player cannot compromise, it will be realized at the first step before other countries are involved. The regret cost will be lower. No matter how many stakeholders are convinced to adapt the new solutions, if the biggest players do not accept to involve, the project will fail and will not be implemented. The regret cost will be much higher if the biggest players are involved to the project at the end and do not accept the solution.
- Earlier the biggest players are involved into the project, quicker the return on investment from the project is gained. The impact of the biggest players on the outcome is usually higher than the smaller players. Therefore, if the biggest stakeholders start implementing the solution at an earlier phase, the impact of the project will be felt earlier.
- If the biggest players join the project at first, the project will be designed based on the inputs and the requirements given by the biggest players. That means, it is higher chance to convince the biggest players to implement the solution. Once the biggest players implement the solution, it is easier to involve smaller actors because of the reasons mentioned above.

CHAPTER 4: CASE STUDY PART II

Software Design and Development



4 Case Study: PART II - Software Design and Development

As it has been mentioned in previous chapter, chapter 3 and 4 is analyzing a case study about a software development project with continuously evolving requirements and complex stakeholder environment in order to gain insights for supporting the research question of this study: *“How can software projects with unclear requirement definitions and stakeholders with different sense of urgencies be managed successfully?”*

The topic of the case study is already explained in Chapter 3 but if it is rephrased it can be summarized like this: The case study is about a simulation software development project that aims providing insights to the client of the project on their decision-making on allocating the right amount and quality of their workforce capacity to find the balance between the workforce supply and demand. In that way, they could be able to shrink the costs on their workforce planning and management side and contribute to their business goal.

As it is explained in Chapter 3, the client of the project does not have the capability to build the tool they need, so asks consultancy from Accenture to help them. Accenture, together with a simulation software development company, Macomi, develops a simulation software tool to the client that simulates different workforce capacity planning scenario and provides insights to the user about the effects of the chosen scenario on the KPI levels.

In chapter 3, the stakeholder analysis and management aspect of the project has been analyzed to gain insight for the second part of the research question: managing stakeholders with different sense of urgencies in a software development project. It explained the stakeholders involved to the project, their power relations, encountered problems related to the power dynamics and stakeholder management approaches used in case study.

Chapter 4 will take it one step further: It will describe how the software is developed and the used methodology to achieve this end product. Chapter 4 will analyze how the used methodology for the software development managed the changing requirements in each iteration. The analyses in chapter 5 will also give insights on how the chosen software development methodology changed the sense of urgencies of the actors during the software development process and how the involvement of the stakeholders to the process was affected from that. With these insights both parts of the research question will be touched: managing the changing requirements and managing different sense of urgencies. Another important detail is that insights gathered in Chapter 4 mainly refer to the planning phase of the project, while insights from Chapter 5 mainly refer to the development and implementation phase of a software development project.

As the focus of this chapter is the adaptability of software development methodology on changing requirements, this chapter will start with a thorough requirement analysis, will continue with the capabilities of the simulation software tool. Then, the project planning will be explained. Finally, the methodology used to implement the software will be described with the implementation technique of the requirements and its effect on the involvement of the stakeholders.

4.1 Problem Definition of Client and several observations from the organization:

Before the problem is carried to Accenture, following observations are made in Client (Accenture, 2014):

- Within the Field Service Organizations, maturity of the workforce planning and management is relatively low.
- Workforce planning and management capabilities are limited at the local level and mostly respond to an ad-hoc need.
- The capacity of the analytical capabilities is reaching their boundaries for many organizations and mostly bound to traditional tools (mostly excel-based).
- A step-change improvement is needed in workforce planning and management capability, which requires a dedicated approach supported by an advanced tool-set.

Regarding the observations, Client highlighted the following business issues (Accenture, 2014):

- What is the expected workload resulting from the changing demand?
- What is the required capacity level to meet the business objectives?
- What is the consequence of changes in process, system or capacity on business performance?

4.2 Offer from Accenture/Planning Phase of the Project

Considering their business issues, Client consulted Accenture to bring a solution on their Workforce Planning and Management decisions. Accenture analyzed the observations and needs of Client, understood their expectations. Then, Accenture made several internal meetings to decide what kind of solution they can provide to Client and how feasible their expectations are. In order to describe their offer, Accenture arranged a workshop with other stakeholders to present their solution plan. The content of this first workshop was identifying the requirements of a mature workforce management capability, supporting by the concrete case studies. Accenture put forward the commandments to success on Workforce Planning and Management based on their experiences to identify what a contemporary workforce planning and management is. These commandments are used as a base to give a shape to the general requirements of the proposed solution. The delivery of the project was designed considering these requirements and the business plan of Client.

10 commandments for success in Workforce Planning and Management will be described together with the requirements that gave shape to the design of the final delivery. These 10 commandments can be found in the Appendix Chapter **Error! Reference source not found..** Based on the commandments of a successful Workforce Planning and Management, Accenture made Client the following offering to reach their business goal (Accenture, 2014):

Detailed simulation models for:

- Strategic Workforce Planning
- Capacity Planning and Scheduling
- Capacity Optimization
- Business Planning
- Process Optimization

This chapter will provide information how the software is implemented and which techniques and methods are used during the project issued in the case study. First, the capabilities, limitation of the final delivery will be discussed.

4.3 Capabilities and the Limitations of the Final Delivery:

- The final delivery is planned to be scheduling software, which aims to dispatch jobs. However, it is not aimed to tell the best scheduling scenario. It only supports the decision-making of the scheduling team on selecting the right concepts to be used (Accenture, 2014).
- The software enables the exact prediction of the future based on the given input. Again, it does not tell the best scheduling scenario, it only helps understanding and managing the variability of different concepts (Accenture, 2014).

4.4 The Simulation Tool

The simulation tool was aimed to give detailed insights to the decision makers in the organization. Accenture planned a four-month project for Client to develop simulation software that gives insights to the organization for supporting their decision-making on workforce solutions. The insights obtained from the tool was purposed to support decision makers at Client to find out the answers of the following questions (Accenture, 2014):

- What are the optimal resource staffing levels for Client (locations/qualifications/working hours/...)?
- What kind of steps should be taken to optimize the organization and what would be the impact of this solution on the staffing levels/KPIs?

The software tool does not aim to tell the optimal decision. It provides detailed insights to the organization on supporting their decisions by comparing different workforce scenarios. The software tool is designed to give outputs for workforce-planning scenarios created by the users. Based on the created scenarios, the decision makers can give the optimal decisions considering their business plans (Accenture, 2014).

Some of the benefits that the workforce model should bring to Client over the coming years are mentioned below (Accenture, 2014):

- It should enable to meet the customer demand better.
- It should improve the long-term capacity planning.

4.5 Project Planning and Requirement Analysis

This section will give detailed information about the software development methodology of the project and how the chosen approach helped adapting the changing requirements and involvement of the stakeholders.

If we refer to the research question, the purpose of this study is proposing an approach for the management of software projects with unclear requirement definitions and stakeholders with different sense of urgencies. In order to give a recommendation, the situation of the case study and the used methodology should be discussed and evaluated. As the research question stresses the unclear requirement definition characteristic of the software project, in this section first thing to elaborate is how the requirements were set and evolved during the phases of the project. And then, we will figure out to what extent and how the selected methodology helped to manage the dynamic requirement environment.

4.5.1 Project Planning

The project with Client is planned in four iterations. The project was planned to grow in the following way: The first iteration was aimed to match the demand and supply, in the second iteration the all market specific characteristics were embedded. In the third iteration, the scope of the job roles was increased. In the fourth iteration, the capacity planning tool was finalized. The details of the plan for each iteration and the model output are given in Figure below. After each iteration, Accenture scheduled a workshop with all the decision makers in Client. Based on the plan presented at the Kick-off meeting, the consultant analyzed and presented how far they were in line with the project plan. The output of the previous iteration was also presented and discussed in these workshops. According to the feedbacks from the client and the negotiations between all stakeholders, new requirements are built to give a shape for the development of the tool in the next iteration (Sengur, 2015). The project planning of each iteration and deliverables of the project can be seen in the Appendix, Chapter **Error! Reference source not found..**

Before discussing the methodology used in the software development, requirement setting will be discussed. The reason of discussing the requirement setting is showing how requirements changed iteration to iteration to understand the existing situation of the project.

4.5.2 Requirement Analysis:

The requirement setting was a serious and detailed process in the project as it forms the core of the software development (Sengur, 2015).

Before each iteration, negotiation sessions are arranged with all stakeholders to set the requirements that will be implemented in the next iteration. The requirements set at the initial phase of the project were rough and on the high level. Once stakeholders are more involved into the process and understand the capabilities of the designed tool, requested requirements became more realistic and precise. In order to avoid the ambiguity, in the first iteration requirements that are the most basic and overlapped by each stakeholders are executed. In each iteration, requirements are evolved and changed. The process is adapted the changing and volatile requirement environment of the project (Sengur, 2015).

In the project, Accenture used a formal definition for the requirements. They were initiated based on the needs of the stakeholders involved (US and DE) but written at the specification level. Before the implementation, the requirements that are decided to be implemented are signed by the actors that has role setting these requirements. In this case, both US and DE offices have their signature under the requirements they set together with Accenture. After then, they are executed on the software development (Accenture, 2014).

Two different types of requirements were used in the project: User requirements and design requirements (Accenture, 2014):

4.5.2.1 User Requirements for the Capacity Planning Model:

Main User requirements: Defined per iteration as an outcome of the workshops and validation meetings.

Detailed User requirements: Collected during the programming process and validating the certain steps by the markets. Progress of incorporation in the tool will be structurally tracked and transparent reported towards the markets.

4.5.2.2 Design Requirements:

There were three main design requirements of the developed software (Accenture, 2014):

- **Flexibility:** The product shall embed market specific characteristics (not only for the pilot markets, but also non-pilot markets).

Therefore, the software was needed to be developed flexible enough to adapt the characteristics of different markets.

- **Develop Capability:** Supporting process should be developed to ensure embedding in the organization.
- **Transparency:** Mismatches with current resource staffing should be visible.

The figure in the Appendix, Chapter **Error! Reference source not found.** shows how the requirements evolved from iteration to iteration in different phases of the project life cycle:

4.6 Methodology used in the Project

Accenture used agile approach to manage the needs of different stakeholders in this project. The software is implemented iteratively and incrementally, which helped the project owner to adapt the process to the changing and dynamic requirement setting (Sengur, 2015).

This section discussed the used methodology in detail by explaining how effective it is on involving the stakeholders with different sense of urgencies to the decision-making process and how responsive to the dynamic requirement environment.

4.6.1 Iterative and Incremental Development - Agile Approach

In order to bring a feasible solution based on some parameters such as cost and efficiency, Accenture follows an iterative and incremental approach for this project. As it is mentioned in the literature survey chapter 2.1.7, the concept of building a system/software through iterations is called iterative and incremental development (Larman, 2004). In this case, as exemplified in the literature (Larman, 2004), the partial system grows incrementally by adding up new features to the system in each iteration sequentially.

It is mentioned in chapter 2.1.7, Iterative and Incremental Development (IID) is a flexible project management approach that enables modifications on the requirements (Dingsøyr T. D., 2010). Iterative and incremental approach that was used in this project is similar to the one schemed in the book of Craig Larman (see in Figure 9). Referring the chapter 2.1.7, according to Craig Larman, iterative development is a management approach that splits the overall life cycle of the project into several iterations in sequence and enables building the software as self-contained mini projects each compose of several activities such as requirement analysis, design, programming and test individually. The delivery of each iteration is a stable, integrated and tested *partially* complete system. However, the final iteration delivery is the final product, which is ready to deliver to the client (Larman, 2004). It is seen the same pattern of development in the case study: The whole implementation was broken into several subsets of requirements thanks to the IID approach. After each subset, a feedback was taken from the client about the requirements. The expanded or adjusted requirements were used in the next subset that led process continues iteratively. One of the main advantages of IID in this project was that it was flexible through the steps so any changes on requirements did not have substantial effect at the end of the project (Sengur, 2015).

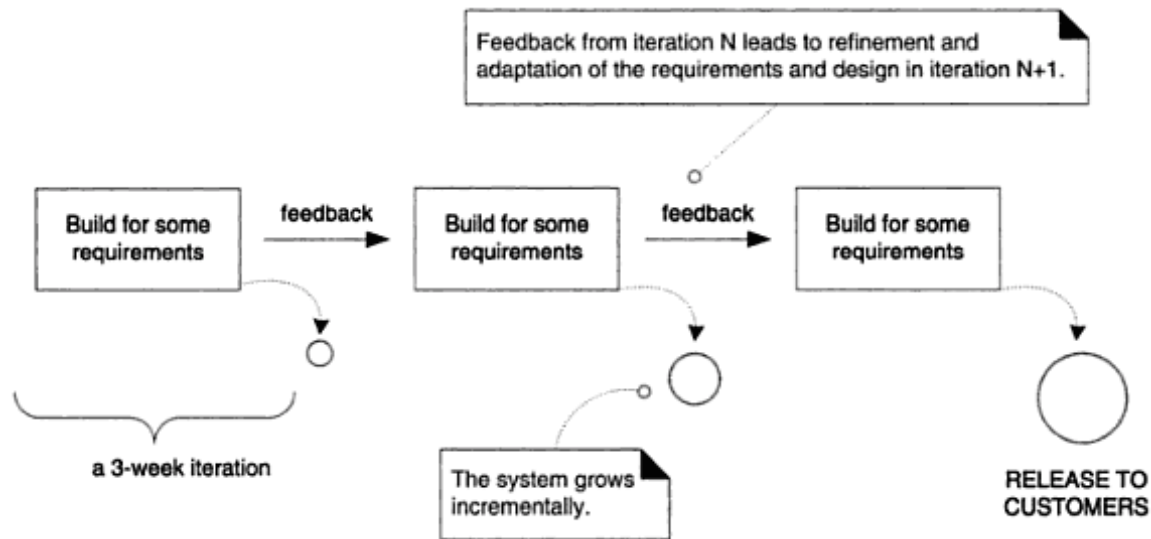


Figure 9: Iterative and Incremental Development (Larman, 2004)

4.6.1.1 Project Consultant's Insights: How IID approach is applied in the project?

In this project, the process was developed deeper (incremental) and broader (iterative) on each phase. For example, while only the “scheduling” part was developed in the first iteration, the second iteration included the “ordering” part as well. And the third included another part of the process and so on. This exemplifies the broadening approach. On the other hand, each process had sub-processes inside. In each iteration, the system was elaborated more in detailed. For instance, in the first iteration the scheduling process was handled considering only the machine, engineer, his availability and activity. However, on the second iteration the scheduling process included more parameters such as travel time and SWO operation. This shows the approach of going deeper on each iteration (Vloo, 2014).

The project consultant described the way of finalizing the project like this: The last iteration of the project is called **interpretation** phase. In the interpretation phase, it is checked whether all the questions that are prepared at the beginning of the project are answered thoroughly. If the questions cannot be answered totally, the team goes back to each iteration and keeps going broader and deeper through them again and again until all the questions are answered. The level of deepness and broadness on the development of the product depends on in which level the questions are answered in each iteration. Therefore, the team should ask on the interpretation part whether they have enough answer to reach their goals (Vloo, 2014).

4.6.1.2 Project Manager's Insights: How IID approach is applied in the project & what is the advantage?

In the following paragraph, it will be examined that to what extent the advantages of the IID technique described in the literature are experienced in this project.

According to the interview done by the project manager, it can be concluded that most of the insights of Greer et. al.'s were justified in this case study, while there are few differences. The following insights are written based on the interview made with the senior manager of the project (Wenzler, 2014), after all the iterations are completed and the end-product was the received the consent of the client and ready for execution on two pilot countries.

The requirements were prioritized in the project but not according to their importance on contribution. The requirements were prioritized according to how basic they are and approval of all the stakeholders in the project. The difference of this project from the ones seen in the literature was the availability of different interests and sense of urgencies in the client side. As all the stakeholders were supposed to use the same product, the consent of all stakeholders was needed before implementing the requirements. Therefore, Accenture prioritized the overlapped requirements that have the consent of all the stakeholders. Those were usually the most basic ones required for the development of the simulation (Wenzler, 2014). The second advantage of incremental software development approach, explained in the work of Greer et. al., significantly helped in this project as the active involvement of the stakeholders were crucial. At the beginning, majority of the stakeholders had a lack of interest to the initialized project. Therefore, it was quite challenging to draw their attention to the project and receiving the complete and the right input for the development of the software. Thanks to the iterative approach, the *partial* complete part of the subsystem was delivered to the client still at an early phase of the project. As they start understanding the output and its capabilities, their sense of urgency affected from that, so started to pay more attention to the project. The more they understood the project, the more actively they started to involve into the project. The more actively they contributed to the project, via feedbacks and transferring the input data, the more accurately the software program was developed. During the project, managers of the project owner (Accenture) encountered some challenges they did not estimate before. Managers needed a rapid and smooth decision-making operation on the planning to overcome those challenges. The advantage of the IID approach was here; the new decisions could easily be adapted to the process, without giving significant disruption. The requirements changed in every iteration and the nature of IID approach enabled inheriting the adjustments quickly and efficiently (Wenzler, 2014).

4.6.2 The role of Agile Approach on managing the different sense of urgencies and changing requirements in the case study

Figure 10 gives a complete overview of the interests and sense of urgencies within the different markets of the client. The insights in this figure about the interests and sense of urgencies of different stakeholders are taken from the interview with the project consultant (Vloo, 2014). The figure also gives information how could agile approach deal with the emerging problems related to different sense of urgencies and changing requirements. As this interview is done still during the implementation phase of the project, the information related to the US is invalid anymore “USA still compare their own product with the new model. They partially adopted the new model.” At the later stages of the project, USA is convinced that the new software is better than their back up tool. They provided more input and involved more in the project to get the highest benefit from the tool. At the end of the project, they fully adapted the developed software by Accenture (Sengur, 2015).

The agile approach helped on (Sengur, 2015):

- Adapting to the continuously changing requirements.
- Involving the stakeholders to the development process of the project
 - Thanks to the meetings after each iteration, the insight of the stakeholders on the product increased, they started to understand to which purpose the final product is served.
 - Consultants gained the trust of the stakeholders as they see the evolution and the benefit of the product.

- Therefore, the stakeholders started to be more eager to involve into the development of the product by giving more feedback, providing more and correct input data.



Difficulties in the project and How Did Agile Approach Help?

- Directors of USA and Germany did not know about the simulation at the beginning of the project. Also, they already had their back-up solution and they did not want to change it. As a result, they were reluctant to the project.
- Although Executive Committee of USA has a lower position than Executive Committee of Central Office (Eindhoven) in the hierarchy, he has bigger to say (more power). Therefore, Executive of the central office has knowledge about the simulation and he is enthusiastic about it, he approves the requirements as long as executives of USA and Germany approves them.
- The understanding (knowledge) of the Head of analytics in the central office was high on the project. Therefore, she was pushing the directors of the USA and Germany to speed up but they and their managers were reluctant.
- Central market had a 6 months of deadline but the local markets had no deadline. Agile approach helped to speed up the project and get the trust of the local markets.
- In order to involve Germany and USA into the project, a step by step (iterative and incremental) approach is implemented.
- Because local markets were unable to understand the capability of the project, their requirements were unrealistic. Accenture had to implement the most basic requirements at the very beginning of the project. This approach gave some time to local markets to understand the model and its capabilities.
- After each iteration the understanding of the local markets increased and their requirements started to be more realistic. They started to trust the model, understand its importance, and involved more into the project.
- At the end, Germany understood that the new simulation model is better than their own model (back-up). They accepted and adopted the new model totally.
- USA is not very open to change. They still compare their own product with the new model. They partially adopted the new model.

Figure 10: Interests and sense of urgencies of different markets at the client

4.7 Mediator:

The mediator of this case study is consultancy firm, particularly project consultant and the project manager as they provide the communication between the client and the programmers. Project consultant took the scrum master role in this project; therefore he arranged all the meetings and negotiation sessions (Sengur, 2015).

As this is a large-scale global software development project, the scrum master used the agile practices and some of the agility supported distributed activities mentioned by Paasivaara et. al., (Paasivaara, Durasiewicz, & Lassenius, 2008). For more detailed information about these activities see Chapter 2.1.10.

He arranged the following agile practices and agility supported distributed activities (Sengur, 2015):

1. Daily scrum meetings
2. Weekly scrum-of scrums
3. Synchronized 4-week sprints
4. Sprint demos
5. Retrospective meetings
6. Unofficial distributed meetings
7. Site visits

Because of the time difference between the different offices among the stakeholders and distances, the scrum master used teleconference tools in the meetings and arranged separate meetings according to the time zone of each stakeholder (Sengur, 2015).

4.8 Dilemmas: Negotiated nonsense vs design nonsense and openness vs closeness

In the case study, the project manager dealt with the dilemma between developing negotiated nonsense vs design nonsense software in the following way:

The project manager used “creating quick wins strategy” on the development of the product. Requirements from US and Europe market are collected and the overlapped requirements are prioritized and implemented at the first iteration of the project. In this way, both countries had win situation from the first iteration. They had to chance to see the results and their trust to the project increased and also their interest as well. They started to provide more input to the project, so the developed software fits to the needs of the stakeholder better, which led to avoid developing negotiated nonsense product. Although the negotiations were the trickiest part of the project because of diversified requirements and expectations and also different sense of urgencies, this challenge is dealt with in an effective way thanks to the quick wins strategy. As both markets see the result from the initial phases, they started to be willing to attend the negotiations more actively, which causes negotiation session more efficiently. In this way, management team could put enough effort on developing sensible software, so they could avoid design nonsense product (Sengur, 2015).

Project manager avoided openness vs closeness dilemma by keeping the scope of the negotiations narrow. There were enough topics in the agenda to keep the key stakeholders (US and Europe markets) in the loop. For instance, the workforce planning of Europe market is centrally managed while US market has a decentralized scheduling system. The software product should have been interesting enough to both parties, which have completely opposite scheduling system to keep them in the process. After aligning with the software development

team project manager from Accenture made sure that the proposed software product is adaptable to both parties' needs, offers a solution beyond the capabilities of their back-up system and have an added value to their workforce planning. Therefore, the proposed project was appealing to both parties because they could find a prospect of gain for themselves (openness). On the other hand, the promise of the project manager was not unmanageable with respect to the available time, cost and resources they had. For instance, client's intention was to have a system that gives the best workforce planning solution. This expectation was unrealistic because the best solution might change from country to country. Instead, Accenture proposed a model that simulates different scenarios and gives the responsibility of choosing the best solution to the client based on their own criteria. Such kind of boundaries prevented the negotiation process being sluggish and away from a solution (closeness). With this balance, project manager avoided the two extremes of the openness vs closeness dilemma on managing the process (Sengur, 2015).

4.9 Conclusion

Agile approach was successful on managing the changing requirements in the project and increasing the understanding of the stakeholders about the end product. They understand more about the benefit of the software and their sense of urgencies changed and came closer to the initiator of the project (Central Market). For that reason, they provided more input data and started to be more enthusiastic on joining the meetings and contributing the development phase after each iteration.

4.10 Reflection of the Case Study

Based on the results taken from the project, although Accenture chose the most difficult approach among the analyzed options, regarding the alignment of the requirement setting (choosing two pilot countries that have the most diversified requirements), the process went very smooth. There was no problem on adapting the changing requirements in any of the iteration of the project life cycle thanks to the adaptability characteristic of agile approach to the changing requirements. From that perspective agile approach succeeded in this project. However, in many projects seen in the literature, agile approach does not work successfully. As a reason of the problem, the communication between the software developers/programmers and the client is given. The reason of the success of agile approach in this project is that there was mediator between the client and the programmers, who translated the capabilities of the developed product to the client and the needs of clients to the software developers. Accenture took the mediator role in this project. Accenture provided effective communication between the programmers and the client. Thanks to the effective communication between the client and the programmers both sides could able to understand their needs and capabilities, what is feasible and what is not. There were no issue because of the communication problem. The wishes/needs of the client were translated to the requirement by Accenture (mediator) in the specification level. Any technical detail such as programmers' feedbacks regarding the constraints and the capabilities of the product is communicated with the client on the high level, in a non-technical way. Therefore, the intended requirements could be able to implemented in the software, in given time limitations, which is mostly not the case in the literature.

4.10.1 Reflection by the software consultant

While adapting the changing requirements went smooth in the project, according to the feedback of the software consultant that he gave in the interview (Fumarola, 2015) the negotiation part was the most difficult part of the project. As there were several stakeholders

with different sense of urgencies, setting up a decision process and having an effective result was the most challenging part of the project.

Also, as it is known that although the results and the used approach in the project resulted successfully, Accenture initially intended to involve UK into the process as well on the development phase of the project. However, UK denied involving the decision-making process, so Accenture had to discard UK from the initial phase of the project and focused on the most important two giant local markets. The question is, how could Accenture involve all the stakeholders they intend to involve to the decision-making process.

Moreover, according to the feedback received from the software consultant (Fumarola, 2015), the negotiation and alignment with the stakeholders was the most challenging and time consuming part of the project, which would be nice to work on.

For this reason, next section will investigate how could the literature be applied to the case study to deal with the challenges faced.

As the main challenge in this case study was about involving the stakeholders that have different sense of urgencies and have alignment with them, the research focus is on integrating the stakeholder management approaches discussed in the literature survey with the agile stakeholder approach that is successful on managing the changing requirements.

4.10.2 Reflection by the Researcher

After reviewing the literature on software development methodologies and stakeholder management techniques together with actively working in and observing the case study project about developing a software on a multi-actor environment with different sense of urgencies, researchers have the following reflections on the project. This section will have a focus that Accenture faced in the project and the researcher's suggestion based on the literature to Accenture to show how could the faced problem be managed more successfully.

What is the Problem with the Stakeholders?

The agile software development projects usually start with the stakeholders, who are really enthusiastic about the project. Therefore, the architecture of the system is developed according to the needs or requirements of the stakeholders, who are actively involved in the process and continuously giving feedbacks. However, there are usually other stakeholders, who are not as enthusiastic as the ones that involve in the project from the beginning. These unenthusiastic stakeholders do not set their requirements or give feedbacks in the initial sprints of the project. In other words, they do not give high attention to the project while the architecture of the system is developed. After the model is started to be used, they usually become eager to use the model as well but they come up with different requirements. Because the system architecture is developed according to the requirements of the stakeholders from the initial stages, the new requirements do not usually fit into the built model. Therefore, it is usually problematic for the project team to adapt the new stakeholders (countries) to the system (Fumarola, 2015). In this case study, the consultancy agency, Accenture, faced a similar type of problem (Sengur, 2015):

Problem: The intention of Accenture at the beginning of the project was starting the project with three countries: US market, European market and UK market. However, UK market was reluctant to give input to the project and join the meetings in earlier iterations. Therefore, the software developed based on the requirements given by US and Germany. The tool was adapted

to the needs of UK afterwards. **Question:** How could Accenture involve UK market into the project in earlier phases?

Proposed Solution:

What would be the right strategy to involve UK to the project in an earlier phase? Hans de Bruijn offers a strategy in his book for the stakeholders that resist involving the decision-making: “priming strategy” (de Bruijn & ten Heuvelhof, 2008). The purpose of priming is changing the problem perception of the actors. Stakeholders might have different problem perceptions from each other, and all competing for the priority. If a stakeholder has different problem perception, then the initiator has a problem. Even if they have the shared problem perception, other important issue that the initiator needs to concern is whether other actors have the similar sense of urgency against the problem. If not and the initiator wants to involve those stakeholders into the project or decision-making process, then he can succeed it by changing the perception of the actors to the problem. For example in this case, instead of describing the objective of the software precisely, Accenture could bring a broader perception to the problem that could draw the attention of UK as well.

When Accenture realizes reluctance from UK towards involving in such a project, instead of telling “this simulation software will help Client to determine the best workforce planning solution by simulating different scenarios and showing their effects on the KPI levels” Accenture could have used a more strategic approach on explaining the purpose of the project.

While presenting the project, Accenture could have served the purpose of the project in the way it serves their business goal (priming strategy). For example, they could have been said “this project aims for contributing to the goal of expanding the market share of Client by “X” percentage. If the KPI levels are improved, Client can approach this target. The proposed simulation software tool is helping Client to increase their KPI levels, therefore strongly advised each office to adapt and implement the tool on their workforce planning.” This kind of presentation would bring a completely different perception to Client towards the proposed software. It could have drawn the attention of UK because it is their business goal as well. They could have had more interest on joining the meetings and find out how this goal could be realized. As the proposed system is serving their business goal, the project would be perceived as high urgency by UK. With this kind of approach, Accenture would implement “priming” strategy that Hans de Bruijn suggested in his book by changing the problem perception of UK. By just serving the project in a slightly different way, UK’s perception towards the project would change and have the same perception and sense of urgency with the initiator, which is Eindhoven office in this case. With this new perception, Accenture would create an incentive for Client UK to join the project and gaining the support of reluctant stakeholder.

CHAPTER 5: PROJECT MANAGEMENT GUIDELINE



5 Project Management Guideline

After thorough analyses of literature about software development project management techniques and stakeholder management approaches, and also gathering insights from a case study, this chapter harmonizes all insights and offers a management guideline for software development projects having unclear requirement definition and stakeholders with different sense of urgencies.

Main stakeholders for this guideline are the client, users, contractor (project management team) and the software development team. The scope of the guideline is after the project is handed over to the contractor until the software is developed for the key stakeholders that are determined at the planning phase of the project. The main focus of the guideline is managing the stakeholders that have different sense of urgencies while making sure to have sufficient progress is achieved, which meets the expectations/requirements of all key stakeholders.

The guideline starts with a flowchart Figure 11 that explains the process, rank and the owner of each methodology offered in this chapter. Then, application of each methodology will be described in separate sections.

There are three stakeholders that are the owners of the activities/methodologies presented in the flowchart: Client, Project Manager (Contractor) and Software Development Team.

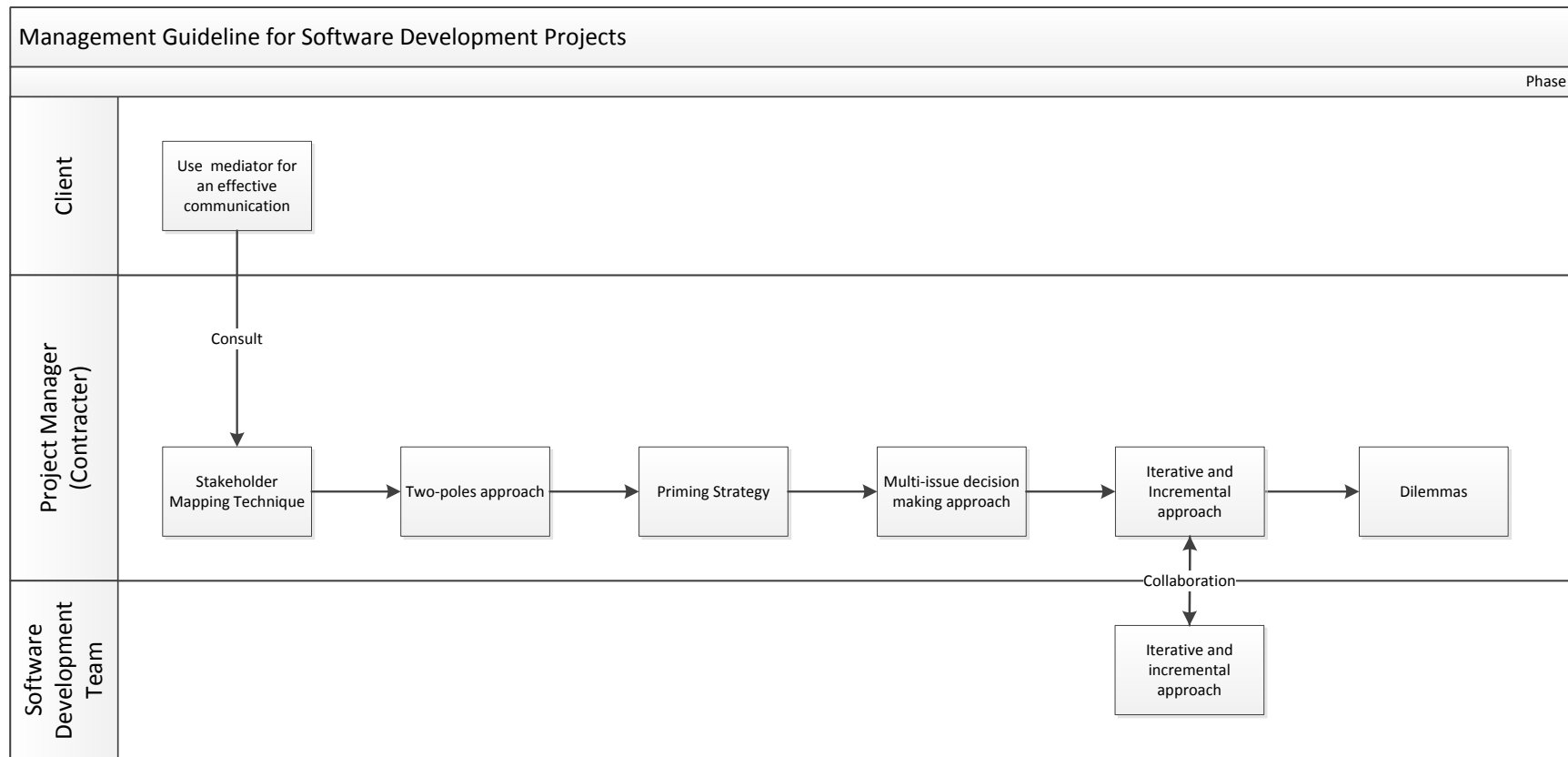


Figure 11: Management Guideline for Software Development Projects

5.1.1 Stakeholder Mapping Technique:

The first methodology to plan a software development project should be “Stakeholder mapping technique”. It is used in the preparation phase of the project by the project manager (contractor). Stakeholder mapping technique is needed to understand the capabilities of each stakeholder that will take advantage of the software and the power relations in between. It helps to identify the stakeholders that have productive and obstructive power.

In stakeholder mapping technique, the characteristics of the stakeholders should be identified, which will later be used to decide which ones to prioritize. Therefore, the project manager should classify the involved stakeholders according to the following classifications. These classifications also give tips to the project manager which strategy to use for the belonged stakeholder:

- **Saviour** – powerful, high interest, positive attitude or alternatively influential, active, backer. They need to be paid attention to; you should do whatever necessary to keep them on your side – pander to their needs.
- **Friend** – low power, high interest, positive attitude or alternatively insignificant, active, backer. They should be used as a confidant or sounding board.
- **Saboteur** - powerful, high interest, negative attitude or alternatively influential, active, blocker. They need to be engaged in order to disengage. You should be prepared to ‘clean-up after them’.
- **Irritant** – low power, high interest, negative attitude or alternatively insignificant, active, blocker. They need to be engaged so that they stop ‘eating away’ and then be ‘put back in their box’.
- **Sleeping Giant** - powerful, low interest, positive attitude or alternatively influential, passive, backer. They need to be engaged in order to awaken them.
- **Acquaintance** –low power, low interest, positive attitude or alternatively insignificant, passive, backer. They need to be kept informed and communicated with on a ‘transmit only’ basis.
- **Time Bomb** - powerful, low interest, negative attitude or alternatively influential, passive, blocker. They need to be understood so they can be ‘defused before the bomb goes off’.
- **Trip Wire** – low power, low interest, negative attitude or alternatively insignificant, passive, blocker. They need to be understood so you can ‘watch your step’ and avoid ‘tripping up’.

Outcomes of the stakeholder mapping technique can be summarized in a table like below:

Stakeholder Name	Area of Interest	Attitude (+/-)	Power (+/-)	Interest (+/-)	Stakeholder Type

First column represents the name or role of the stakeholder. The second column gives information what is the real interest of the given stakeholder from the project. Third column shows if the stakeholder has a positive or negative attitude towards the project. Forth column shows the power of the involved stakeholder. This is important information because powerful actors can have a higher influence on the outcome of the project. Fifth column shows if the

stakeholder is interested in the project or not. If a stakeholder is not interested in the project, even if it is the most powerful actor, it will not have an effect on the project because it has no interest in the project. The final, sixth column shows the type of stakeholder. The project manager based on the information on other columns should fill this column. He should decide in what classification stakeholder is put, whether it is sleeping giant or friend etc.

After deciding which stakeholder belongs to which type, project manager should decide what kind of strategy to apply to each of them. Project managers can use the table below to decide what strategy to apply:

Attitude	Power	Interest	Type	Strategy
+	+	+	Savior	Empower
+	-	+	Friend	Collaborate, Consult
+	+	-	Sleeping Giant	Involve, Inform
+	-	-	Acquaintance	Inform

Next step is based on the decided strategy, project manager should decide what action to take against which strategy and who are the responsible actor from those actions. Project manager can use the table/template below to decide for instance, how to act “inform” kind of strategy takes place:

Strategy	Responsible Actor	Action
Inform		We will keep you informed
Consult		We will keep you informed, listen to you, and provide feedback on how your input influenced the decision
Involve		We will work with you, ensure your concerns are considered and reflected in the alternatives considered, and provide feedback on how your input influenced the decision
Collaborate		We will incorporate your advice and recommendations to the maximum extent possible
Empower		We will implement what you decide

5.1.2 Which stakeholders to involve: Two-poles approach:

After identifying, which parties have productive, and what are the power relation between the stakeholders via stakeholder mapping technique, next step is deciding which actors should be involved to the process. In a software development project with complex stakeholder environment and divergent requirements, project manager should make a selection among the stakeholders to give a prioritization. Prioritization is important on a software development project to decide whose expectations are met first and on which stakeholders’ interests should be invested most.

The purpose of the decision makers/project managers in such projects should be involving the **sleeping giants** into the design and development process of the software as actively as possible. For this purpose they should design the most optimal strategy to involve those actors into the

process. Based on the analyses in this research, it is concluded that two-poles approach is the most efficient approach to develop software in a complex stakeholder environment and changing requirements. When there are diversified requirements among the stakeholder, it is difficult to set up the requirements in the first iteration of the project. Therefore, it is recommended to select two powerful stakeholders, which have the most diversified requirements. As there will be very small overlap between the requirements of two selected actors, the requirement set up will continuously evolve in each iteration and the requirements will be prioritized and then implemented as a result of the negotiation sessions of the project.

Selecting two giant pilot stakeholders that has the most diversified requirement and systems requires designing the software according to the requirements of the most diversified stakeholders and adapting others after the tool is being implemented in pilot actors.

Benefits of the two-pole strategy:

If the biggest fighters are managed to compromise, it is easier to align compromise with the less powerful stakeholders because of the following reasons:

- If the proposed solution has already implemented by the biggest and the most diversified players and worked, it will be easier to convince the other parties to involve because the benefits of the new system will be visible. Their perception about the proposed solution will be like: "If this solution works with the biggest players, it will work with rather smaller parties too!"
- The smaller players will want to follow the pattern that bigger players adopt. People have a tendency to follow the successful and powerful examples (bandwagon effect)
- If the biggest players are already committed to the new system, small players will feel the pressure to adapt it as well or big players can use forceful power on the smaller once. Smaller or less powerful players will not resist so much to accept the new system because they will not want to harm their relations with the bigger players. It is likely for them to need the support of big players on other issues in the future.
- Earlier the bigger players are involved into the project, less costly to give up the project. If the biggest fighter cannot compromise, it will be realized at the first step before other countries are involved. The regret cost will be lower. No matter how many stakeholders are convinced to adapt the new solutions, if the biggest players do not accept to involve, the project will fail and will not be implemented. The regret cost will be much higher if the biggest players are involved to the project at the end and do not accept the solution.
- Earlier the biggest players are involved into the project, quicker the return on investment from the project is gained. The impact of the biggest players on the outcome is usually higher than the smaller players. Therefore, if the biggest stakeholders start implementing the solution at an earlier phase, the impact of the project will be felt earlier.

- If the biggest players join the project at first, the project will be designed based on the inputs and the requirements given by the biggest players. That means, it is higher chance to convince the biggest players to implement the solution. Once the biggest players implements the solution, it is easier to involve smaller actors because of the reasons mentioned above.

5.1.3 Priming strategy:

The purpose of priming is changing the problem perception of the actors. Stakeholders might have different problem perceptions from each other, and all competing for the priority. If a stakeholder has different problem perception, then the initiator has a problem. Even if they have the shared problem perception, other important issue that the initiator needs to concern is whether other actors have the similar sense of urgency against the problem. If not and the initiator wants to involve those stakeholders into the project or decision-making process, then he can succeed it by changing the perception of the actors to the problem.

If the project manager wants to change the problem perception of the stakeholder, he needs to learn the real interest of the stakeholder. He can find this information in the stakeholder-mapping table that is explained in the previous section. Afterwards, project manager should find a way to present and initiate the relevant stakeholder in the way that serves their business target. For that, one small addition can be done to the scope of the project and that part of the project can be marketed to the relevant stakeholder. In this way, the attention of the stakeholder can be drawn but on the other hand, the stakeholder can accept the rest of the scope of the project.

5.1.4 Applying multi-issue game in software development project environment having several stakeholders with different sense of urgencies

This approach is discussed in the literature survey chapter but as it has been only used in the political environment, its applicability on a project-based environment has not been discussed. The following paragraphs explain why this approach is necessary for a software development project having stakeholders with different sense of urgencies and how to apply. Again, as this approach is not applied in the case study and only applied in a political environment in literature, an expert opinion is needed for validation, which will be done in the next chapter.

Stakeholder involvement in every part of the project cycle is crucially important in software projects because the software is implemented based on the continuous input provided by the client/stakeholders. However, in most cases involving the client to the software development process becomes a major issue. The reasons vary:

- Software projects are very technical for many clients, their understanding is shadow on the final delivery and it's benefits. That causes stakeholders showing reluctance on involving and contributing the process.
- They resist on change because it is difficult. They have a tendency to keep their own system, which is already in use, even though the new system offers more efficiency or less cost, etc.
- They have conflicting interests with some of other stakeholders, so lack of belief on achieving their goals on the project. Therefore the project is not always interesting to them.
- They already have their backup system, they want to keep their old system, which they are used to or committed to.

Given the reasons above, involving the stakeholders to the development process of the software becomes a major difficulty. In order to increase the speed of their involvement to the process, multi-issue decision-making game that is introduced by de Bruijn et. al., (de Bruijn & ten Heuvelhof, 2008) can be a useful tool. The initiator of the project can make the project attractive to the stakeholders they want to involve by using the multi-issue gaming that is discussed in literature survey chapter.

Multi-issue game might be a useful tool to draw the attention of each stakeholder and involve them into the development of the software. If an incentive is created for each stakeholder, they will be willing to provide more input and feedback to speed up the process of the implementation.

How to Apply:

1. The initiator of the software development project asks each stakeholder he wants to involve into the decision-making/software development process to write down their expectations/wishes from the initiated project.
2. Initiator puts all the wishes/requirements written by the stakeholders into the agenda to make the project appealing for each actor and prevent them to make excuse not to join the process.
3. Initiator invites the stakeholders into the decision-making process for the software development.
4. Once they are in the decision-making process, they will not be eager to leave the process even if their initial wishes that are put into the agenda at the planning phase of the project are not totally realized. It is because they will jeopardize not only themselves but also other stakeholders' prospect of gain in the project once they decide to leave.

If one or few of the stakeholders decide to leave the process after their input is used in the progress, they will jeopardize the success of the project. The software will already been started to developed based on the input from these stakeholders. Once the stakeholders leave, the software developers will need to adjust the software, which will cause delay and cost in the project. That means, left stakeholders will jeopardize other stakeholders' prospect of gain. It is likely to have a conflict between stakeholders. They will not want to harm relations with other stakeholders as they will need their coalition and support on other projects in future to realize their gains.

On the other hand, once they involve into the process, their level of understanding about the project and the final delivery increases, they become more aware how the software serves to their interests, so more eager to provide input and involve the sessions for the requirement generation and/or adjustments. Also, after they have an understanding of the capabilities of the software, they will figure out what is feasible to implement and what is not. And, they will start to set more reasonable requirements in each iteration. That means they will abandon some of their wishes that had initially had and will be more productive on contributing the progress of the project.

5.1.5 Iterative and incremental approach – Agile Software Development Approach

Based on the insights from the literature survey in 2.1, it is recommended to use iterative and incremental/agile approach on a software development project, where there is unclear requirement definition at the initial stages of the project. Moreover, the results of the case study (see chapter **Error! Reference source not found.**) justify the effectiveness of the suggested methodology. Therefore, we suggest managers dealing with a software development project with continuously changing requirements to use iterative and incremental / agile approach.

The agile approach helps on:

- Adapting to the continuously changing requirements.
- Involving the stakeholders to the development process of the project
 - o Thanks to the frequent meetings, the insight of the stakeholders on the product increases, they start to understand to which purpose the final product is served.
 - o Gained the trust of the stakeholders as they see the evolution of the product
 - o The stakeholders started to be more eager to involve into the development of the product by giving more feedback, providing more and correct input data

Other mentioned benefits of agile approach (see chapter 3.2 for a broader explanation):

1. Improved Communication and Coordination
2. Quick Releases
3. Flexibility of Design Quicker Response to Changes
4. More Reasonable Process
5. Increased Quality
6. Better Customer Focus
7. Improved Focus - Better Prioritization
8. Increased Productivity
9. Better Morale
10. Testing First

Additionally, agile methodology utilizes IID in a way to simplify the project process by reducing the planning complexity, emphasizing the customer value and encouraging the stakeholder involvement and collaboration. The involvement of the stakeholder in the whole process of the development especially from the first prototype decreases the risk of unsatisfactory products at the end of the project. Since a feedback can be obtained from the stakeholders initially after the first prototype, it is still early enough to make necessary improvements until the final product. Furthermore, contact with the development team and the stakeholders by means of prototypes prevent the team to focus too much on the technological innovation rather than the real requirements (see chapter 3.1)

In summary, agile approach is a substantial art of a software development project with continuously evolving requirements because of the reasons mentioned above. As scrum is the most popular and recent agile methodology for the software development projects, it is highly recommended to the project managers to use scrum on the development of their products.

How to Apply:

The Scrum team starts the sprint with a detailed plan that lasts up to four hours. It ends with the sprint review meeting, where the stakeholders take part; review the state of the business,

market and technology. Another type of meeting in scrum is retrospective meeting. In retrospective meetings, the effectiveness of the team is evaluated; plans are made to enhance the quality of the teamwork. Lastly, daily scrum meetings takes place 15 minutes each day by the scrum team. The following questions are addressed in daily scrum meetings:

- What did I do yesterday?
- What will I do today?
- What difficulties (obstacles) do I have?

Scrum has three main deliverables: Product backlogs, sprint backlogs and burn-down charts. The backlog is for the client's requirements and the burn-down charts show the cumulative remaining work.

In terms of administration, Scrum methodology consists of three roles namely the team, the product owner and the scrum master. The team is the developer of the product for the product owner. Through the sprints, there is a close collaboration between these two groups until the final product. At the end of each iteration, the additive requirements are given as a feedback to the team for the next iteration. One of the unique properties of the scrum methodology among the others is the existence of the scrum master. The scrum master is the main link between the development team and the product owner. The scrum master also controls the progress of the development team in the progress.

If the software development project is on a global and large-scale one, additional agile and agility supportive distributed practices should be used in order to have an efficient practice of scrum methodology. As the scrum master initiates these practices, they will be explained in the next section under "mediator".

5.1.6 Mediator:

After the client consult a contractor, the consultancy firm assigns a scrum master for the project. This person (mediator) is called as scrum master, who is coordinating the communication between the product owner and the team members that are developing the product. Scrum master is arranging and actively supervising all the meetings between the team and the product owner. Scrum master is a part of agile approach, who arranges daily, weekly and monthly meetings. However, if the software development project is a global and large scale one, different practices are needed for better communication among the teams. For this, the following agile practices are suggested:

1. **Daily Scrum meetings:** These meetings can be arranged by using telephone conferencing and web-cameras. It is possible to share applications as well though it is not needed.
2. **Weekly scrum-of-scrums:** They are arranged once a week, which takes half an hour. One team member from each team and all scrum masters attend this meeting. Each team discusses what have been done last week and what is the plan for the next week.
3. **Synchronized 4-week sprints:** If there are several teams in a project they come together each month (4-week sprints), at the end of each sprint.
4. **Distributed sprint planning meetings:** they are divided into several meeting. Their number depends on how many local markets are included in the process. In these meeting, how scrum is applied in each team is discussed. Microsoft NetMeeting can be

used for the application sharing. These are the virtual meetings take place via teleconferencing.

5. **Sprint demos:** Meetings are for sharing the developed demos. For this purpose, teleconference and application sharing tools are used.
6. **Retrospective Meetings:** it is a one-hour meeting directly after the demos. The team, product owner and scrum master participate the meeting to discuss the done improvements. Questions to ask in retrospective: “What has been good during this sprint?”, “What has not been that good?” and “What kind of improvements could we do?”.

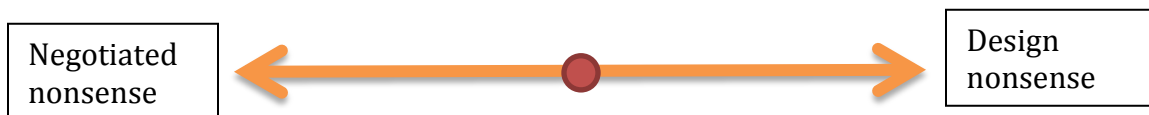
To enable better quality of work, better and higher quality of communication between the stakeholders and improved motivation among the team members, it is advised to scrum master/mediator to arrange the following agility supporting distributed activities:

- **Unofficial distributed meetings:** Through chat, email, teleconferences over the internet using headsets
- **Centralized version control:** It enables team members that are abroad to connect the server via VPN to check the code on a daily basis.
- **Site visits:** It is needed during first iteration to discuss how to apply scrum
- onsite system expert
- **Frequent visits to the site:** It is effective especially at the critical moments to collocate the team
- **Annual gatherings:** They are social events bring all the teams together to discuss the future of the product and effective for the team building.

5.1.7 Dilemmas faced in a software development project with multi-actor environment, having different sense of urgencies

Software development projects, having stakeholders with different sense of urgencies require significant amount of discussion and negotiation as all stakeholders need to align on a consent of what to implement and what not to. While putting effort on negotiation, initiator should not forget that he needs to deliver a successful product as an output of the project. Therefore, he needs to find the right balance in between. The following paragraphs are explaining the possible dilemmas an initiator will face and the way to deal with.

1. The first dilemma: Negotiated Nonsense vs Design Nonsense



If the initiator of a software development project gives the weight to the negotiated nonsense, there will be a risk of developing insufficient quality of software or end product. The reason is that the initiator puts excessive amount of effort and time on negotiation. As a result, all the stakeholders are satisfied with the tradeoffs and negotiation results but there is not enough time to improve the quality of the software or end product itself. On the other hand, if the

initiator puts high attention to design nonsense but little attention to the negotiated nonsense, there is a risk of building a robust and high quality software but the one none of the stakeholders like. It causes a non-implementable solution.

How to mitigate?

In order to deal with the dilemma, the initiator should find a balance between negotiated and design nonsense. By this balance, it will be possible to develop robust software that the stakeholders like and willing to implement. In order to keep the momentum and stimulate the progress, the initiator can use the design principles that guarantee progress, which are explained in page 25. The rules applied here can be interpreted to a software development project as follows:

1. Stimulate “early participation”:

In order to avoid “design nonsense”, project manager should make sure that the software is developed based on the requirements set based on the input from the user. Therefore, it is important for project manager to make sure there is an active involvement of stakeholders to the process. However, as simulation development and adapting the tool to the stakeholders require a change process and need active participation of all stakeholders for the successful development of the product, slow start characteristic of change process may apply here. The reluctance on involving into the design and development phase at the initial iterations is mostly because of different sense of urgencies among the stakeholders:

- The expectations of the stakeholders from the software product might vary or even be unrealistic (non-implementable).
- The parties that are required to involve may not have sufficient resource that provides regular input and feedback to the development team.
- The start date of the project may not be the ideal time for some stakeholders.
- They might be totally uninterested to the new product as they already have back-up solution and think that their solution is the best option for them.
- They may not be open to change because it is difficult.
- Etc.

In order to avoid or lessen the effect of “slow start” characteristic of a change process, mentioned in Chapter 4.8 (when there is a change process and a wide participation is needed to produce a successful product, the process is usually characterized by a slow start), the initiator should take some actions: It is important for the initiator to introduce arrangements to the parties at the start of the process and more importantly making the waiting game (joining the process at later stages) a less appealing option for parties. For instance, if one of the stakeholders does not provide their input at the first iterations, the software will not be developed according to the requirements of that party. This will take longer for this stakeholder to adapt the developed tool because the tool will be needed to adjust according to the new requirements. That causes extra time, cost and a delay on taking advantage of the benefits of the developed product. Especially if the software is aimed to reduce the cost of the organization or increase their revenue, a delay on taking advantage of the benefits may become a big loss for the stakeholder, who refused to participate at early stages. Initiator may point out the negative consequences of not involving to the process at the beginning to stimulate “early participation”.

Early participation of the stakeholders will make the negotiations more organized and mostly at the beginning of the project life cycle, so the team can focus more on the development of the

product at the rest of the iterations, which will enable avoiding to develop design nonsense product.

2. Creating Quick Wins:

In a software development project, it is inevitable to have conflicting demands because stakeholders have different systems from each other or expectations, which causes variety in requirements among stakeholders. In order to keep all parties in the process, the requirement implementation should be balanced for both parties. If only one party's requirements are prioritized and the requirements of the other party are left to the end, the second party can lose its interest and may have a tendency to leave the process. On the other hand, the other party, whose requirements are prioritized, may want to leave the process at the final iterations as their requirements are already implemented. However, in order to have healthy communication among the stakeholders and making negotiations efficiently, all parties should have sufficient interest and attendance during the whole life cycle of the project. Therefore, project manager should make sure to give quick wins to both parties at the beginning of the process to keep their interest active. On the other hand, he should not go too far away on giving quick wins because the rest of the process may no longer be appealing to some parties. This is the job of the project manager to find the right balance to give how much quick win to each party. In other words, the project manager should maintain that both parties' requirements are implemented in a balanced and logical order.

In this way, the project manager can effectively communicate and negotiate with both parties during the entire project life cycle and at the same time, the software can be developed efficiently due to the active involvement and input of the stakeholders in the entire project. As the software will be developed based on the continuous input of the users, project manager will avoid developing software that is negotiated nonsense. On the other hand, because the negotiation is done effectively, they can put enough effort to develop high quality software, which leads avoiding design nonsense software.

2. Second dilemma in a software development project can be faced is openness and closeness in negotiations.



Openness means the parties joining the process should be offered an incentive to join the process. Openness is applied to parties that are not easily persuaded to join the process. If those parties have a prospect of gain, the process will become more appealing to them. Openness is the strategy that is used in multi-issue game that is discussed in previous section. As described in multi-issue gaming section, project manager (initiator) should make the negotiation agenda broad enough that each key stakeholder can find something appealing to join the process. On the other hand, initiator should be careful on scoping the negotiation agenda. If there is too many points in the agenda, the negotiation may take too long, which may cause a sluggish process that will produce a clear result. In contrast, if the initiator (project manager) narrows the topics in the negotiation agenda too much (closeness), stakeholders may not find an interesting topic, so may refuse to join the process. Therefore, the initiator should find the right

balance between an openness and closeness on managing the stakeholders with different sense of urgencies in a software development project.

5.2 Conclusion

This chapter provided a guideline for the project managers, engaging with software development project that have continuously changing requirements and stakeholders with different sense of urgencies. Seven methodologies/approaches are suggested to tackle the characteristic challenges of a software project with given circumstances. All those methodologies are suggested based on the insights derived from the literature and the case study, supported by several interviews with the project consultant and manager together with the author's own insights from the project. Suggested methodologies/approaches in the guideline will be validated in the next chapter with expert opinion.

CHAPTER 6: VALIDATION OF THE GUIDELINE



6 Validation of the Guideline

In order to validate the proposed strategies, the following approach is used:

- Strategies used in the literature and defined as successful are observed.
- Strategies used in the case study and gave successful outcome are observed.
- Strategies used in the case study but failed are observed.
- A positive strategy used in the literature is matched with a failed part of the project and expert opinion is asked to validate the proposed strategy.
- If the strategy is successful both in the literature and case study, the proposed strategy is directly put into the guideline without having an expert opinion. It is considered a valid approach.

Table 2: Validation Strategy

#	Strategy	Observed & successful in Literature?	Observed and succeeded in Case Study?	Expert opinion needed?	Validated by expert?
1	Use “ stakeholder mapping technique ” to identify the power relations between the stakeholders and their characteristics.	Yes	Yes	No	
2	Involve the stakeholders that have the most divergent requirements: “ two-poles approach ”	No	Yes	Yes	
3	Use “ priming strategy ”: To make the project more appealing for the reluctant stakeholders, by influencing their problem perception.	Yes	No	Yes	
4	Use “ multi-issue decision-making approach ” to involve the stakeholders into the decision-making process in software development.	Yes	Yes	No	
5	Use “ iterative and incremental approach ” for the management of the software development tool because of its characteristics of adaptability to the continuously changing requirements.	Yes	Yes	No	

6	Use “mediator” for an effective communication between the programmers and the client.	Yes	Yes	No	
7	Dilemmas: “openness vs closeness” and “negotiated nonsense vs design nonsense”	Yes	Yes	No	

According to this strategy there are two methodologies suggested in the guideline that need expert opinion for validation:

3. Two-poles approach
4. Priming strategy

Still, all these approach written in Table 2: Validation Strategy, are validated by two software development manager.

The first expert is the project manager of the case study used in this research. He has extensive management experience with software development project having complex stakeholder environment. As he uses these techniques regularly in his projects within Accenture, he confidently approved and validated all the approaches mentioned in this research. Second expert is a software development manager, who has seven years of experience in this field. She approved and validated all the methodologies suggested in the management guideline. The interview details with the experts are given in the appendix.

Following sections discuss each methodology separately regarding if expert opinion needed to validate and what is the approach of the experts to these techniques.

6.1.1 Stakeholder Mapping Technique

As it is a successful approach in the literature and succeeded in the case study as well, no expert opinion required. It is considered as a valid approach on identifying the stakeholders. However, when it is asked in the interviews, both experts approved the approach.

6.1.2 Two-poles Approach

Two poles approach is validated by an interview that took place on 10th July 2015 (Wenzler, 2015) with the project manager of the case study. Interview took 1,5 hours and the questions that has been asked is listed in the appendix. The same approach will be used in a low voltage network project that has been run by the same manager with the same consultancy agency but with different client. The drivers of the project and sense of urgencies were similar to this project (details will be discussed in appendix). All the crucial factors that initiate the project is same. Therefore, it has been concluded that this low voltage network project is a good match to compare and contrast the case study used in this methodology in the sense of two-poles approach. According to the expert opinion, the results showed that the two poles approach succeeded in both projects. That’s why he concludes this approach is a successful one for the projects with unclear requirement definition and stakeholders having different sense of urgencies/expectations.

6.1.3 Priming Strategy

It has been seen as an effective methodology by both of the interviewees in order to change the problem perception of the stakeholders. Both approved the methodology.

6.1.4 Multi-issue decision-making

Multi-issue decision-making approach is a proven approach both by the literature and the case study results. Therefore, it is considered valid, no expert opinion needed. However, when it is asked in the interviews, both experts approved the approach.

6.1.5 Iterative and incremental approach:

This approach is considered as a robust methodology in the literature for the adaptability of continuously evolving/changing requirements in software projects and also succeeded in the case study in that sense, there is no expert opinion required for this approach. It is considered valid. However, when it is asked in the interviews, both experts approved the approach.

6.1.6 Mediator:

Mediator is a proved approach in literature for successful convey of the information between the programmers and the client. It also eased the setting of requirements and enabled a smooth communication in case study. Therefore, no expert opinion needed for validation. It is considered valid.

6.1.7 Dilemmas:

The dilemmas “openness vs closedness” and “negotiated nonsense vs design nonsense” is explained to the interviewees. Both confirmed those dilemmas in a software development project with complex stakeholder environment, so it is validated. Therefore, this is a factor that software development project managers should take into account.

6.2 Conclusion

Interview done by the experts validated two-poles approach, priming strategy and multi-issue gaming as successful methodologies for the given circumstances. As the rest of the approaches are already been considered as valid, it can be concluded that the methodologies given in chapter 5 is a valid strategic guideline for a project manager dealing with a software development project having continuously changing requirements and stakeholders with different sense of urgencies.

CHAPTER 7: CONCLUSION



7 Conclusion

In this chapter, the research question and sub-questions will be answered based on the insights gathered in this research and then a general conclusion will be given.

7.1 Answers to Research Question:

This research aimed to answer the following question: *“How can software projects having unclear requirement definitions and stakeholders with different sense of urgencies can be managed successfully?”*

To have a detailed answer to the research question, the following three sub-research questions are answered.

- What type of approach is needed to deal with different sense of urgencies in a software development project?

In order to deal with different sense of urgencies in a software development project, soft systems skills are required, where more communication among the stakeholders takes place. As the main purpose of software development projects is delivering a sensible product (software) within limited amount of time, budget and resources, project manager should use communication techniques that are target oriented. Project manager should know what outcome he wants to get out of the negotiations with stakeholders and purposefully manipulate the direction of these conversations to reach the outcome he planned in advance. In order to do that, project manager should have a structural plan on organizing the negotiations. Every negotiation session should have a purpose and project manager should effectively manipulate the direction of the discussion towards his own area of interest. If he cannot use the negotiation sessions cleverly and efficiently, the conversation might easily turn out to be sluggish processes that do not give any results. As there is limited amount of time in a software project and given the fact that the project manager has the responsibility of delivering high quality product, he should make the right balance between putting enough effort on negotiations and delivering and implementable product. When there is different sense of urgencies in a stakeholder network, it is more difficult to come up with quick results because each stakeholder expects their demands to be met, which will cause project manager lots of effort and time. This may jeopardize the progress of the software. Therefore, project manager should decide the key stakeholders in the project and focus on their interest, discard others from the decision-making sessions. This will give a clear focus to project manager and ease the orientation and manipulation of the negotiation/communication sessions.

- What type of approach is needed to deal with unclear requirement definition in a software development project?

When there is unclear requirement definition in a software project, requirements cannot entirely be set at the beginning of the project. The requirements are re-formulated and evolved in each phase of the project. When this is the case, continuous input of the stakeholders and new planning is needed in order to re-set and implement the new requirements. This requires iterative and incremental approach. The process should be designed iteratively and incrementally and requirements should be implemented gradually. With this approach, changing requirements can easily be adapted to the process and development of negotiation nonsense product can be avoided.

- What kind of methodologies could be suggested in order to successfully manage a software development project, having unclear requirement definition and stakeholders

with different sense of urgencies and which stakeholders would be the action owner of these methodologies?

1. **“Stakeholder mapping technique”** to identify the power relations between the stakeholders and their characteristics => By project manager
2. **“Two-poles approach”**: Involving the stakeholders that have the most divergent requirements. => By project manager
3. **“Priming strategy”**: To make the project more attractive for the reluctant stakeholders by changing the problem perception of a key stakeholder. => By project manager
4. **“Multi-issue decision-making approach”**: To involve stakeholders into the decision-making process in software development project. => By project manager
5. **“Iterative and incremental approach”** for the management of the software development tool because of its characteristics of adaptability to the continuously changing requirements. => By software development team
6. **“Mediator”** Find a mediator for an effective communication between the programmers and the client. => By client
7. Make a good balance between negotiation and high quality end product to deal with the dilemmas: **“openness vs closeness”** and **“negotiated nonsense vs design nonsense”** => By project manager

7.2 Main Conclusion

This research is searching an answer to the research question: *“How can software projects having unclear requirement definitions and stakeholders with different sense of urgencies can be managed successfully?”* Previous section gave a detailed answer to this question. Based on the insights derived from the literature and the lessons learned from a case study, seven different methodologies are combined and presented in the guideline (research sub-question R3 summarizes these methodologies). The purpose of this research is providing valid management methodologies to software development managers who are dealing with projects having unclear requirement definition and stakeholders with different sense of urgencies. Based on the learned lessons, iterative and incremental approach, most popularly agile methodology is effective to adapt the changing requirements to the project lifecycle and keeping the stakeholders in the process for continuously providing input. On the other hand, when there is a complex stakeholder environment, particularly different sense of urgencies take place, it is suggested to project manager to prioritize the key stakeholders and focusing on their interests when initiating the project. When all the stakeholders are kept in the loop, it is difficult to manage the process and derive conclusions from the negotiations. This will cause sluggish processes with no clear end result. However, as the project manager of a software development project has the responsibility to deliver negotiated and design sensible software within the limited time, budget and resources, he should efficiently manage the negotiations to be able to make sure that sufficient effort is put on the development of the software. The proposed guideline in this study suggests valuable methodologies in a logical and structured order to effectively manage the negotiations and at the same time progressing on the development of a implementable product/software.

Scientific Contribution of the Research: *Scientific contribution of this research is adapting stakeholder management techniques that are generally used in a political environment into software development project management environment and merging them with software*

development methodologies. The proposed guideline becomes a bridge between stakeholder management world (soft systems skills) and software development world (hard systems skills). By doing this, proposed methodologies in the guideline provide a solution both the changing requirements characteristic of a software development project and different sense of urgencies coming from complex stakeholder environment.

7.3 Reflections

Insights from the literature showed that the proposed software development methodologies offered by the literature remains insufficient when there is complex stakeholder environment with different sense of urgencies. More soft system skills and structured stakeholder management methodologies are required to be adapted to the software development projects to manage different sense of urgencies. The literature offers a bunch of study on software development methodologies to adapt and bunch of study on managing different sense of urgencies. However, there is no structured methodology that combines these two fields. Therefore, this study provides a unique added value on combining and harmonizing stakeholder management methodologies to deal with different sense of urgencies and software development techniques that have capability to adapt changing requirements. However, most of the techniques suggested for stakeholder management are designed for a political environment. Therefore, it was questionable whether these techniques could be applicable in a software development environment, where an end delivery (software product) is delivered with limited time, budget and resources. As the literature does not provide an example how to adopt methodologies suggested for managing different sense of urgencies on a software development environment, researcher had to use her own judgment which of the proposed techniques for stakeholder management could be implemented on a software development environment. She documented the methodologies she chooses in the literature survey section.

Second part of the research is the case study. She took part in a software development project, where there is unclear requirement definition and stakeholders with different sense of urgencies, as a researcher. She observed the used techniques and methodologies both from the stakeholder management perspective and software development perspective. She compared them with her findings from the literature and found out to what extend they are applied in the case study or were there new techniques that she did not discover in the literature but succeeded in the case study on managing different sense of urgencies and changing requirements. She figured out that most of the stakeholder management approaches she found in the literature are applied in the case study and worked well. Only priming strategy is not applied in the case study but noted as a successful methodology on changing the sense of urgencies of reluctant stakeholders. On the other hand, two-poles approach is not implemented in the literature but succeeded from a stakeholder management perspective in the case study.

Additionally, experiencing stakeholder management methodologies she found in the literature succeeded in the case study showed that the methodologies applied in a political environment could be adapted on a software development project environment.

As used methodologies in the literature are verified just in one case in this research, it was not sufficient to conclude that the proposed guideline is scientifically valuable. The reason is that, the methodologies that worked in one case do not guarantee that they will work in another project. The reason of the success might be luck, so there the researcher had to find another way to find validate her proposal guideline. Due to the time and resource limitations, applying

the methodologies in different cases and testing their validity was not an option for this research. Therefore, the proposed guideline is verified with an expert opinion.

The guideline proposes a technique/methodology, and then explains the importance of it, in which stage of the project and by which stakeholder it should be used. Researcher interviewed two experts to validate the methodologies in the guideline. They both agreed that the chosen methodologies well addresses the highlighted characteristic of a software development project mentioned in the research question “undefined requirement definition and stakeholders with different sense of urgencies”. They confirmed the guideline is implementable in a software development environment and would be effective to deal with changing requirements and different sense of urgencies in a complex stakeholder environment. One of the experts also confirmed that he has been using most of the proposed techniques in several projects and they are effective tools on delivering negotiation and design sensible products (software). By this validation, the proposed guideline became scientifically valuable.

7.4 Further Study

As a further study, the proposed guideline can be implemented in a real case or a serious game can be designed to teach the application of the proposed methodology to the software development managers in a project management environment. In that way, they can experience applying stakeholder management techniques together with software development techniques without any failure consequence.

7.5 Suggestion to Accenture

Accenture is a big global country that manages many large-scale global projects. When this is the case, there is usually complex stakeholder environment with conflicting interests and different sense of urgencies. My suggestion to Accenture from this research is that project managers within Accenture that are managing software development projects in a complex stakeholder environment should use the guideline in this research as a handbook. Although Accenture uses many of the stakeholder management techniques in projects intuitively, they don't have a structured methodology, which sometimes causes instinctive decisions against faced stakeholder complexities or they learn by experiencing in the projects. This sometimes causes inefficiency in negotiations. This guideline will be a handbook to many inexperienced consultants within Accenture with full of valid insights. This can steep their learning curve and also they can try the methods that are not implemented by Accenture but used in literature and given in the guideline.

8 Bibliography

- 1 About Accenture. (2014). Retrieved December 9, 2014, from <http://www.accenture.com/us-en/company/Pages/index.aspx>
- 2 Abrahamsson, P. (2002). *Agile Software Development Methods: Review and Analysis*. VTT publications.
- 3 Accenture. (2014). *Kick-off workshop*.
- 4 Accenture. (2014). *Strategy Operations & Functions - June 2014*.
- 5 Adolph, S., Hall, W., & Kruchten, P. (2011). Using grounded theory to study the experience of software development. *Empirical Software Engineering* , 16 (4), 487-513.
- 6 Archibald, R. D. (2003). *Managing high-technology programs and projects*. New York: John Wiley & Sons.
- 7 Baker, B. N., Murphy, D. C., & Fisher, D. (2008). "Factors Affecting Project Success" *Project Management Handbook* (Second Edition ed.).
- 8 Baskerville, R. L. (1999). Investigating Information Systems with Action Research by Richard Baskerville. *Communications of the Association for Information Systems* , 2 (19).
- 9 Begel, A., & Nagappan, N. (2007). Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study. *First International Symposium on Empirical Software Engineering and Measurement* (pp. 255-264). IEEE.
- 10 Belzer, K. (2001). Project management: still more art than science. <http://www.pmforum.org/library/papers/2001/Artthan-Science.pdf> , pp. 1-6.
- 11 Bosch, F., & Postma, S. (1995). *Strategic Stakeholder Management: A Description of the Decision Making Process of a Mega Investment Project Europe's Biggest Oil Refinery*. Rotterdam School of Management, Rotterdam.
- 12 Bourne, L., & Walker, D. H. (2004). Advancing project management in learning organizations. *The Learning Organization* , 11 (3), 226-243.
- 13 Bryson, J. M. (2007). What to do when stakeholders matters. In *Public Management Review* (pp. 21-53).
- 14 Bucero, A. (2004). Smart Emotions. *PM Network* , 18 (11), 22.
- 15 Company Profile. (2014). Retrieved December 9, 2014, from Philips: <http://www.philips.co.uk/about/company/index.page>
- 16 Cowie, G. (2003). The importance of people skills for project managers. *Indust Commer Train* , 35 (6), 256-258.
- 17 Crosby, B. L. (1991). Stakeholder Analysis: A Vital Tool for Strategic Managers. In *Technical Notes*.
- 18 de Bruijn, H., & ten Heuvelhof, E. (2008). *Management in Networks On multi-actor decision making*. Routledge.
- 19 De Bruijn, H., Ten Heuvelhof, E., & In't Veld, R. (2010). *Process Management: Why Project Management Fails in COMplex Decision Making Processes* (2nd Edition ed.). Springer.

- 20 DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Making sense of qualitative research* , 314-321.
- 21 Dingsøyr, T. D. (2010). *Agile Software Development: Current Research and Future Directions*.
- 22 Dingsøyr, T., Dybå, T., & Moe, N. B. (2010). *Agile software development: current research and future directions*. Springer Science & Business Media.
- 23 Drurya, M., Conboy, K., & Power, K. (2012). Obstacles to decision making in Agile software development teams. *Journal of Systems and Software* , 85 (6), 1239-1254.
- 24 Feagin, J., Orum, A., & Sjöberg, G. (1991). *A case for case study*. University of North Carolina Press.
- 25 Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development* , 9 (8), 28-35.
- 26 Fumarola, M. (2015, June 26). Software Development Consultant. (S. Sengur, Interviewer)
- 27 Greer, D., & Ruhe, G. (2004). Software release planning: an evolutionary and iterative approach. *Elsevier* , 243-253.
- 28 Guide, A. (2001). *Project Management Body of Knowledge (PMBOK® GUIDE)* (4th Edition ed.). Newtown Square, Pennsylvania, USA: Project Management Institute, Inc.
- 29 Highsmith, J. A. (2002). *Agile software development ecosystems* (Vol. 13). Addison-Wesley Professional.
- 30 Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004). Software Quality and Agile Methods. *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04)* .
- 31 Kezner, H. R. (2013). *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons.
- 32 Kliem, R., & Ludin, I. (1997). The people side of project management.
- 33 Kloppenborg, T. J. (1999). From my experience: Leadership in project life cycle and team character development. *Project Management Journal* (30), 8-13.
- 34 Kumar, S., & Phrommathed, P. (2005). *Reserach Methodology*. Springer.
- 35 Larman, C. (2004). *Agile & Iterative Development: A Manager's Guide*. Boston: Pearson Education, Inc.
- 36 Lau, R. Y. (2005). Towards Genetically Optimised Multi-Agent Multi-Issue Negotiations. *Proceedings of the 38th Hawaii International Conference on System Sciences* . IEEE.
- 37 Loo, R. (2003). A multi-level causal model for best practices in project management benchmarking. *Benchmarking: An International Journal* , 10 (1), 29-36.
- 38 Luo , X., Jennings, N. R., Shadbolt, N., Leung, H.-f., & Lee, J. H.-m. (2003). A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence* , 148 (1-2), 53-102.
- 39 Madeleine, A. (2004). *Madam Secretary: A Memoir*. London: Pan Books.

- 40 *Manifesto for Agile Software Development*. (2001). Retrieved 01 15, 2015, from <http://agilemanifesto.org>
- 41 Mantel, J.R., S. J., & Meredith, J. R. (2011). *Core concepts: project management in practice* (8th Edition ed.). New York.
- 42 Mathew, M. (n.d.). *Doing Business In India: A Cultural Perspective*. Retrieved May 2015, from www.stylusinc.com/business/india/cultural_tips.htm
- 43 Meghann, D., Conboy, K., & Power, K. (2012). Obstacles to decision making in Agile software development teams. *Journal of Systems and Software* , 85 (6), 1239-1254.
- 44 Murray-Webster, R., & Simon, P. (2006, November). *Making Sense of Stakeholder Mapping*. Retrieved December 18, 2014, from PM World Today: <http://www.pmforum.org>
- 45 Newcombe, R. (2010). From client to project stakeholders: a stakeholder mapping approach. In R. Newcombe, *Construction Management and Economics* (pp. 841-848).
- 46 Nicholas, J. M., & Steyn, H. (2012). *Project Management for Engineering, Business and Technology* (Forth Edition ed.). New York: Routledge.
- 47 Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). Distributed Agile Development: Using Scrum in a Large Project. *2008 IEEE International Conference on Global Software Engineering* (pp. 87-95). IEEE.
- 48 Pearlson, K., & Saunders, C. (2004). *Managing and Using Information Systems: A Strategic Approach*. New York: John Wiley & Sons, Inc.
- 49 Pearlson, K., & Saunders, C. (2006). *Managing and Using Information Systems: A Strategic Approach*. John Wiley & Sons, Inc.
- 50 Robert M. Coehoorn, R. M., & Jennings, N. R. (2004). Learning an Opponent's Preferences to Make Effective Multi-Issue Negotiation Trade-Offs. (pp. 59-68). ICEC.
- 51 Rowley, J., & Slack, F. (2004). Conducting a literature review. *Management Research News* , 27 (6), 31-39.
- 52 Ruuska, I., & Vartiainen, I. (2003). Critical project competences – a case study. *Journal of Workplace Learning* , 15 (7/8), 307-312.
- 53 Schwaber, K. (1997). Scrum Development Process. *Business Object Design and Implementation* , 117-134.
- 54 Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*.
- 55 Senapathi, M., & Srinivasan, A. (2012). Understanding post-adoptive agile usage: An exploratory cross-case analysis. *Journal of Systems and Software* , 85 (6), 1255-1268.
- 56 Sengur, S. (2015, June). Case Study on a software development project. (S. Sengur, Interviewer)
- 57 Snavely, W. B. (1981). The impact of social style upon person perception in primary relationships. *Communication Quarterly* , 29 (2), 132-143.
- 58 Sommerville, I. (2000). *Software Engineering*. Harlow, England: New York: Addison-Wesley.
- 59 Sommerville, I. (2010). *Software Engineering* (9th edition ed.). Addison-Wesley.
- 60 Stake, R. (1995). *The art of case research*. Newbury Park, CA: Sage Publications.

- 61 Strang, K. D. (2003). Achieving organizational learning across projects. *PMI North America Global Congress*. Baltimore, USA.
- 62 Strode, D., Huff, S. L., Hope, B., & Link, S. (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software* , 85 (6), 1222-1238.
- 63 Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). Distributed Scrum: Agile Project Management with Outsourced Development Teams. *Proceedings of the 40th Hawaii International Conference on System Sciences*. IEEE.
- 64 Takeuchi, H., & Nonaka, I. (1986). The New Product development Game. *Harward Business Review*.
- 65 Tarhan, A., & Yilmaz, S. G. (2014). Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process. *Information and Software Technology* , 477-494.
- 66 Tellis, W. M. (1997). Application of a Case Study Methodology. *Thw Qualitative Report* , 3 (3), 1-19.
- 67 Thamhain, H. (2004). Linkages of project environment to performance: lessons for team leadership. *Int J Proj Manage* , 22 (7), 533-44.
- 68 The Standish Group. (2009). *The Standish Group*. Retrieved from <http://www.standishgroup.com>
- 69 Verbraeck, A. (2015, May). Action Research. (S. Sengur, Interviewer)
- 70 Vloo, W. D. (2014, August 22). How the software product developed. (S. Sengur, Interviewer)
- 71 Vloo, W. D. (2014, May). Management of a large-scale and global software development project. (S. Sengur, Interviewer)
- 72 Wenzler, I. (2014, August). Evaluation of the success of the software development project. (S. Sengur, Interviewer)
- 73 Wenzler, I. (2014, August 15). How the project managed? (S. Sengur, Interviewer)
- 74 Wenzler, I. (2015, July 31). Project Manager. (S. Sengur, Interviewer)
- 75 Wenzler, I. (2015, June). Success of the two-poles approach. (S. Sengur, Interviewer)
- 76 Yin, R. (1984). *Case study research: Design and methods* (1st ed. ed.). Beverly Hills, CA: Sage Publishing.
- 77 Yin, R. (1989). *Case study research: Design and methods* (Rev. ed. ed.). Newbury Park, CA: Sage Publishing.