MARIA CAROLINA MELLO PASSOS

UNDERSTANDING THE BELIEF SYSTEMS BEHIND SOFTWARE ENGINEERING PRACTICES: STUDIES ON EVIDENCE-BASED PRACTICES IN AN INDUSTRIAL SETTING

Tese apresentada ao Programa Multiinstitucional de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, Universidade Salvador e Universidade Estadual de Feira de Santana, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Manoel Gomes de Mendonça Neto
Co-Orientadora: Prof. Dr. Daniela Soares Cruzes

Salvador

2014

Passos, Carol.

Understanding the belief systems behind software engineering practice: Studies on evidence-based practices in an industrial setting / Carol Passos – 2014.

176p. : il.

Inclui apêndices.
Orientador: Prof. Dr. Manoel Gomes de Mendonça Neto.
Co-Orientadora: Prof. Dr. Daniela Soares Cruzes.
Tese (doutorado) – Universidade Federal da Bahia, Instituto de Matemática, Universidade Salvador, Universidade Estadual de Feira de Santana, 2014.

1. Engenharia de Software Experimental. 2. Prática de Software. 3. Sistema de Crenças.

I. Mendonça Neto, Manoel Gomes de. II. Cruzes, Daniela S. III. Universidade Federal da Bahia, Instituto de Matemática. IV. Universidade Salvador. V. Universidade Estadual de Feira de Santana. VI. Título.

CDD – 005.16
CDU – 004.41

MARIA CAROLINA MELLO PASSOS

UNDERSTANDING THE BELIEF SYSTEMS BEHIND SOFTWARE ENGINEERING PRACTICE: STUDIES ON EVIDENCE-BASED PRACTICES IN AN INDUSTRIAL SETTING

Esta tese foi julgada adequada à obtenção do título de Doutor em Ciência da Computação e aprovada em sua forma final pelo Programa Multiinstitucional de Pós-Graduação em Ciência da Computação da UFBA-UNIFACS-UEFS.

Salvador, 14 de Fevereiro de 2014.

_____

Prof. Manoel Gomes de Mendonça Neto (orientador), Ph.D.
Universidade Federal da Bahia – UFBA

_____

Prof[a].Christina von Flach Garcia Chavez, D.Sc.
Universidade Federal da Bahia – UFBA

_____

Prof. Roberto Almeida Bittencourt, D.Sc.
Universidade Estadual de Feira de Santana – UEFS

_____

Prof. Guilherme Horta Travassos, D.Sc.
Universidade Federal do Rio de Janeiro – UFRJ

_____

Prof. Fábio Queda Bueno da Silva, Ph.D.
Universidade Federal de Pernambuco – UFPE

RESUMO

Diversas teorias na área da saúde e administração de empresas buscam respostas para a questão fundamental de por que as pessoas se comportam da maneira como fazem. Estas teorias têm como objetivo compreender as crenças subjacentes a uma intenção de comportamento e também são usadas para caracterizar como as pessoas progridem da intenção para a prática em ambientes corporativos.

Nesta tese, nós focamos nossa atenção no entendimento do sistema de crenças que há por trás da prática de Engenharia de Software. Nosso trabalho tem o objetivo de caracterizar sistemas de crenças, aplicando teorias de comportamento em times de projetos de software em termos de fatores de influência, tais como crenças, atitude, cultura organizacional e os seus valores, as normas subjetivas, a confiança e autonomia da equipe, os quais realmente impactam as práticas de software em ambientes organizacionais.

Nossa pesquisa passou por dois ciclos, compreendendo três anos de estudo em empresas de software Brasileiras. Um longo estudo de caso etnográfico foi conduzido, aplicando observação-participante, entrevistas e análise de documentos. Um conjunto de entrevistas sobre a origem e impactos das crenças foi realizado com profissionais de diferentes projetos e empresas. Arcabouços conceituais foram construídos com base nos modelos das teorias comportamentais para focar e limitar a coleta de dados e guiar a síntese dos casos de acordo com as questões de pesquisa colocadas.

Os resultados demonstraram uma forte influência das experiências passadas e contextos organizacionais nas práticas de desenvolvimento de software.A partir das descobertas da pesquisa ficou claro que apenas crenças não conduzem o time à ação e ao comportamento. Fatores como atitude direcionada a um comportamento têm bastante influência na prática. Novas informações obtidas sobre algo contribuem para formação de opinião ou predisposição para agir e têm o potencial de afetar a atitude a depender da força das crenças relacionadas, o que leva a intenção de comportamento.

Outro ponto importante é o alinhamento das crenças do time de projeto. Crenças comuns ao time são refletidas em práticas realmente aplicadas, contudo existem crenças sem atitude que não resultam em ação, assim como, conflitos no time que claramente dificultam a

adoção de novas práticas. Todas estas conclusões motivaram a busca de teorias comportamentais que melhor explicassem e conceituassem o comportamento humano.

O estudo mostrou que é possível caracterizar sistemas de crença em contextos de projeto de software dentro de uma perspectiva comportamental. Nós fomos capazes de prover uma rica narrativa para a pesquisa de Engenharia de Software e nossa abordagem gerou recomendações práticas e úteis para empresas.

A principal contribuição desta tese é aprofundar conhecimentos e experiências relevantes sobre a caracterização das crenças em contextos organizacionais e em como estas crenças, e demais fatores de influência, realmente impactam práticas, processos e decisões nos projetos da indústria de software.

Palavras chave: Sistema de Crenças; Cultura e Valores Organizacionais; Intenção de Comportamento; Ação Racional; Práticas de Times de Projeto; Prática de Desenvolvimento de Software; Estudo de Caso na Indústria.

ABSTRACT

Many theories in health care and business administration seek answers to the fundamental question of why people behave the way they do. They aim to understand the beliefs underlying an intention or behavior. These theories are currently used to find out how people progress from intention to practice in business environments.

In this dissertation, we focus our attention on understanding belief systems behind software engineering practice. Our work aims to characterize a belief system applying behavioral theories in software project teams in terms of the influence factors, such as beliefs, attitude, organizational culture and values, subjective norms, team confidence and autonomy, that actually impact on software practices in industrial settings.

Our research went through two cycles, comprising three years of study in Brazilian software companies. A long-term ethnographic case study was conducted, employing participant observation, interviews, and document analysis. A set of interviews on origins and impacts of beliefs was performed with professionals from different project teams and companies. Conceptual frameworks were built based on behavioral theory models to focus and bound the collection of data and guide the synthesis of the results on the research questions posed.

The results showed the strong influence of past experiences and organizational contexts on the software development practices of project teams. Based on the findings of the research, it became clear that beliefs alone do not lead project teams to action and behavior. Factors such as attitude toward behavior have a significant influence on practice. New information about something contribute to shape an opinion or predisposition to act and have the potential to affect the attitude depending on the strength of related beliefs, which leads to behavior intention.

Another important issue is how consonant are the beliefs of a project team. Common strong beliefs are reflected into practices that project teams actually adopt. However there are beliefs without attitude that do not result in action, as well as team conflicts that hinder the adoption of new practices. All these findings motivated a search for behavioral theories that could explain and conceptualize human behavior.

The study showed that it is possible to characterize belief systems in software project contexts within a behavioral perspective. We were able to provide rich narrative accounts for software engineering research and our approach has led to practical and useful recommendations for companies.

The main contribution of this dissertation is to deepen relevant knowledge and experience on the characterization of beliefs in organizational contexts and how they and other influence factors actually impact practices, processes and decisions in software industry projects.

Keywords: Belief System; Organizational Culture and Values; Behavior Intention; Reasoned Action; Project Team Practices; Software Development Practice; Industrial Case Study.

A minha família e meus amigos e orientadores pelo apoio, direção e confiança.

ACKNOWLEDGEMENTS

At the end of this journey, I wish to thank all of those who, somehow, have made the implementation of this work possible.

I thank my parents, ALCEU and ARLINDA, who provided opportunities for my professional and personal development; JOSÉ MARIO, my husband, for the great encouragement during my search for knowledge; and my sister-friends CAROL and ILA for being by my side.

I would like to thank my two advisors, Professor MANOEL MENDONÇA and Professor DANIELA CRUZES, for their support, enthusiasm, and constant encouragement during these 5 years of study. Thanks to my classmates, RENATO, THIAGO, MÁRIO and AMANCIO, for the exchange of experiences and knowledge. Also, I share with my English Teacher, TOM, the merits achieved with this dissertation written in English.

Special thanks to my two favorite bosses, LIKISO e ANA PAULA, for all endorsement they gave me during this journey and I would like to thank the SOLUTIS organization for supporting this work. I return the affection of my co-workers, SUELEN, KARINA, ALEX, BRUNO, ISMAEL and ELISEU, for their devotion and tolerance to dedicate their precious time to support my research. And I am grateful to all involved in this study, specially the interviewees for their insights and cooperation.

I thank the CAPES Foundation who granted me a scholarship during the course of this PhD and NTNU for welcoming me so well. Also, thanks to my Norwegian friends who gave me support, knowledge and self-confidence to begin a new career as a Computer Science researcher.

Success is not final, failure is not fatal: it is the courage to continue that counts.

Winston Churchill

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF TERMS

Conceptual Framework	It explains the main aspects to be studied, their key factors, variables and relationships among them in order to arrive at a balanced and comprehensive understanding of the subject.
Within-Case Analysis	It is a method to draw and verify descriptive conclusions about the phenomena in a bounded context that make up a single case.
Cross-Study Synthesis	Research synthesis is a collective term for a family of methods for summarizing, integrating, combining, and comparing the findings across different studies on a topic or research question.
Cross-Case Analysis	It is an appropriate method to synthesize aspects of an object of study by seeing see processes and practices across multiple cases to understand how they are qualified by local conditions and, thus, to develop deeper descriptions and powerful explanations.
Behavioral Theory	A theory that explains and predicts individuals' actions and practices. It seeks to predict behavior and understand its relationship with beliefs, attitude and social norms.
Subjective Norms	The expectations of other people in a social environment, their influence on a specific behavior, and also how they will view the behavior.
Self-Efficacy	is concerned with judgments of how well one can execute courses of action required to deal with prospective situations.

LIST OF ACRONYMS

OC	Organizational Culture
TRA	Theory of Reasoned Action
TPB	Theory of Planned Behavior
TAM	Technology Acceptance Model
SE	Software Engineering
IS	Information Systems
HBM	Health Belief Model
CVM	Competing Values Model

Chapter

1

This chapter presents the motivation of this work, including key aspects from literature, the research goals and approach, and describes the research problem, highlighting the main results. Finally, the outline of the dissertation is presented.

1 INTRODUCTION

A project team belief system is one of the foundations of the set of Software Engineering (SE) practices adopted. Most of the practitioners rely on beliefs to make their technology decisions. In spite of not having a formal trial basis, a belief system is inevitably taken on board when a particular method or tool is adopted (Wernick and Hall, 2004). Important decisions are made through an interactive process involving people who influence each other. So, the beliefs and values of the project team can guide interactions between its members, and especially actions oriented toward certain software practices.

Beliefs can be defined as conceptions, personal ideologies, and perceptions of the world that shape practice and orient knowledge. The concept of belief implies the existence of a mental state with intentionality, interacting with goals and influencing ordinary actions (Aguirre and Speer, 2000). These beliefs are built over a set of interactions, relationships, processes and activities of the group. Beliefs exist in the form of expectancy-rules and these rules are tested for a given situation. In this context, actions are driven by what is believed, by what is culturally assumed to be true about the world (Funda Savasci-Acikalin, 2009).

Folklore consists of legends, music, oral history, proverbs, jokes, fairy tales and customs that are the traditions of some culture, a group of people or community. The notion of folklore has many cultural aspects. In addition, folklore can also serve to validate a culture, as well as to transmit a culture's morals and values (Georges and Jones, 1995). Folklore and

beliefs are dependent on a surrounding context (culture, region, community or organization) and, at first, they do not have formal trial basis.

Research has documented that practitioners' beliefs and technical folklore related to work processes in organizational context have a significant impact on their practices and that this influence on practices plays out in interesting ways (Aguirre and Speer, 2000), (Funda Savasci-Acikalin, 2009), (Douglas and Wykowski, 2010). For example, evidence shows that there is a strong connection between a belief system and an organizational culture (Dubé and Robey, 1999), (Wernick and Hall, 2004), (Alavi, Katworth and Leidner, 2006), (Tolfo and Wazlawick, 2008), (Iivari and Iivari, 2011). Unfortunately, these studies did not directly address or characterize belief systems in a software project context. This dissertation focuses on this issue.

There is a common understanding that beliefs and goals are related, but this connection has not been completely established empirically. In order to determine what goals an individual is likely to have at any given time, it is necessary to have an available set of beliefs about the individual's behavioral intention in a given circumstance. In the SE area, this lack of understanding may be due, in part, to an incapability to investigate, understand and document the nature and effect of the belief systems underlying current SE theory and practice. Also, even today there are studies that focus their analysis on the object (technology or practice) and do not consider the subject (individual, team, organization).

Many theories in health education and business administration seek answers to the fundamental question of why people behave a certain way. Behavioral theories, like the Theory of Reasoned Action (TRA) (Fishbein and Ajzen, 1975),have received considerable attention within fields related to social behavior. In a subsequent work, one of the authors has extended the Theory of Reasoned Action to the Theory of Planned Behavior by including a measure of perceived behavioral control. The Theory of Planned Behavior (TPB) (Ajzen, 1985) is today one of the most popular socio-psychological models for the prediction of behavior. The TPB is a generalization of TRA that is indicated to predict behavior which is not entirely under an individual's volitional control. It fits well in the context of SE research within software industry and provides a theoretical framework in mapping software project team behavior.

According to these theories, people's behavioral intention is based on their posture and how they believe others would think about their behavior. They are derived from the study of attitude and behavior in the social psychology setting and provide a suitable framework for conceptualizing human behavior. These theories are very well established in some areas, like public health, with assessment studies used to determine behavioral intention of participants (Montano and Taplin, 1991) , (Vanlandingham et al., 1995), (Davis and Venkatesh, 1996), (Ghorab, 1997), (Sutton, McVey and Glanz, 1999), (Venkatesh and Davis, 2000), (Randolph et al., 2009), (Roberto et al., 2011), (Peslak, Ceccucci and Sendall, 2011), (Blank and Hennessy, 2012).

In particular, behavioral theories can be helpful in generating rich and detailed accounts of software project teams, the interactions between their members, and, especially, the actions oriented toward certain software practices. The TRA and TPB models appear to provide a good framework to understand the influence of a belief system on team practices. Behavioral theories fit into the context of our research, because it allows us to study the way beliefs and attitudes are formed and their relationship to behavior and practice with room to explore other relevant aspects. However, the application of a theoretical approach to SE can be challenging. The problems and objects of study in SE require approaches suited to their dynamics and contexts.

1.1 MOTIVATION

Research findings can influence decisions at many levels, but only if one knows how to translate them into actionable and convincing information. Research evidence can provide a reference against which decisions and choices may be evaluated. It is possible to provide the mechanisms needed to assist practitioners to adopt appropriate technologies and practices in their contexts (Kitchenham ,Dybå and Jørgensen, 2005). In order to achieve a more integrated approach to adopting research findings, both practitioners and researchers

have to develop coordinated mechanisms to support the continuing evolution of SE knowledge (Dybå, Kitchenham and Jørgensen, 2005).

Unfortunately, evidence-based paradigm is not wide spread in the software industry. Practitioners deal regularly with technology decisions, and do not have (or cannot access) trustworthy empirical evidence to support them. Busy professionals need to access relevant and reliable knowledge easily. They also need to trust and believe in the information. However, most software development organizations have limited time and money for determining which of the available technologies will be most beneficial to their specific contexts.

Nonetheless, decisions have to be made. Issues such as efficiency and effectiveness constantly guide software engineers. They have to adapt to new technologies, situations or contexts. Frequently, they are under pressure to adopt immature technologies because of market and management pressures (Dybå, Kitchenham and Jørgensen, 2005). Most of the practitioners rely on their beliefs or the community's folklore to make their decisions. In spite of not having a formal trial basis, these beliefs are built over a sophisticated set of relationships and communication channels.

The introduction of a particular technology (practice, method, tool or technique) to common practice in a SE project requires a good understanding of the context, generation and analysis of evidence, packaging and support, and careful deliberation of the perspective user for the new practice (Pfleeger, 1999). Understanding the belief systems behind software team practice is not a trivial task because it involves more than just building theoretical foundations of the subject. Thus, the definitive step towards the introduction of a new practice into an organizational culture is to convince people of the usefulness of the practice to them. This requires a deep grasp of the organization belief system. How do people come to believe that something is useful to them and reach the decision to use it in their organization or particular project? A theoretical basis for this kind of process obtained by empirical research needs to be evolved through a collaborative and reflexive research approach to investigate the origin, sources and impact of beliefs on SE practices. This dissertation focuses on this issue and, to our knowledge, it is the first software engineering dissertation particularly focusing on this subject.

1.2 RESEARCH GOALS

This research aims at characterizing belief systems of software project teams. The main goal of this project is to deepen relevant knowledge and experience of the characterization of team beliefs in organizational contexts and how they actually impact practices, processes and decisions in software industry projects. In order to reach this goal, the study addresses the influence factors associated to project team beliefs and its attitude toward behavior, the organizational culture and subjective norms, and the resulting sense of self-efficacy of the team to predict behavior intention and also document the inconsistencies between declared beliefs and real practice in industry software projects. We expect to produce innovative results by examining the significant relationships between project team beliefs and software engineering practice and behavior using a qualitative approach in order to motivate the improvement of software development processes in the software industry.

Thus, the goals of this research are expressed by the following Research Questions (RQ):

1. **RQ1**: What are the main factors that influence software development team behavior intention and practice?
2. **RQ2**: How do these factors influence the adoption of new practices in software teams?
3. **RQ3**: How do common and conflicting beliefs and cultural values impact positively or negatively on software team behavior and practice?

In order to answer question RQ1 it was necessary to first understand and find ways to identify and map the organizational and team level factors associated to beliefs related to software practices, their origin (when), sources (where) and contexts (under what circumstances) and how they influence project team attitude toward development practices. Upon obtaining qualitative evidence that explained how those factors impact software industry project decisions and highlighted a significant connection between team belief systems and organizational culture, it was possible to answer RQ2. Regarding RQ3, it was necessary to identify salient beliefs and uncover the beliefs which could hinder or benefit the adoption of new software practices.

1.3 RESEARCH APPROACH

As illustrated in Figure 1, we ran a long-term study, involving software development projects, which went through a first cycle of 18 months. The research also included another cycle of 12 months with the aim of characterizing project team belief systems of software companies in Brazil. During the first cycle, we applied an ethnographic approach, immersing ourselves in the day-to-day project activities. In the second cycle, a series of interviews was performed and we conducted focus group meetings with professionals involved in software projects of four companies. The final stage of the research is also presented in this document. It involves the synthesis of the results of the second cycle through a cross-case analysis that compared and clustered the influence factors on software practices originating from the case studies conducted.

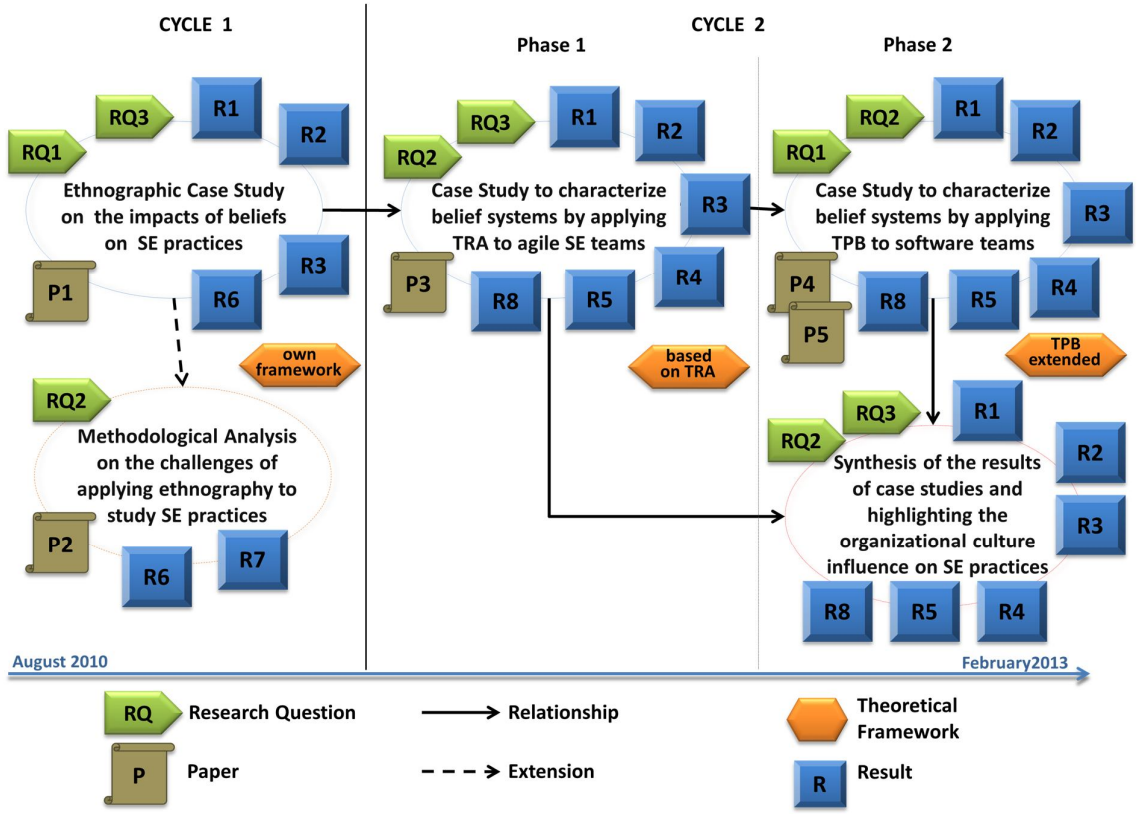

Figure 1 Research Overview

The first cycle of the study, described in Passos et al. (2011) and shown as P1 in Figure 1, adopted an ethnographic approach to investigate the impact of beliefs on SE practices. It aimed at understanding and gaining knowledge on beliefs and the phenomena related to team practices in the software industry. It also performed a methodological analysis about the benefits of using a collaborative ethnographic approach in software organizations (Passos et al., 2012, shown as P2 in Figure 1). We did this by introducing ourselves into real world software projects to delve deeper into the complexity of team belief systems and uncover meanings of behaviors, actions and events. We performed interviews, observations and document analysis, using data from different sources to generate a comprehensive understanding of social interaction and its subtleties in the context of a software organization. The study showed that team beliefs emerge and evolve from past experiences reapplied in past or current projects.

From the results, it became clear that beliefs alone do not lead project teams to action and behavior. Factors such as attitude toward behavior have a significant influence on practice. Also, we noticed a significant connection between belief systems and organizational culture. Lastly, we also observed that there were some common and conflicting beliefs between the team members, which fostered or hindered the adoption of new practices. All these findings led to an evolution of our initial conceptual framework into an underlying behavioral theory, a more suitable instrument for conceptualizing human behavior.

For the second cycle, we conducted two new case studies applying behavioral theories, derived from insights gained in the first cycle of the research. In the first study of the second cycle (Passos, Cruzes and Mendonça, 2013a, shown as P3 in Figure 1), we applied TRA to agile software project teams of one company, focusing on the influence factors that actually impact on software practices in industrial settings. In a 12-month period, we conducted focus group meetings and a set of interviews with professionals from different profiles and involved in the company's agile projects. This study confirmed that beliefs arise from two main sources: past and current project experiences. These past experiences emerged from a personal hands-on approach, i.e., what did or did not work well in previous projects. However, they were taken into account without much regard for the present or original contexts. The results also showed that organizational support and culture were very important to achieve self-management team effectiveness.

The second case study of the second cycle was conducted in three different software companies to characterize a belief system applying TPB to software project teams (Passos, Cruzes and Mendonça, 2013b, shown as P4 in Figure 1) (Passos et al., 2013c as P5 in Figure 1). It involved a set of interviews and information gathering about the contexts of the projects for six months. The findings also confirmed the influence of a team belief system on software practices and a significant connection between organizational culture and subjective norms and project team behavior intention.

Afterwards, we extended and refined the work by synthesizing the results of the case studies of the second cycle and highlighted the organizational culture influence on the observed practices. In this extension, we worked with four software companies and three data sources: context mapping, focus group meeting and interview notes. We achieved an in-depth understanding of the industrial setting through a more comprehensive data analysis of all these sources. After adding new settings to the research, we could apply the knowledge acquired to real-life situations of the new scenarios combined with the ethnographic scenario of the first cycle of the research. We also generated rich and detailed social accounts of software project teams, involving the influence factors associated to team beliefs, its attitude toward behavior, the organizational culture and subjective norms.

It is important to mention that we started the research with our own conceptual framework developed on the basis of our perceptions about the main aspects to be studied – the key factors or variables – and the presumed relationship among them. In particular, we focused on the social interactions, communications, and relationships that arise as an intrinsic part of adopting new software development practices to evolve the initial conceptual framework into an underlying theory.

After approximately one year and a half of ethnographic research(cycle 1), a set of behavioral theories caught our attention because they help to predict the intention and behavior of people quite well, and provide a simple basis for understanding the relationship between beliefs, attitude and subjective norms. Furthermore, these behavioral theories are well established in some areas, like health care and business administration, with assessment studies used to determine behavioral intention of participants (Montano and Taplin, 1991) , (Vanlandingham et al., 1995), (Davis and Venkatesh, 1996), (Ghorab, 1997), (Sutton, McVey and Glanz, 1999), (Venkatesh and Davis, 2000), (Randolph et al., 2009), (Roberto et al.,

2011), (Peslak, Ceccucci and Sendall, 2011), (Blank and Hennessy, 2012). Thus, two new conceptual frameworks were built, first much like the TRA and TPB models, and later refined into our underlying theory for representing the significant relationships between team belief systems and practices in software engineering organizations. The evolution of this conceptualization of belief systems is described in Chapters 3, 4, 5 and 6 of this dissertation.

1.4 MAIN RESULTS AND CONTRIBUTIONS

This work presents an approach for the representation and mapping of organizational and team level factors associated to beliefs related to software practices, their origin and contexts and how they impact project team attitude toward practices and project decisions. This approach involved the definition of an underlying theory as an instrument for representing team belief systems in SE contexts. The proposed theory was an evolution of our conceptual frameworks after the incorporation of some variables and relationships arising from the study of the TRA and TPB models.

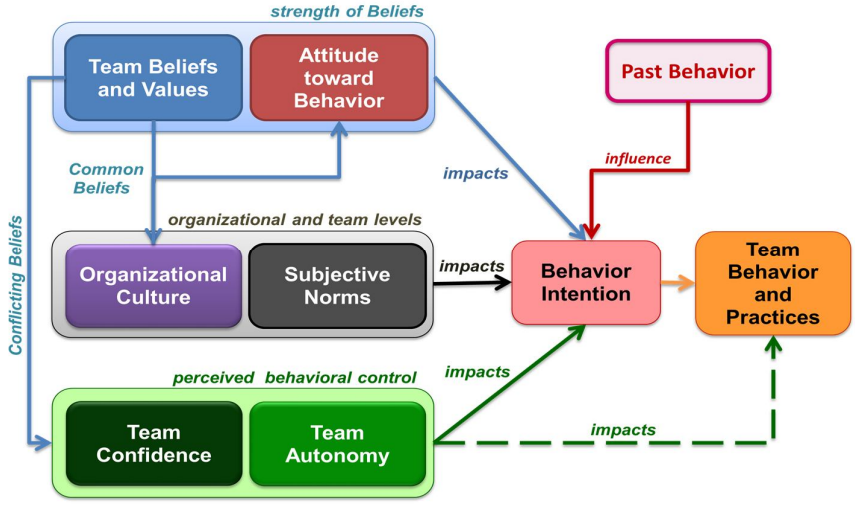

Figure 2 Theoretical Model proposed

Figure 2 shows the resulting theory. In it, the strength of beliefs is represented by *team beliefs and values* combined with *team attitude*, sustained by the information team

members have about a certain behavior and its likely consequences. Any information about a subject or matter that a project team believes in contributes to shape an opinion or predisposition to act and have the potential to affect team attitude, which leads to behavior intention and practice. Another component, represented by *organizational culture* combined with *subjective norms*, also impacts the team's behavior intention and, consequently, the team's practices. Therefore, the behavioral intention is affected by what others think and the strength of their opinion in context of the organization. In addition, the *perceived behavioral control* component denotes people's perception of the degree to which they are capable of, or have control over, performing a given behavior. It increases the likelihood that people will expend effort and persevere in their attempts to perform a practice.

Four other important elements are part of this theory. First, in the studies described in Chapters 3, 4 and 5, *past experience* or *repeated behavior* proved to be a very strong influential factor and predictor of behavior intention. As a behavior became more habitual, a significant relationship between repeated past experiences and intention was evidenced. Secondly, the element of *perceived behavioral control* directly affects behavior besides interacting with behavioral intention to affect behavior. Thirdly, *common strong beliefs and values* of project teams can reinforce the strength of organizational culture and their influence on project team decisions. In addition, a good fit between the organizational culture and the basic assumptions of a project team reinforce the strength of the team attitude toward a new practice. And lastly, the *team belief and values* element, when in conflict, has a direct relationship to the *perceived behavioral control* component, hindering *team confidence and autonomy* and impacting project team practices negatively.

Thus, as evidenced by the proposed theory, the three research questions posed earlier led to new knowledge that can help and support the software industry and bring new insights to academic research. At the end of the research, we achieved the following results:

R1. Identification and mapping of team beliefs origins, sources and contexts;

R2. Characterization of significant relationships between team belief systems and practices in software engineering organizations;

R3. Representation of *common* and *conflicting* beliefs and their impact on practices, processes and decisions in industrial contexts;

R4. Understanding of how organizational culture and norms actually impact software industry practice and the role of repeated behavior in this context;

R5. Guidelines to improve software team practices through the knowledge of team belief system.

As part of this research, a long-term ethnographic case study was conducted. Difficulties and decisions were recorded and compared with those encountered in the literature.

Regarding methodological issues, our main results were:

R6. Relevant insights on how to deal with the key challenges of applying ethnography to study software practices;

R7. Presentation of the benefits of using a collaborative ethnographic approach, including a participatory action research strategy – collaborating with the participating companies is as important as studying them;

R8. Clarification on how to apply behavioral theories to study software engineering practices through the proposal of an underlying theory adapted to the problems and objects of study in the field of Software Engineering.

All these results are illustrated in Figure 1, which outlines the cycles and phases of this research and the three case studies conducted, their focus and research methods, as well as the relation to research questions, papers and results.

1.5 DOCUMENT OUTLINE

The organization of this document has been defined to facilitate its comprehension by the readers. After the introduction chapter, which describes the motivation, goals and scope of the research, the document is organized as follows:

1. CHAPTER 2 – presents the theoretical background and literature review on the topic, describing the foundations for this work and related work under the perspective of the Software Engineering industry.

2. CHAPTER 3 – presents the research approach and design used in the first cycle of the research, that involved a long-term ethnographic case study, and describes the key findings encountered by answering the research questions posed.
3. CHAPTER 4 – describes how we conducted the case study of phase 1 of the second cycle of the research, illustrating the research methodology and process involved, and presenting the main results by answering the research questions.
4. CHAPTER 5 – presents the descriptions of the case study and companies involved in phase 2 of the second cycle, including the research methods used, the challenges overcome and the results found.
5. CHAPTER 6 – discusses the cross-case analysis findings by synthesizing the results of the second cycle, and presents the implications for research and practice, and recommendations for software companies.
6. CHAPTER 7 – presents an overall discussion of the conclusions, explores the major contributions, and inherent limitations and validity threats of our work, and outlines possible future research in the area.

Chapter

2

This chapter elaborates on the background and related work and provides a comprehensive view of the foundations for this work. First, the chapter presents the context of the software practice adoption process and how team beliefs and values are underlying the adoption of new practices. Then, it discusses the main factors that influence the strength of the relationship between intention and behavior. Subsequently, the behavioral theories of reference are presented. Finally, the chapter describes the ethnographic research method under the perspective of our experience in the software engineering industry.

2 BACKGROUND AND RELATED WORK

Since the success or failure of any new practice is a direct result of its adoption rate, it is critical to have absolute confidence in the adoption process before beginning. While certain strategies encourage faster practice introduction, adoption is not something that can be dictated. Adoption occurs when practitioners decide for themselves that the practice provides them with a visible benefit. It happens when their beliefs and values guide attitude oriented toward the practice. They can weigh the practice benefits against any perceived pain, such as giving up the comfort of an old way of doing something.

Practitioners are faced regularly with project decisions and are guided by aspects of efficiency and effectiveness. They need to select and decide to use a set of practices, such as: task estimation, software testing or component reuse, in the context of software industry projects. The mindset of practitioners, when making a decision, can be very much influenced by the technologies and practices in use at that time by the software community. Therefore, it is known that compelling evidence is not enough to ensure the adoption of a new practice. The adoption decision requires a deep understanding of the overall dynamics of activities and

interactions in the organizational and project contexts. In addition, the practice must be packaged and supported so as to make it friendlier and easier to understand and use (Pfleeger, 1999).

Promising practices (Leseure et al., 2004) are a coherent, integrated collection of ideas, values, procedures, techniques and tools. They are promising rather than best, because every practice needs customization before it can perform well. Promising practices have consistently shown results superior to those achieved by other means, and that is used as a benchmark. They are expected to generate higher performance levels everywhere and in any context, because of the process of developing and following a standard way of doing things that multiple organizations can use. Examples of promising practices are the adoption of quality certification systems, like ISO 9000 or CMMI[1] model.

In some cases, the adoption decision is triggered by a need associated with normative pressures applied by customers and the market. So, many software organizations grab new and promising practices without much consideration for empirical investigations of their benefits. This rushed adoption, frequently unsupported by careful empirical evaluation, has been successful in some cases, but disastrous in others. Software practices applied by one team in one project context will not necessarily be the same in another project, since there is a clear influence of the project context on how things happen. Since the object of investigation interacts with the context; it must be implemented in different ways to fit into the specific context. Thus, the project context should be placed at the center of all relevant software project discussions (Petersen and Wohlin, 2009) and has great relevance to characterize how each practice relates or is influenced by beliefs and how these relationships affect software project decisions.

[1] CMMI Institute. http://cmmiinstitute.com/

2.1 BELIEF SYSTEM

Belief is the psychological state in which an individual holds a proposition or premise to be true. The concept of belief implies the existence of an individual and mental state with intentionality, interacting with goals and influencing ordinary actions (Aguire and Speer, 2000). The belief condition requires that someone believes or accepts something. It is also possible to define belief as a state of mind that embodies trust and confidence in something (Douglas and Wykowski, 2010), (Funda Savasci-Acikalin, 2009). Although some beliefs persist over time, others weaken or disappear and new beliefs are formed. Beliefs represent the information we have about the world in which we live; they form the foundation of many of our responses and actions to events and situations of this world (Ajzen and Gilbert Cote, 2008).

Although people can form many different beliefs about something, it is assumed that only *salient* or top-of-mind beliefs operate as determiners since they are readily accessible and activated spontaneously without much effort (Middlestadt, 2012). Salient beliefs are the most frequently mentioned outcomes and commonly exceed a particular frequency. They are those that first come to mind when respondents are asked open-ended questions and may be highly predictive of both intentions and behavior (Sutton et al., 2003).

A belief system is a set of mutually supportive beliefs as a function of several influence factors derived from different levels (organizational, team, individual). It is related to what people remember and perceive, how people attribute causes to events and circumstances, how people feel, and what motivates them. For example, the beliefs identified as being common to Software Developers, may be considered to form a definition of who is a proper software-based systems developer, in contrast to Mechanical Engineers, who will work under a different belief system producing different sorts of results. According to Douglas and Wykowski (2010), belief systems are comprised of two parts: the structure or the set of standards regarding what people believe and the actual content of what people believe. Belief content takes the form of the assumptions one holds about the reason, meaning of events and facts and the behavior of others. The structure of what one believes shapes and sustains the

contents of his beliefs as informed by perception, memory and emotion related to culture and surrounding environment.

There are significant scientific studies in other areas about the impact of beliefs on team or group practices. The majority of existent empirical findings about beliefs, attitude and practices are not suitable to guide the software industry in the adoption of new practices, because they are focused in other areas, such as educational (Ernest, 1989), (Aguire and Speer, 2000), (Bain and McNaught, 2006), (Funda Savasci-Acikalin, 2009), (Mansour, 2009) or health care (Montano and Taplin, 1991), (Vanlandingham et al., 1995) (Sutton, McVey and Glanz, 1999), (Randolph, 2009), (Roberto et al., 2011), (Blank and Hennessy, 2012). As our focus is on the software development context, in our approach the belief system is directly associated with the strength of software team beliefs and attitudes, and the culture and norms on organizational and team levels that impact team behavior intention in the software industry.

Regarding the educational area, Ernest (1989) investigated the impact of beliefs on the teaching of mathematics. He aimed at identifying the key factors that determine the autonomy of the mathematics teacher, and hence the outcome of teaching innovations. From Ernest's point of view, the practice of teaching mathematics depends on a number of factors, most notably: the teacher's mental contents or schemas, particularly the system of beliefs concerning mathematics and its teaching and learning; the social context of the teaching situation; and the teacher's level of thought processes and reflection. Only by considering all these factors is it possible to begin to explain the complex notion of the autonomous mathematics teacher.

Aguire and Speer (2000) explored the relationship between teacher beliefs and goals and presented a detailed analysis of how specific beliefs can influence the formulation of teaching goals that, then, influence the actions of each teacher. For them, beliefs affect practice in complex ways. They described the nature of teacher beliefs and how they manifest themselves in the ongoing development of teaching goals, and also explained that what teachers profess to believe and what they actually do in the classroom may or may not be consistent. Only the beliefs that are the most salient can exert strong influence on practice during the act of teaching. They examined shifts in the teaching goals which provided insight

into which beliefs were playing a particularly influential role in the formulation of the new goals.

Funda Savasci-Acikalin (2009) critically analyzed research studies regarding the relationship between teacher beliefs and practice with other influential factors discussed in the science education literature. His findings indicated that this relationship has a complex nature and should be considered within context because it is context-dependent. The results were consistent with the idea that teaching beliefs have a significant influence on classroom practices. However, other factors such as school community influence, and the pressures to cover the curriculum and preparing students on exams are some of the possible factors that may influence teacher classroom practice as well as teachers' beliefs about teaching and learning, and should be taken into account by researchers. Another issue raised by the authors refers to the research method. In-depth research that combines survey information with a long period of class room observations involving case studies and longitudinal studies with a small number of participants seem to be more valuable in order to understand the complex relationship between teacher beliefs, practice and school context.

Consistent with Funda Savasci-Acikalin's ideas, Mansour (2009) argues that teacher beliefs and practices cannot be examined out of context, but should always be situated in a physical setting in which constraints, opportunities or external influences may derive from sources at various levels, such as the individual classroom, the school, and the community. He presented socio-cultural perspectives to explain the consistency and inconsistency of teacher beliefs and practices. According to him, there is more than one social factor which can affect or shape teacher beliefs. Also, his findings indicated that the relationship between teacher beliefs and their practices are far from straight forward. Beliefs are influenced by the interaction within the nested social contexts in which they are situated. Beliefs can be contradictory, and compete for priority; have indirect but strong effects on teaching practice, and be context-dependent, so they have different levels of strength in different contexts.

The educational research described here has some points in common with our work. First, the results highlighted the complex nature of the relationship between belief and practice and its context-dependent character. Second, they recognized the particularities of a belief system and its determination by influence factors derived from sources at various levels

(organizational, group and individual). Third, they pointed out that beliefs can be contradictory, sometimes people do not actually practice what they preach, and only the most salient beliefs can exert strong influence on practice. Besides, some authors advocated that observations involving case studies and longitudinal studies with a small number of participants are a good strategy to deal with the complex relationship among beliefs, practice and context.

With respect to studies in health care, Montano and Taplin (1991) conducted a prospective study for the prediction and understanding of mammography participation applying the TRA model and Health Belief Model (HBM).The HBM measures of perceived susceptibility and efficacy were used, which showed significant positive correlations with mammography participation. The study didn't explore the whole situation of women's beliefs and attitudes in determining participation, but past behavior, normative beliefs, and emotional state were all found to be significant direct predictors of intention and participation.

Vanlandingham et al. (1995) also applied the HBM and the Theory of Reasoned Action (TRA) to an analysis of unsafe sexual practices. The HBM model accounts for relevant beliefs related to susceptibility to risk, risk severity, possible benefits or barriers, and self-efficacy. The TRA considers the set of salient beliefs concerning the anticipated consequences of the behavior in question.TRA and HBM have much in common and both models predict that salient beliefs about behavioral consequences should be predictive of behavioral practice. For both models, it is important to take into account the socio-cultural perspectives to explain the consistency and inconsistency between belief and real practice. Thus, both models provide useful frameworks for investigating this problem, even in quite distinct contexts. TRA was the better model in the context of this research due, in large part, to the accommodation of normative influence.

In the work of Sutton, McVey and Glanz (1999), the findings from the analysis of behavioral beliefs suggested that beliefs based on personal experience, in contrast to those beliefs acquired through information from outside sources, are likely to be highly resistant to change. Also, normative beliefs with respect to the person who is the main reference were also significantly related to practice. Some years later, Randolph (2009) conducted a study to understand the sexual behavior of seriously mentally ill women. He postulated that a better understanding of the psychological determinants of safer sex attitudes and normative beliefs

would likely help in the association between these constructs and safer sex intentions. In the same line, Roberto et al. (2011) agreed that understanding the factors that underlie attitudes and normative beliefs would provide valuable information for both theoretical and practical reasons. In addition, Blank and Hennessy (2012) reported that underlying beliefs concerning sexual behavior for the severely mentally ill can be elicited and these beliefs co-vary with social characteristics in predictable ways.

Likewise, from the health care research perspective, the relationship between belief and practice is context-dependent and socio-cultural factors are relevant to predict behavior and intention. Also, the identification of salient beliefs was determined as a prerequisite for behavior prediction. As identified in our research, underlying attitudes and normative beliefs, with respect to the person who is the main reference, are highly predictive of both intention and practice. Regarding the origin and sources of beliefs, some authors suggested that beliefs based on personal experience are stronger then beliefs acquired through information from outside sources, even when these past experiences were taken into account without much consideration for their original context.

There are some scientific studies closer to the Software Engineering area involving beliefs and practices. In the Information Systems (IS) research area, Wernick and Hall (2004) investigated the software development-related aspects of IS in relation to a model of science which describes scientific communities and considers how scientists form communities, and how each of these communities is united by a common underlying agreement on principles and practices. They aimed at providing support for the investigation of Software Engineering (SE) practice. For them, the mindset of a developer when making a decision can be considered in the context of the influences arising from the tools and techniques in use at that time. So, the use of different SE tools and techniques, by a group involved in several sets of beliefs, may produce different results when applied in practice. In this scenario, they claimed that practitioners are divided into communities, characterized by the unified thinking of its members, and therefore, the application by analogy of the view of scientific activity to SE practice was justifiable.

Beyond that, Limayem, Hirt and Cheung (2007) examined the role of habit in the context of information system continuance. Their results supported the argument that habit (past behavior performed repetitively) acts as a moderating variable of the relationship

between intentions and IS continuance practice. The data suggested that satisfactory past experiences with a behavior are a key condition for habit development as they increase one's intention to repeat the same course of action. In addition, a relatively stable context is another important prerequisite for habit development. Thus, it was clear that, if an individual or group has taken a decision to pursue a certain practice and this behavior has led to a satisfactory outcome, the next time a similar situation arises they will perform the behavior in question, because they believe that they will achieve success.

These two last studies reported are closely related to the work described in this dissertation. First, the results highlighted the context-dependent nature of the relationship between belief and practice. Second, they recognized some influence factors associated to project team beliefs and practices. Third, they identified that beliefs based on personal and satisfactory experience can exert strong influence on practice. And lastly, they pointed out the complex nature of the relationship between belief and practice. However, the problems and objects of study in SE require approaches suited to their dynamics and contexts. For example, the real world benefits, obtained by changing from one software development practice to another, are still poorly understood. The details of how beliefs form practice on a daily basis of a project team remains underexplored and the connection between beliefs and project team goals has not been carefully explored.

The nature and effect of the belief systems underlying current SE theory and practice are related to the reasons for the differences in approach to SE development. This diversity may be seen by practitioners and their managers as being valuable choices between several different methods, techniques and tools. Thus, understanding the reasons for and impact of these differences strengthens the effectiveness of research in the SE area.

2.2 ORGANIZATIONAL CULTURE

Culture is a pattern of shared basic assumptions that a group learned when solving problems of external adaptation and internal integration and has worked well enough to be

considered valid and, therefore, to be taught to new members as the correct way to perceive, think, and feel in relation to those problems (Schein, 1992). Culture is the learned result of group experiences and its presence is believed to be revealed in the patterns of ideas, morals, customs, attitudes and actions of individual organization members. It is intangible, like an abstraction from concrete human behavior and it cannot be realistically disconnected from the thoughts and feelings which constitute the individual (White, 1959).

For Schein (1992), culture is something separate, a distinct entity which identifies and differentiates a social group. It can be seen as a set of variables peculiar to a particular society that can be managed and changed to meet managerial needs, revealed in the patterns of attitudes and actions of individual organization members. The significance of culture is particularly strong in the corporate world, where new assumptions and beliefs are created, discovered or developed by a given group.

Corporate culture can be seen as the social or normative glue that holds an organization together. For practitioners, it provides a way of keeping their organizational world closer to their lived experience and expresses the key values, social ideals and the beliefs that organization members come to share. It is related to the institutionalized way of thinking and acting of people in a company. Organizational culture is seen as being central to organizational success rather than other factors such as physical structure, strategy or formal rules and procedures. It includes the organization's self-image, as well as constitutive and regulative rules that organize beliefs and actions in light of the image (Smircich, 1983).

Organizational culture is manifested as an informal and hidden force, which arises over time during the organization history. This force exerts tremendous influence on the models of behavior of its employees and how they act and justify their actions (Kym and Park, 1992). A realignment of status, power, and working habits follows the implementation of new technologies and practices in line with groups' shared values and meanings. Capturing the organizational culture is helpful in understanding why a new technology or practice does not take place in an organization. It will also be helpful to determine why a project fails or succeeds in the adoption of a certain practice.

Thus, organizational culture forms the context in which software development takes place and directly affects the process of selection and use of technologies and practices in industry software projects with respect to team beliefs, attitudes, intention and behavior.

Organizational culture plays a role in managerial processes that directly or indirectly influence software projects and their context (see Figure 3). Variation across cultural values may lead to different perceptions and approaches in the software development process. Culture helps to illuminate organizational situations. For example, if, for a given group, only ideas that survive an intensive debate are worth acting on, the group members will never engage in a decision process without prior discussion. This is an underlying force within the group that determines the behavior patterns of its members.

The assimilation of a new practice or behavior in SE projects requires either the practice to fit the organizational culture or the culture to be shaped to fit the behavioral requirements of the new practice. So, a proper fit between the values and beliefs incorporated in the software development context and the organization's overall values will lead to a more successful implementation (Strode, Huff and Tretiakov, 2009), (Tolfo and Wazlawick, 2008).

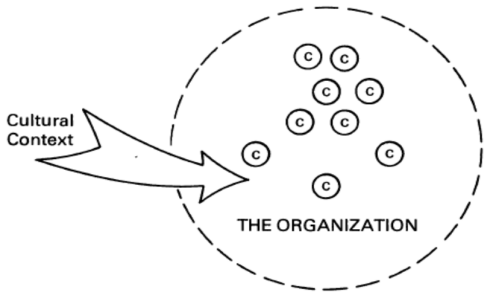

Figure 3 Organization and Culture (Smircich, 1983)

Organizational Culture Framework

According to Schein (1992), culture can be analyzed at different levels or degrees to which the cultural phenomenon is visible to the observer. The most central issue for organizations is to understand the deeper levels of their cultures and how these levels are related.

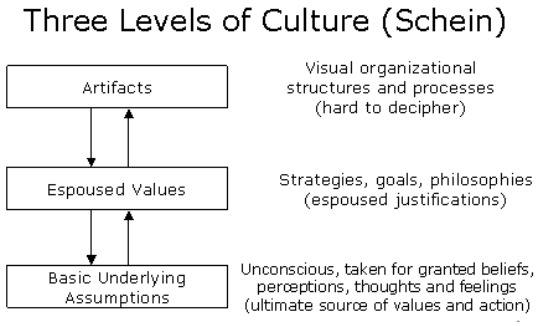

Figure 4 Levels of Culture (Schein, 1992)

The essence of a group's culture is its pattern of shared, basic assumptions that represent a degree of consensus derived from repeated success in implementing certain beliefs, values and a given norm of behavior. According to Figure 4, basic underlying assumptions provide a basic sense of identity; actually guide behavior and lead group members to think and feel about ongoing events, activities and human relationships. People will be comfortable with others in a group who share the same set of assumptions, and be uncomfortable in situations where different assumptions operate. However, when members of a new group bring their own culture from prior groups, the new group will develop modified or new assumptions in critical areas of its experience and form the basis for collective action.

At the next level, espoused beliefs and values represent more visible ideals, ideologies, aspirations and goals of a group or organization. Espoused values are often embodied in an ideology or organizational philosophy and provide meaning and comfort to the group. They provide a mechanism through which organizational members interpret events, actions, and other issues. In this sense, values can be seen as a set of social norms that define the rules or context for social interaction through which people act and communicate. Espoused values are abstract, and many times, contradictory. In many organizations these values reflect the desired behavior, but not the actual behavior. So, in analyzing espoused values, it is important to discriminate carefully those which are consistent with the underlying assumptions that guide behavior, from those that are part of the ideology of the organization or only aspirations for the future. Thus, to get a deeper level of understanding and to predict future behavior correctly, it is necessary to grasp more fully the level of basic assumptions.

At the surface level, culture is manifested through artifacts that include the visible products of a group or organization. These artifacts usually involve the physical environment,

language, technology, organization myths and heroes, noticeable behavior patterns, observable rituals and ceremonies. Formal descriptions of how the organization works and organization charts also are within the artifact level.

While assumptions may be preconscious and invisible, espoused values are more visible, even debatable, with individuals having a greater awareness of them. The use of new artifacts might act to either reinforce or reshape existing values. But, although the artifacts are the most visible manifestation of culture, they can be hard to decipher. Observers can describe what they see at the artifact level, but they cannot ensure what those events mean in the given group. Just if the observer lives in the group long enough, the meanings of the artifacts will gradually become clear. Perhaps due to this issue, the majority of prior work in the SE area, aimed at exploring the relationship between organizational culture and practices, has done so in terms of values-based theories of culture (Dubé and Robey, 1999), (Robinson and Sharp, 2005), (Alavi, Katworth and Leidner, 2006), (Iivari and Huisman, 2007), (Tolfo and Wazlawick, 2008), (Strode, Huff and Tretiakov, 2009) (Iivari and Iivari, 2011), (Tolfo et al., 2011), (Passos, Cruzes and Mendonça, 2013a), (Passos, Cruzes and Mendonça, 2013b). These are important empirical studies in the SE area that consider the influence of organizational culture on the software project practices.

Alavi, Katworth and Leidner (2006) investigated the types of cultural values that exist in organizations and how the cultural values might be associated with certain types of knowledge management practices, technology choices, and related outcomes. For them, good cultural values such as sharing, openness, and trust will lead to positive behaviors. Therefore, organizations should seek to promote and build the types of cultural values that support their specific objectives. Consistent with Alavi, Katworth and Leidner's line of thought, Tolfo and Wazlawick (2008) aimed at identifying aspects of organizational culture that may influence favorably or unfavorably the use of extreme programming (XP) practices. They analyzed dimensions of organizational culture from the perspective of the values of XP method. From Iivari and Huisman's (2007) standpoint, organizational culture was interpreted in terms of the Competing Values Model (CVM) which focuses on values as core constituents of organizational culture in two dimensions: change versus stability and internal focus versus external focus. The combination of them forms distinct types of culture. In the same direction, Strode, Huff and Tretiakov (2009) studied the relationships between culture indicators

adopted from the CVM. In this context, Iivari and Iivari (2011) proposed a number of hypotheses about the relationship between organizational culture and the deployment of agile methods. The hypotheses was to a great extent explained by the dimensions of organizational culture identified in the CVM and by the fact that agile methods are contrasted with ad hoc development and with traditional methods.

As Dubé and Robey (1999) proposed, an interview approach based on the storytelling technique is appropriate to obtain a qualitative, interpretative, and contextual understanding of organizations and interpret the culture implications for organizational practice. Stories contain specific details about organizational members, events, and actions with times and places. Stories are useful in research on organizational culture because they contain interpretations of organizational events and policies. So, insights into software development practices can be easily revealed through a cultural interpretation of organizational stories.

The research reported in this dissertation and its outcomes are related to the Dubé and Robey study in several ways. First, we both conducted exploratory and interpretive case studies that generated qualitative data for analysis and interpretation. Second, we extracted stories from the interviewees and analyzed them to identify and group the content themes, and produce broader cultural themes that represent the organization's cultural context. The interviews were designed to gather stories about particular events in the work context of software development practices. And finally, based on stories and project artifacts, we provided valuable guidelines for managers and researchers as they seek to effect and understand the influences on software development practices.

However, unlike the Dubé and Robey study, our research involved ethnographic work during the first cycle, which lasted 18 months. In particular, ethnography is a good way of getting close to the reality of a social phenomenon (Passos et al., 2012). Besides also referring to Schein's organizational culture framework (Schein, 1992) consisting of the three levels of culture, Dubé and Robey (1999) applied three perspectives (integration, differentiation, and fragmentation) as alternative interpretive frameworks for data collection and analysis of cultural data. They sought insights available from all three cultural perspectives by incorporating multiple interpretations as part of their research method. In the second cycle of our research, we chose to conceptualize organizational culture in terms of the

three levels of culture, more specifically by basic assumptions and espoused beliefs and values in accordance with the TRA and TPB models.

2.3 BEHAVIORAL THEORIES

2.3.1 The Theory of Reasoned Action

The Theory of Reasoned Action (TRA), showed in Figure 5, is derived from the study in the social psychology setting and seeks to explain and predict individuals' actions and practices. Fishbein and Ajzen (1975) state that people's actions are mostly rational and based on a systematic evaluation of the information available to them. In other words, people consider the implications of their actions and act based on a reasonable assessment of those implications.

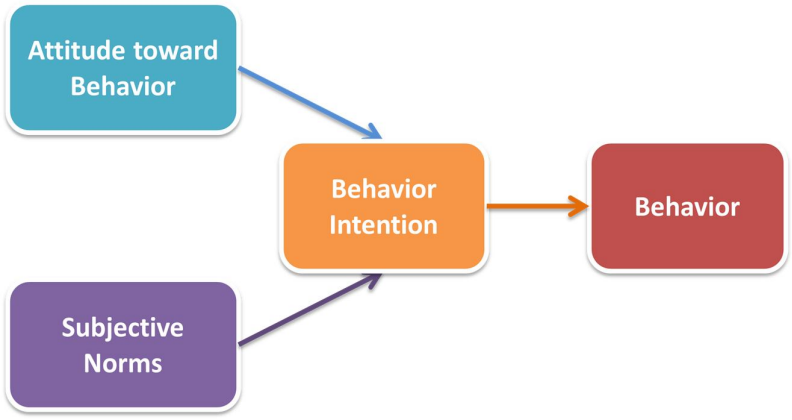

Figure 5 The TRA Model

Fishbein and Ajzen (1975) gave us robust definitions of attitude and behavior. A solid body of work for a more uniform study of these terms sprang from their research (Hennessy, 2012). They sought out a way to not only predict behavior, but also to understand its relationship with beliefs, attitude and subjective norms. The subsequent separation of

behavioral intention from behavior allows for the explanation of limiting factors on attitudinal influence (Ajzen and Fishbein, 1980).

TRA suggests that a person's behavioral intention depends on the person's *attitude* about the behavior and *subjective norms*. It means people behave based on their attitude and how they believe others would think about their behavior. Behavioral intention measures a person's relative strength of intention to perform an action. The more favorable the attitude and the subjective norm, the stronger the person's intention to perform the action in question will be. So, TRA has proved to be very useful in understanding human behavior and it is largely used in health assessment studies (Montano and Taplin, 1991), (Vanlandingham et al., 1995), (Sutton, McVey and Glanz, 1999), (Randolph, 2009), (Roberto et al., 2011), (Blank and Hennessy, 2012).

As an adaptation of TRA to the field of Information Systems (IS), the Technology Acceptance Model (TAM) (Davis, 1989) is one of the most influential extensions of Fishbein and Ajzen's theory in software related literature. It suggests that two specific beliefs – perceived ease of use and perceived usefulness – determine one's behavioral intention to use a technology. The idea is that the easier a technology is to use, the more useful it can be and enhance job performance. When TAM was used in a voluntary environment, attitude did not add additional information to the explanatory power of the model. Therefore, the model was altered by Davis and Venkatesh (1996) and the variable attitude was omitted. The authors affirm that the omission of attitude helped to better understand the influence of perceived ease of use and perceived usefulness on the key variable of interest – intention. Although TAM has received extensive empirical support through validations, applications, and replications in the IS area (Adams, Nelson and Todd, 1992), (Limayem, Hirt and Cheung, 2007), (Kerimoglu, Basoglu and Daim, 2008), (Jan and Contreras, 2011), (Venkatesh and Davis, 2000), it is not so suitable for explanations involving issues related to Software Engineering (SE) theory and practice, especially because TAM does not consider attitude as an important dependent variable of interest, ignoring the essential social processes of software development.

On the other hand, TRA has created a specific definition of *attitude* separate from beliefs, subjective norms, behavioral intention and behavior, allowing each term to have its own separate role in the theory (Ajzen and Fishbein, 2005). Attitude is essentially information we have obtained about someone or something that we form an opinion or predisposition

about. All new information has the potential to affect attitude based on the strength of beliefs. For TRA, each person is able to process information in a systematic manner forming opinions and comprehension, not just as a passive listener.

In particular, the TRA model provides a good framework to understand the influence of a belief system on team practices. Moreover, TRA was tested in organizational contexts several times (Becker, Randall and Riegel, 1995), (Limayem, Hirt and Cheung, 2007), (Peslak, Ceccucci and Sendall, 2011).

2.3.2 The Theory of Planned Behavior

In more recent publications, Ajzen (1985) has extended the Theory of Reasoned Action (TRA) to the Theory of Planned Behavior (TPB) by including a measure of *perceived behavioral control* (see Figure 6), which it is argued will increase the prediction of intention and behavior in those instances where the behavior not entirely under the control of the individual or group.

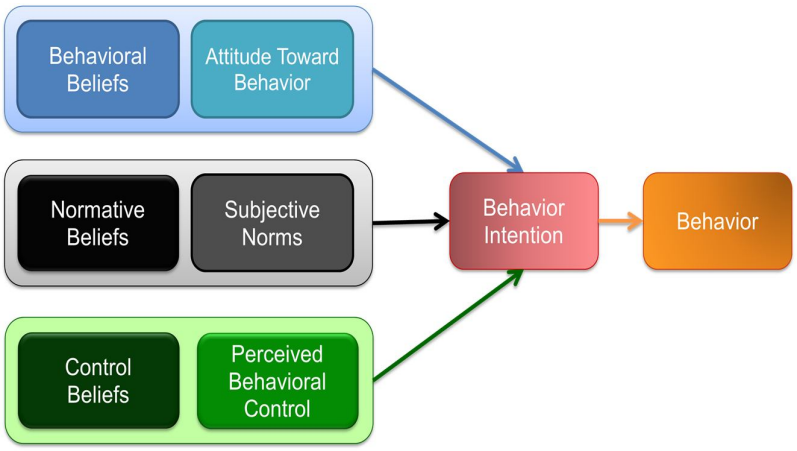

Figure 6 The TPB Model

Although the TRA model can predict the probable behavior, it may not predict the actual behavior, because people do not always do what they intend to do and there may be other factors that will cause them to go against their initial intention. When used to explain

behavior that is not fully a conscious choice or decision, the TPB is expected to perform better than the TRA. According to TPB theory, human behavior is guided by three kinds of considerations: (i) beliefs about the likely outcomes of the behavior and that the evaluations of these outcomes produce a favorable or unfavorable attitude toward the behavior; (ii) beliefs about the normative expectations of references, i.e., about opinions of important others; and (iii) beliefs about the presence of factors that may facilitate or inhibit behavior performance and that the perceived power of these factors gives rise to perceived behavioral control.

The evolution of the TPB and research is marked by a debate about the meaning of the third variable (Yzer, 2012). The current dual-aspect conceptualization of perceived behavioral control is determined by two important factors such as perceived *autonomy* and *confidence* related to how easy or difficult behaviors can be. Those factors can be both internal (knowledge, skill, willpower) and external (time, money, resources, cooperation of others). From this point of view, people believe that they can carry out their intentions when they believe that they have the resources and opportunities to perform the behavior and when they believe that they can freely make the decision to use those resources and opportunities. The perceived power of influence factors that may inhibit behavior performance will cause people to go against their initial intention. On the other hand, when their confidence level related to some subject is substantially high, it is reflected into practices that people actually adopt and seek for positive results. The Theory of Planned Behavior places the construct of self-efficacy belief of Bandura (1982) within a more general framework of the relations among beliefs, attitudes, intentions and behavior. When the behavior control is high, the TPB reduces itself to the TRA.

We believe that TPB can be helpful in generating rich and detailed accounts of software project teams, the interactions between their members, and, especially, the actions oriented toward certain software practices. Both models (TRA and TPB) are theoretical approaches to individual-level processes, however, there are relevant evidence that supports the notion of isomorphism and homology between individual and team-level behavioral constructs and processes (Chen et al., 2002), (DeShon et al., 2004), (Chen and Kanfer, 2006).

Thus, TPB fits in the context of our research within the software industry, because it allows us to study the way beliefs, attitudes and a sense of self-efficacy are formed and their relationship with behavior and practice, with room to explore other relevant aspects.

The TPB model appears to be better for dealing with the complexities of the influence of a belief system on project team practices in organizational contexts. Moreover, TPB has been widely used to predict and explain health-related intention and behavior and these results were successfully replicated (Sutton, McveyAnd Glanz, 1999), (Sutton et al., 2003), (Middlestadt, 2012), (Yzer, 2012).

2.4 COLABORATIVE ETHNOGRAPHIC RESEARCH

Ethnographic research methods have played a substantial role in sociological research during the last half-century and are now used in many disciplines, especially those that involve social and human factors (Crang and Cook, 2007). They have become used to address a good range of research questions related to SE practices (Sigfridsson and Sheehan, 2001), (Karn and Cowling, 2006), (Robinson, Segal and Sharp, 2007), (Sharp, Souza, and Dittrich, 2010), (Boden, Müller and Nett, 2011), (Passos et al., 2011).

In particular, ethnographic methods were very helpful to our research in generating rich and detailed accounts of software project teams, their interactions with project members, their approaches to software practices, as well as their past experiences. However, applying an ethnographic approach to SE can be both challenging and demanding. The problems and objects of study in SE, by their very specific nature, require an expert knowledge of the concepts under study and extensive data collection that lasts for months and years, rather than days and weeks.

Ethnography is about immersing in the culture of a group for an extended period of time in order to tell a credible, rigorous, and authentic story, giving voice to people in their local context. It is about participating in social relations, seeking to understand actions within the context of an observed setting and how people act and make sense of their environment. Typically, the ethnographic story relies on verbatim quotations and descriptions of scenarios that allow an inside perspective of the context of the people under observation. A key purpose

of ethnography is to provide a detailed, in-depth description of everyday life and practice (Forsey, 2010).

Ethnography derives from traditional anthropology where time in the field is needed to get a comprehensive description and understanding of a setting, group, or culture under study (Fetterman, 2010). For most anthropologists, it involves actually living in the communities of the studied people, participating in their activities, interviewing them, drawing maps of the context, and collecting artifacts (Hammersley, 2006). Additionally, since ethnography aims to generate holistic social accounts, ethnographic research can identify, explore, and link social phenomena, which, on the surface, seem to have little connection with each other.

Ethnographers adopt a cultural lens to interpret observed events, actions, and behaviors, ensuring that they are placed in a culturally relevant and meaningful context. The acquisition of knowledge in ethnographic research is a cyclical process. It begins with a panoramic view of the community, closes in to a fine focus on details, and then goes back to the larger picture again, but this time with detailed information. Concurrently with data collection, ethnographers devote a large amount of their fieldwork time to formally analyze and reanalyze their data (Fetterman, 2010). While still in the field, interim reports are produced to enable feedback on the data-gathering cycle. Narratives can be made and revised to take into account the ethnographer is evolving knowledge.

In this context, the primary research technique is *participant observation*, which involves direct observation, participation in the life of the observed group, and collective discussions through an intensive involvement with people in their natural environment, usually over an extended period of time. The traditional principles of participant observation and ethnographic fieldwork require a full immersion of the researcher in the chosen field of study, learning the day-to-day and extraordinary essence of social and cultural life by being there (Robson, 2011).

Critics of the approach are concerned about researchers getting over-involved with the people and environment under study, perhaps disturbing and changing the natural setting (Robson, 2011). By contrast, some sociological ethnographers do not actually live with the people they study or spend most of their time with them. Instead, they focus on what happens in a particular institution when in operation; their participant observation is part-time,

but sufficient to complete the research. In any case, ethnography must have a holistic view of the culture or group under study and crosscheck, compare, and triangulate the findings (Crang and Cook, 2007).

The decision of leaving the field should be based on more than one reason (Fetterman, 2010). The best reason is the belief that enough data has been gathered to say something significant about it, but sometimes the proximity of a deadline is the determining reason. Different researchers require different levels of confidence about research findings and conclusions, but when the general picture reaffirms itself over and over again, it is probably time to leave the field and continue the job from outside. Moreover, the researcher must always check whether the data will support the findings or invalidate them.

After leaving the field, the ethnographic work continues with the final stages of analysis in which field notes, interim reports, papers, and so on are used to draw an overall picture of how the practice system works, and which data will sustain or break the findings. Thus, ethnography is both a research method and a product, typically in the form of a text document. The written work can be shared with participants, to verify its accuracy, and with colleagues for review and consideration. If the ethnographer adopts a cooperative approach, he/she might share documents with community members, who can edit and co-write the ethnographic findings.

Nowadays, a more *collaborative perspective* to ethnography is commonly required in organizational contexts. In contrast to traditional ethnography, contemporary collaborative ethnographic research has applied strategies, such as *participatory action research* (Lassiter, 2005), (Sjøberg, Dybå and Jørgensen, 2007), (Fægri, Dybå and Dingsøyr, 2010), (Boden, Müller and Nett, 2011), (Santos and Travassos, 2011), that aim not only to analyze but also to improve work practices. Dittrich, Rönkkö and Eriksson (2008), for example, adopted an ethnographically inspired action research approach, which combines qualitative social science fieldwork, with a problem-oriented method, technique, and process improvement. However, it is noteworthy that ethnography and action research are two different approaches (as shown in Table 1). Ethnography focuses on the culture and values of a target group with the aim of generating rich and detailed accounts to produce an in-depth description of the group's life, while action research focuses on processes and practices and how to improve them in a problem-solving paradigm (Lassiter, 2005).

Table 1 Ethnography vs. Action Research

	ETHNOGRAPHY	ACTION RESEARCH
Goal	Go Native In-Depth Understanding of Culture Generate Rich and Detailed Social Account	Process Improving Knowledge Advancing Problem-Solving
Focus	Culture and Values	Process and Practices
Main Technique	Participant Observation	No specific technique
Fieldwork	Holistic Comparative Contextual	Essentially Collaborative Reflexive Problem-Solving Driven
Final Product	In-Depth Description of Group's Life	Improved practice New knowledge

The challenge is to redirect and reinvent ethnography along such lines as *collaborative ethnography* − the collaboration of researchers and subjects in the production of ethnographic results, involving a side-by-side work of all parties in a mutually beneficial research program. When doing ethnography through participant observation, the researcher must assume an attitude toward being there enough to experience the environment and nuanced aspects of socio-cultural life. This, at first, does not allow a collaborative approach to the research. However, in organizational contexts, it is expected not only to analyze, but also improve work practices.

Engaging in a shared process with the practitioners allows access to some subtle dynamics of the group processes and practices. As participant observers and interviewers we realized, for example, that we needed to give feedback to the study participants about their beliefs and impacts on practices, so that they could act as a unit and resolve the conflicts that hindered the adoption of some practices. We sought to communicate well and openly, while also being honest, trusting, realistic, and objective in our research. As a result, the participants were able to prioritize action to solve their problems. Thus, during this research, we dealt with this challenge in a similar manner as reported in social science literature. We have adapted our approach to ethnography grounded in collaborative research practice and we have engaged in a mutual knowledge exchange with the participants, based on Lassister (2005) and Lewis and Russell (2011).

Another challenge in the context of the collaborative research is the balance between *participant observation* and *participant listening*. Participant observation traditionally involves direct observation as the method of data collection. Other appropriate

methods involve informal interviews, participation in the life of the group, collective discussions, analysis of personal documents produced within the group, self-analysis, results from activities undertaken off or online, and life stories. Participant listening is an important technique employed by ethnographers, particularly among those who live in an interview society, where interviewing has become common practice. It is an appropriate way of participating and getting involved in organizational contexts. The interviews allowed a deeper understanding of the obtained data through our observations (Forsey, 2010).

In the context of this work, beliefs are intangible. It was very challenging to only observe group members and try to understand what each person's beliefs were in the context. The data collected in the interviews enabled connections between the participants' stories and our observations as researchers. Thus, we strongly recommend that researchers be both a *participant listener* and *observer*, because the data captured in the field can reflect more of what is heard than what is seen; casual conversations and formal interviews can be construed as part of what is observed in the field. The same approach is advocated by several authors who argue that the practice of participant listening should sit alongside with participant observation as an equally valid way of gaining ethnographic knowledge (Robinson, Segal and Sharp, 2007), (Reeves, Kuper, and Hodges, 2008), (Forsey, 2010), (Boden, Müller and Nett, 2011).

Chapter

3

This chapter provides a description of our long-term ethnographic case study of cycle 1. It presents the research process and approach applied in the first cycle, including the initial conceptual framework, research questions and goals, data collection procedures employed, instruments used, and the type of data analysis performed. Lastly, this chapter discusses the key findings encountered in cycle 1 by answering the research questions posed.

3 MAPPING THE IMPACTS OF COMMON AND CONFLICTING BELIEFS ON SOFTWARE PRACTICES

The work of this dissertation was divided into two cycles and three case studies, including a methodological analysis about the benefits of using a collaborative ethnographic approach and the synthesis of the results of the second cycle through a cross-case analysis on the research questions posed (see Figure 1). The details of the research approach, design and results for each of these studies are in Chapters 3, 4 and 5.

3.1 CYCLE 1– ETHNOGRAPHIC CASE STUDY

This section presents the approach and design used in the first cycle of the research. It involved a long-term ethnographic case study to investigate the impact of team beliefs on software development practices and a methodological analysis about the benefits of using a collaborative ethnographic approach.

3.1.1 Conceptual Framework

We conducted the first cycle of the research with our own conceptual framework developed based on our perceptions about the main aspects to be studied – the key factors or variables – and the presumed relationship between them.

As illustrated in Figure 7, technical folklore, organizational culture and project context influence beliefs without (or with very little) empirical evidence. Together, these three aspects exert great influence on behavior and practices of SE practitioners and on how they think or make technology decisions and choices. Research findings provide an empirical reference with which beliefs and values can be confirmed. The group named Evidence-Base Practice represents the influence of knowledge from empirical research onto team practice. This influence is based on empirical evidence and also affects beliefs and perceptions about SE practice.

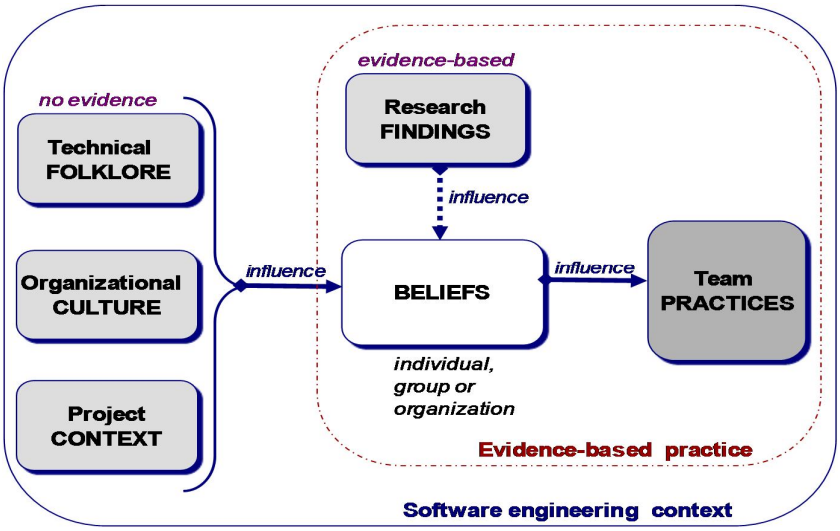

Figure 7 CYCLE 1 – Conceptual Framework

This conceptual framework was built to focus and bound the collection of data and guide our first cycle of the study. It explains the main aspects to be studied in order to arrive at a balanced and comprehensive understanding of beliefs and their impact on team or group practices.

3.1.2 Research Goals and Questions

The main targets of this cycle were to capture and map beliefs origins and sources, identify significant relationships between team beliefs and software development practices, and represent team belief systems and their impact on practices, processes and decisions in software industry projects based on our conceptual framework (see Figure 7). It was also an important goal to provide a support mechanism for organizations, based on conclusions derived from the ethnographic study through a participatory action research approach, on how to use a team belief system to promote software practice improvement.

Thus, the goals of the first cycle of the research were expressed by the sub-research questions as follows:

1. **RQ1.1**: What are the origins and sources of Software Engineering beliefs?
2. **RQ1.2**: How do these beliefs impact current software project practices?
3. **RQ1.3**: What practices are being benefited and hindered by those beliefs?

Regarding research question RQ1.1, we brought forth empirical evidence about the beliefs origins and sources, observing how things actually happen in practice. It was necessary to find ways to capture and understand the impact of beliefs on SE practices and work processes in order to answer question RQ1.2.With respect to question RQ1.3, we found out that there were conflicting beliefs between project team members. Thus, it was necessary to identify salient beliefs and uncover the beliefs which could hinder or benefit the adoption of new software practices.

Another goal was set to this first cycle of the research, it involved a methodological proposal on how to expand the use of empirical results in the software process, including the selection and adoption of new technologies and practices in software industry context. This goal proved to be too difficult to achieve. After a deep exploration of the subject, we now believe that it requires a very extensive and embracing work involving a representative number of participating companies around the world, in order to determine how to present empirical results for different purposes and contexts, it is very difficult to identify what information practitioners need from empirical studies, and what is most convincing for practitioners to believe in it. This kind of effort could not fit in our research schedule, so, at

the end, we decided to limit our work on better understanding how team belief systems and their underlying factors actually influence software industry practices. However, Section 7.3 lists the subject, the study of the expansion of the use of empirical results, as a promising subject for future work.

3.1.3 Research Methods

This research has an essentially qualitative design, where the central research method is case study (Easterbrook et al., 2008) (Runeson and Host, 2008).We applied an ethnographic approach during the first cycle of the research, immersing ourselves in the day-to-day project activities for several months to delve deeper into the complexity of a team belief system. We documented what occurred through a variety of means that included field notes, audio recordings of discussions and meetings, access to several documents and artifacts, and records of interviews with practitioners. We adopted an approach that aimed explicitly to help software companies to improve the work practices under study, getting some characteristics of action research, but mainly based on more modern approaches of ethnography (Lassiter, 2005) (Forsey, 2010) (Boden, Müller and Nett, 2011) (Lewis and Russell, 2011). In our context, this participatory approach was required, so we worked collaboratively to support their process improvement initiative.

Data Collection

In the scope of this first cycle, we collected data from seven software projects with, on average, with the duration of six months each.

The data collection involved semi-structured interviews, observations, context mapping and document analysis, following the rules of Crang and Cook (2007), Fetterman (2010) and Robson (2011). It involved constant meetings and practice observation and the establishment of the frequency and scope of the interviews based on project dynamics and methodology. Our initial conceptual framework (Figure 7) was used to focus and bound the collection of data, and explained the main aspects to be studied, their key factors, constructs,

variables and relationships between them. It was a direct step from this conceptual framework to the research questions designed.

The purpose of this data collection was to obtain a deep understanding of the overall dynamics of processes and activities, comprising the main aspects of the software projects reality. At the same time, it was important to cast light on the respondents' past experiences, beliefs emerged or evolved from these repeated experiences, impacts of new software practices on projects and unexpected effects of known and new methods or techniques.

Interviews

After a literature review and an evaluation of the research objectives and questions, we opted for one interview-based qualitative data collection technique called *Storytelling* and we used a specific story form – the *War Story* (Lutters and Seaman, 2007). Storytelling is both familiar and powerful. It is deeply rooted in the ethnographic approach, which seeks to capture the perspective of study participants and their own perception of reality, preserving the participants' natural language, values, beliefs and mental models. War Stories can form the basis of human communication that can be leveraged to better understand human behavior. It is a holistic approach that accesses both the internal states and external environment of the participant and pays attention to all contextual detail, which is so vital in this sort of research.

We used an iterative approach in which we defined a questionnaire, used it for a set of interviews, analyzed the data, and improved the instrument and process for the next round of interviews. The first step was, therefore, to define the first version of the interview questionnaire. We tried to keep it as simple as possible and still cover the typical questions of a War Story questionnaire. They usually have *warm-up*, *past experiences*, *lessons learned* and *reaction* questions. For this cycle, we had a few of each type.

Warm-up questions aim to put the interviewees in the right set of mind, to focus them on the interview subject. Our first version of the questionnaire had two warm-up questions that asked about the participants' background and experience, present position, and the main challenges of their current software project. *Past experience* questions intend to investigate how living experiences can influence the participants' current behavior, trying to cover the main aspects of a software project reality. Our questionnaire included four of these

questions. *Lessons learned* questions try to capture the beliefs that emerged or evolved from the project experiences. This is the main focus of our research, but to keep the questionnaire simple and balanced for the research kick-off, we only had three War Story questions about lessons learned. They aimed to understand the basis of how beliefs and cultural values of the participants came to be and also how they determined the participants' choices and actions during the projects. *Reaction* questions ask about the participants' reactions and personal opinions on the impacts of the use of new practices, methods and processes during the software project. The questionnaire had two reaction questions that directly asked about the adoption of software practices and the unexpected effects of them.

Besides the typical War Story questions, we added a few questions to identify beliefs related to practices that affected project measures (*Metric-based* questions). It helped to corroborate findings via triangulation with the organization metrics. We had four of them in the interview questionnaire (see Appendix C).

In total, 26 interviews were conducted with people on the front lines of the projects. Each interview was made on-site lasting for 30-40 minutes and in a semi-structured way, to allow the respondents to reflect. Each person was interviewed twice, once at the beginning and once at the end of the project. .

The interviews were recorded and transcribed manually. The resulting 26 interview audios were analyzed according to the principles of the War Story technique (Lutters and Seaman, 2007) in order to identify conformance or divergence in the respondents´ replies. Comparing data from different interviews, it was possible to generate a more comprehensive understanding of social interaction and its subtleties in the organizational context.

As described, our War Story questionnaire went beyond asking questions that allow the respondents to generalize on their past experiences or repeated behavior, it asked them to retell and revive specific and directed stories that illustrate the experiences we were trying to capture. The resulting data contained considerable amount of contextual information, which enabled connections between different, but related, stories.

Observations

The interviews were complemented by meeting observations. Field notes and recordings of relevant interviews on completion of observations in project meetings provided

rich insight into social relations, events, and practices. We established specific goals for our observation task and some guided questions to drive the work (see Appendix B).

During observations, we immersed ourselves in the day-to-day project activities to delve deeper into the culture of the company for an extended period of time. First, we gained access to team meetings through formal roles in the projects. Second, we lived and worked among the practitioners in order to grasp their world views and ways of life, and finally, we returned to make sense of the collected data, through writing up an account of the company's culture. We tried to balance the role of participant observer with rigorous fieldwork and knew how to stand in-between the research subjects, contexts, and demands, and the ethnographic product. Our participant observation approach relied on the participants' verbatim quotations and descriptions which allowed us to obtain an inside perspective of their context.

With respect to meeting observations, we produced audio recordings of all project meetings attended or those of which we were aware. We established specific goals for our observation task and some guided questions to drive the work (Appendix B). So we used the goals as a checklist to remind us about what is important to focus more attention on. The guided questions helped us to build a blueprint to conceptualize the steps to extend knowledge and understanding. We were able to draw social and conceptual boundaries to identify or confirm themes, subjects, problems through questions of interest.

We recorded more than 300 project meetings involving the interviewed participants and all the team members. All these audio files were labeled with event date, project name and sprint id, meeting identification, and some of them were listened to again, especially when we needed some confirmation or better clarification of themes, issues, and points of interest. The audios of the project meetings were not transcribed.

Context Mapping

In pursuing rigor in this research, we were involved in the dilemma of sufficiently broad and detailed contextualization. In our case, contextualization involved placing our observations and interviews into a larger perspective and helped to provide a more accurate characterization of the team practices. No study can capture all aspects of a culture or group context. In our case study, for example, the participation in the project team meetings helped us to realize that there were common strong beliefs, in which team members actively acted on the direction of implementing good software practices, even with several unfavorable

conditions. Only with the contextual information related to the origins and sources of their beliefs and the project environment was it possible to capture and understand how strong these beliefs really were.

For context mapping we used some of the context facets identified in Petersen and Wohlin's (2009) checklist. This checklist guided us on how to interpret the impact of the company and project contexts on the case study results, and also the influence of context elements on each other. We identified three levels of context: company context, project context and project team context. In Appendix A, we provide a description of them and related elements, which described the facets. Regarding the context elements, it is important to mention that we do not claim completeness. However, our checklist proved to be a very good instrument that highlighted what ought to be covered in our context descriptions.

We considered the number of employees, the company's standard process and the software process certifications as very important aspects of context. Overall, we collected more than 35 context facets. They were used to start exploring the research questions. They helped to explain the context of software practices and how these practices were influenced by past experiences of the project teams as they become more habitual. Also, the context of the projects investigated was useful in analyzing the team members' subjective evaluations of organizational or project level interventions and outcomes.

In our case study, we had full access to organizational and project artifacts because the participants were comfortable to give us confidential material. As insiders, they knew we were able to handle it and to interpret the data properly. We compared our field notes from interviews and observations with the processes, methodologies and information about tasks laid down in the company's documents. This analysis resulted in a set of checks and detection of meanings in the text and in the study of the relationships between these meanings to get a different perspective of the data collected. We started with a broad and complex set of data to reach segments which helped us to establish relationships and draw conclusions. For example, we concluded that although the company had a very well defined process for software testing, the project teams did not use it to guide the software tests, which resulted in poor coverage, absence of real test cases and low effectiveness of these tests.

Transcription

The process of transcription of data was a critical element of the initial phases of data analysis. We executed our own transcriptions of interviews and sought to register expressions of behavior in context. We found it useful to pause during the process to make notes of feelings or intuitions we had, and the unspoken factors within the interview, such as laughter, long pauses, or changes in tone of voice. We used line-breaks to specify when the conversation moves from the interviewer to the participant and add in attribution. Also, we recorded the start and end of each stage; the date, time and position in the audio of each break and restart.

For each data collected and transcribed, we created both a representation of the interview experience that was used for subsequent cycles of analysis, as well as a set of memos written that recorded some intuitions and insights into emerging thematic structures that contributed to later interpretations of the data. Our understandings were derived through the process of constructing the transcripts by listening and re-listening the records to convert original oral language into a written form.

Coding

We analyzed all the information transcribed via cycles of coding, identifying specific segments of text to label, categorize and translate them into themes. We employed constant comparison, scoring, scaling and clustering of themes. Our process of coding transformed qualitative data into belief classes, so it was possible to identify the relationships between these beliefs and software development practices as well as other key factors of influence. The next step was to validate the code findings with another researcher to seek the completeness of the conclusions. The purpose was to get another researcher to double-check the codes and data to tag the key words, phrases and paragraphs, and group them in short segments of data sets and useful constructs.

After reducing the data to a limited number of classes of beliefs, we characterized them in terms of frequency, origins and sources. We also recorded information about their context and their meaning to each participant of the study. Then we identified and analyzed the effects and impacts of these beliefs on the software team practices during the project. It was part of the job to uncover and understand how practices are actually applied by the project team and not just what the team said about them. We also examined the significant

relationships between beliefs and actual project practices (to check if people were actually doing what they were preaching).

We used the results to guide the discussion and validation of the findings with the participants of the study. At the validation session we performed a focus group with professionals involved in the company's software projects from different profiles, such as: Software Architect, Requirements Engineer, Developer, Tester, Technical Leader and Process Quality Assurance, as people with different responsibilities in the company would have different perceptions. This gave us a bigger picture of the main challenges, their relative importance, and their causal relationships.

The process of transcription and coding was time consuming. We spent an average of five hours of transcription and one hour of preliminary coding for every 30 minutes recorded. The transcripts of the 26 interviews produced 208 pages of text and an average of 15 codes per transcription. Considering cycle 1 of the research, we built about 26 patterns of code that we translated into 5 main themes, focusing on the relevant actions, interactions and events in the past and current projects that might exert influence on the teams' behavior and practice. The collected, transcribed and coded data produced relevant results and conclusions that were used to guide the next cycle of the study.

Table 2 CYCLE 1 – Case Study Data

Attribute	Data
Duration	18 months
Software Projects involved	7
Context Mapping	+35 facets
Observations	300 meetings
Interviews	26
Interview Duration	30-40 minutes
Interview Transcription	208 pages of text
Coding Transcription	15 codes per transcription
Patterns of Code	26
Main Themes	5

Data Analysis

While the first round of data collection was still in the beginning, we started early data analysis, because it helped data-gathering cycle back and permitted the production of interim reports. Our intention was to anticipate what was going on, how things were proceeding and why things occurred as they did, and also, drive the work as a theory in progress (Charmaz, 2006).

Because of the importance of human activities in software development, many of the data analysis methods that are appropriate to SE are drawn from disciplines that study human behavior at the individual, team and organizational levels (Easterbrook et al., 2008). So, a viable strategy for our research involved more than one method, chosen in such a way that the weaknesses of each method can be addressed by the use of complementary methods. Thereby, we chose a combination of three complementary qualitative approaches – ethnographic and thematic analysis in a within-case analysis. The projects investigated during the first cycle of the research were treated like a series of independent empirical studies that confirmed or not emerging conceptual theories.

Ethnographic Analysis

In each case, we incorporated into the analysis captured sources of beliefs and their real impact on software engineering practices. During the process there was a constant iterating between data, literature and emergent theoretical framework, so it was possible to reach a more complete understanding of software engineering practices in a project team context. Thus, this approach provided a framework for studying the culture of the software engineering industry setting, and for uncovering knowledge, beliefs and values which affect software practice.

During the ethnographic analysis and report writing, we used field notes, interim reports and published papers to draw an overall picture of how the practice had worked and which data sustained or broke the findings. When the general picture reaffirmed itself over and over again, it was time to leave the field and continue the analysis from outside. We shared the written work with the participants, to verify its accuracy, and with other researchers for review and consideration. We also shared our findings with SE community members during conferences and symposia, who gave us important feedback about our work.

Thematic Analysis

The thematic analysis process consisted of identifying, analyzing, and reporting patterns (themes) within data from our primary case study. Themes reduce large amounts of codes into a smaller number of analytic units. Using Boyatzis's (1998) approach of building a quantitative description of the frequency of themes and forming clusters of themes, we analyzed all the information collected via cycles of coding which tagged key words, phrases

and paragraphs; identified relationships and patterns; and grouped summarizing segments of data into sets, themes or constructs(see Figure 8 below).

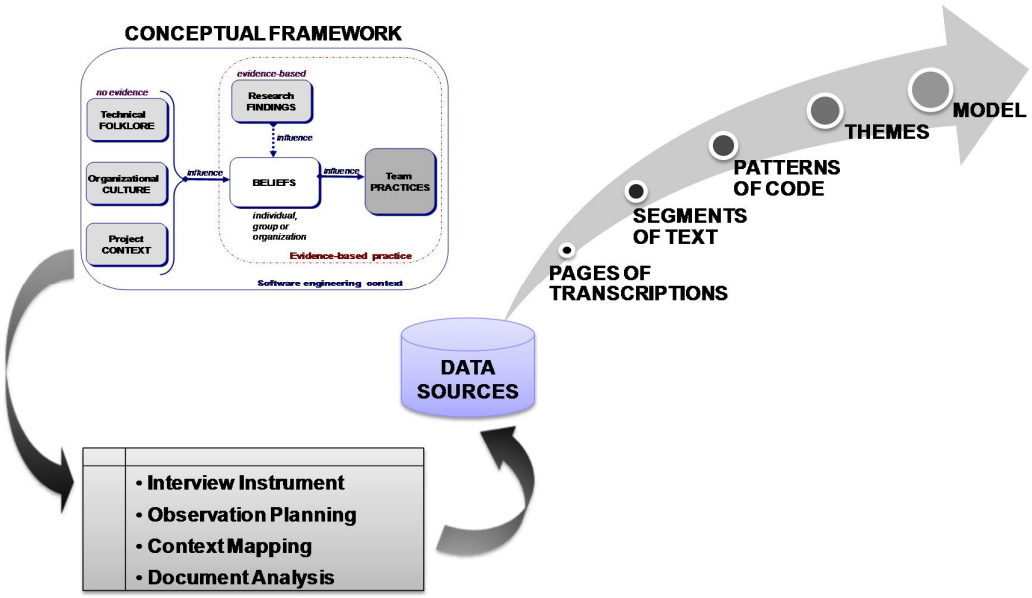

Figure 8 Data Collection and Analysis Process

It was not a single step process; as we analyzed new codes, some of our first round codes was subsumed by other codes, relabeled, or even discarded. As we progressed in the translation to themes, there was some reclassification of coded data into different codes or new codes emerged. The end of this process was reached when we saturated the possibilities of themes emerging from the data collected.

The within-case analysis provided conclusions about what was happening in our case study and supplied the basic material for descriptions, explanations and deduction of implicit theories. Usually it is hard to explain something satisfactorily, therefore, a natural progression is from telling a first story about the phenomenon to building a theory model toward a deeper understanding. Thus, during the data transformation, the evidence was condensed, clustered, sorted and linked over time. We moved through a series of analysis episodes that condensed the case study results into a more coherent understanding of what was happening, how, and why. We represented our main findings in a mind map format (Attride-Stirling, 2001), which was useful to visualize, structure, and classify the beliefs and to organize related information as described in the next section.

3.1.4 Case Study Context and Description

The first cycle of the research involved an ethnographic case study set in a medium sized software company[2] in Brazil, in which projects were applying, for the first time, an agile methodology. This organization has one development center with ISO-9001 and MPS-Br (CMMI like) certifications and three offices in important regions of Brazil as illustrated in Table 3.

Table 3 Company 1 Profile

Company	Age	Operation	Segment	Personnel	Standard Process	SW Process Certification
1	03 years	Brazil	Full IT Service Provider	400	Agile Methodology	ISO-9001 MPS-Br Level E

The company focuses on the development and implementation of initiatives and services that bring innovation to customers, and provides services in software development and evolution as well as information technology infrastructure management services for customers of public and private sectors.

The seven projects under ethnographic study are shown in Table 4. They involved agile and traditional software development with, on average, a duration of six months. The project teams followed a well-defined process for software development. The author of this dissertation had an active role in the target projects because she acted as a quality assurance and process control consultant for the company investigated. There were a variety of domains and some of them related to Brazilian government issues. Each project team had an average of four participants and most of them had worked with JAVA Platform and adopted open source tools for stories and bug tracking. The project teams kept detailed records of project status reports and backlogs for the control and management of software changes. Other important documents, including quantitative measures, were used for project tracking and monitoring. This practice was clearly influenced by past experiences of team members in a CMMI certification program in other companies.

[2]SOLUTIS Tecnologias. http://www.solutis.com.br/

Table 4 Projects Profile of Company 1

Project	Main Goal	Domain	Customer Type	Participants	Duration	Technology Platform	Methodology	Metrics
PGE-1	management of electrical energy consumption	energy industry sector	private	4	6 months	JAVA Platform Open Source Tools	Agile Methodology	productivity quality sprint burn-down
PGE-2	electrical energy interface	energy industry sector	private	3	3 months	JAVA Platform Open Source Tools	Agile Methodology	productivity quality sprint burn-down
SIG	management of financial information	financial sector	private	4	7 months	MS .NET Platform	Agile Methodology	effectiveness cost
SFC	cash flow control	financial sector	private	3	5 months	MS .NET Platform	Waterfall Methodology	productivity quality
SGC	campaign management	government	public	5	3 months	JAVA Platform Open Source Tools	Waterfall Methodology	productivity quality
DMS	tax service control	government	public	9	7 months	JAVA Platform Open Source Tools	Agile Methodology	effectiveness velocity quality, cost
SIIC	management of cultural information	government	public	3	17 months	PHP Platform Open Source Tools	Agile Methodology	effectiveness velocity quality, cost

Sampling is critical for later analysis (Miles and Huberman, 1994). The choices made place limits on the conclusions and study results, which are connected directly to the research questions. We employed the technique of participant observation in seven software projects for 18 months to explore the research questions defined. We considered them a representative sampling in the context described. We worked with real world project data and the participants chosen for the interviews were experienced project managers, technical leaders, developers or testers. They were capable of accurately and reliably answering the interview questions. The frequency and scope of this collection was based on the project dynamics and methodology. We gave greater focus on the most important projects for the company (PGE-1, PGE-2 and DMS). So we believe we covered the most relevant project circumstances with our approach.

After a field contact with people from the software development center, a contact summary sheet was produced (see Table 5). A contact summary is an overall summary of the main points in the contact such as: the main concepts, themes and issues that arise from the interaction between the researcher and the participants of the study.

Table 5 Contact Summary Form

Company:	1	Site:	Temporary Office
Contact Type:		Contact Date:	Wednesday, September 1, 2010
X	Work	Today's Date:	Monday, September 20, 2010
___	Visit	Written by:	Carol Passos
___	Phone		
Project Target:	**PGE-1**		

	THEME / ISSUE	**SALIENT POINT**
1	The start-up of Company 1	PGE-1 is the first software project of Company 1, which started its operation in May 2010 with the employees working in home-office mode.
2	The project PGE-1 progress	Lack of infrastructure resulted in rework and impacted negatively the project. It contributed to delay the project PGE-1.
3	The project PGE-1 schedule	PGE-1 project team worked very hard to compensate time wasting, because of the infrastructure problems, and to be on schedule again.
4	The history of PGE-1 project team	Quantitative measures are used for project tracking and monitoring. This practice is clearly influenced by past experiences of team members in the CMMI certification programs in other companies.
5	The Introduction of SCRUM agile methodology	This project is applying an agile methodology based on SCRUM. It is a new practice for most of the team members.
6	The Introduction of a new technological platform	Company 1 decided to introduce a new architecture for software development based on JAVA Platform and Open Source Software (OSS).

A reflection on the contact summaries like this provided some important information about how the company's projects were being conducted. The project teams have followed a well-defined process for software development and adopted open source applications for project management and bug tracking. The lack of infrastructure at the very beginning of Company 1's life impacted the first projects negatively because the team members worked individually in a home-office setting using a stand-alone operation environment. At this moment, Company 1 decided to introduce new technologies and practices in its projects; the main goal was to improve team performance. Also, the past experiences of important team members influenced the adoption of the practice of using quantitative measures for project tracking and monitoring.

In January of 2011, Company 1 opened a new office in which the software development center was installed. In this new space all project team members could work together, using services provided by a robust data center to support the software development activities. Table 6 describes the major physical characteristics of Company 1's headquarter.

Table 6 Infrastructure Summary Form

Company:	1	Site:	Headquarter
Contact Type:		Contact Date:	Monday, January 10, 2011
___	Work	Today's Date:	Tuesday, January 11, 2011
_X_	Visit	Written by:	Carol Passos
___	Phone		

Observation Target: Software Development Center

Infrastructure Attribute	Description
Location	Salvador - Bahia - Brazil
Size	576 m2
Capacity	84 employees
Local Phone Extensions	44
Internet Routing	10 Mb dedicated link
Data Storage	over 3 Tb
Data Center Servers	10 server computers, including: 1 mail server, 1 file server, 1 web server, 2 database servers, 2 application servers.

Furthermore, the contact summary was useful to help with the coordination of field work, suggest new attributes to be collected for project profiling (see Table 7, Table 8 and Table 9 as examples), guide planning for future contacts with other projects, and help with further data analysis.

Table 7 PGE-1 Profile

Attribute	PGE-1 Project
Domain	Energy Industry sector
Duration	6 months
Customer Type	private
Team Size	4
Team Members	2 full time developers, 1 part time developer, 1 scrum master, 1 product owner + project manager
Technology	JAVA Platform
Tool Usage	Open Source Software
Development Methodology	SCRUM + CMMI Approaches
Non-Functional Requirements	reliability, high performance, continuity, availability
Reuse	high
Stability Requirements	medium stability
Staff Turn-over	low
Metrics	productivity, quality, sprint burn-down

Table 8 PGE-2 Profile

Attribute	PGE-2 Project
Domain	Energy Industry sector
Duration	3 months
Customer Type	private
Team Size	3
Team Members	1 full time developer, 1 scrum master, 1 product owner + project manager
Technology	JAVA Platform
Tool Usage	Open Source Software
Development Methodology	SCRUM + CMMI Approaches
Non-Functional Requirements	reliability, availability
Reuse	high
Stability Requirements	low stability
Staff Turn-over	medium
Metrics	productivity, quality, sprint burn-down

Table 9 DMS Profile

Attribute	DMS Project
Domain	Government - Tax Control
Duration	7 months
Customer Type	public
Team Size	9
Team Members	5 full time developers, 1 full time tester, 1 part time tester, 1 scrum master, 1 product owner + project manager
Technology	JAVA Platform
Tool Usage	Open Source Software
Development Methodology	SCRUM + CMMI Approaches
Non-Functional Requirements	reliability, high performance, continuity, availability
Reuse	high
Stability Requirements	medium stability
Staff Turn-over	high
Metrics	effectiveness, product quality, velocity, cost.

In addition, we identified a very important organizational structure that acted as an internal service provider to the software development projects. Company 1's competency center provided expertise for project support, acting as repository of knowledge for software architecture modeling, software language skills, software testing, methodological innovation, software component development and network design. It was a center of excellence (COE)

with a shared service function within the company for performing some important tasks during the project development. Table 10 summarizes the main attributes of the competency center group.

Table 10 Competency Center Description

Attribute	**Competency Center**
Main Goals	software architecture modeling, methodological innovation, software testing , infrastructure for component reuse.
Location	Software Development Center of Salvador - Bahia - Brazil
Internal Customers	All software projects of Company 1
External Customers	Government
Team Size	4
Team Members	2 senior architects, 1 junior architect, 1 quality assurance.
Technology	JAVA Platform
Tool Usage	Open Source Software
Reuse	high
Staff Turn-over	very low
Metrics	product quality and cost.

We decided to keep all derived contact forms simple and focused on the primary concepts, themes and issues because we used them as instruments to make it easy to do a fast information retrieval and synthesis of what was going on in Company 1.

Regarding document analysis, we analyzed the processes and procedures established in organizational documents and project documentation. We used the information found in these documents to examine, compare and organize the relevant themes or issues of Company 1. It helped to better understand the company and project contexts. The following form (Table 11) illustrates how the document analysis was performed in Company 1. A similar documentation procedure was used during the analysis of the main projects artifacts, such as: project charter, project schedule, actions plan, and project metrics.

Table 11 Document Summary Form

Company:	1	Site:	Headquarter
Contact Type:		Access Date:	Wednesday, March 2, 2011
_X_	Work	Today's Date:	Wednesday, June 1, 2011
___	Visit	Written by:	Carol Passos
___	Phone		

Main Target: Knowledge base of organizational documents - WIKI

General description of document:
processes, methodologies, procedures, templates and forms used to guide the process of software development.

Significance of document:
project team functioning
MPS-Br process assessment (CMMI like)

Brief summary of contents:
software development process composed of documents related to integrated project management, risk and configuration management, measurement and analysis of metrics, organizational process support and definition, organizational training, requirements development and management, technical solution, product integration and quality assurance, software testing and verification.

If document is crucial to a particular group of participants:

(X) senior management	( ) competency center group
(X) project management	( ) IT support group
(X) software project team	(X) quality assurance group

With respect to project meeting observations, we had an active role in the target projects and worked among the practitioners in order to grasp their own points of view about the release planning meetings, when a plan and goals are established and the team decides what will be done and how; the project review meetings, when the team collaborates about what was just done and what to do next; and also the retrospective meetings, when the team is encouraged to revise their own software development practices to make them more effective and productive in regards to people, relationships, process and tools. As our observation approach relied on the participants' verbatim quotations and expressions, we recorded the project meetings involving the interviewed participants and all team members of the seven projects under study. All the audios produced were labeled with meeting identification and date, project name and sprint id. Some of them were heard again during the data analysis process.

The interviews and meeting observations were complementary. In some situations, we identified an issue by interview, and we were able to observe the actions upon it

during a project meeting and, in another time, we obtained data through direct observation and had the opportunity to go deeper during an interview. Thus, the findings could be crosschecked, compared, and triangulated.

All the interviews were recorded, transcribed and headed with information about interviewer, interviewee, total time of audio, transcription and coding, project name and sprint id, and pauses during the process. Table 12 shows an example of how we documented the interviews.

Table 12 Interview Heading

Interview with SCRUM Master of COMPANY 1 – End of PGE-1 Project
09/March/2011 - 18:00hs as 19:01hs
SPRINT: 13

Interviewer: C	**(Role: Researcher and Process Quality Leader)**
Interviewed: B	**(Role: SCRUM Master)**
Total Time Audio:	01 hours and 01 minute
Total Time Transcription:	06 hours and 03 minutes
Total Time Coding:	02 hours and 20 minutes

18/04/2011 – 17:23hs – start

C: .....

B: .....

18/04/2011 – 18:33hs – break in 38:36 minutes of audio

18/04/2011 – 18:53hs – restart from 38:36 minutes of audio

B: .....

18/04/2011 – 19:43hs – final in 01:01 minutes of audio

The participants of this ethnographic study were professionals involved in the company's software projects, using typical development technologies in a typical working environment. Their profiles were diverse and they worked in different positions within the project team, such as: Software Architect, Requirements Engineer, Developers, Testers, Technical Leader and Process Quality Assurance. We sought people with different responsibilities, so we could catch different perceptions.

A fragment of transcription involving our coding process and preliminary analysis is shown as follows.

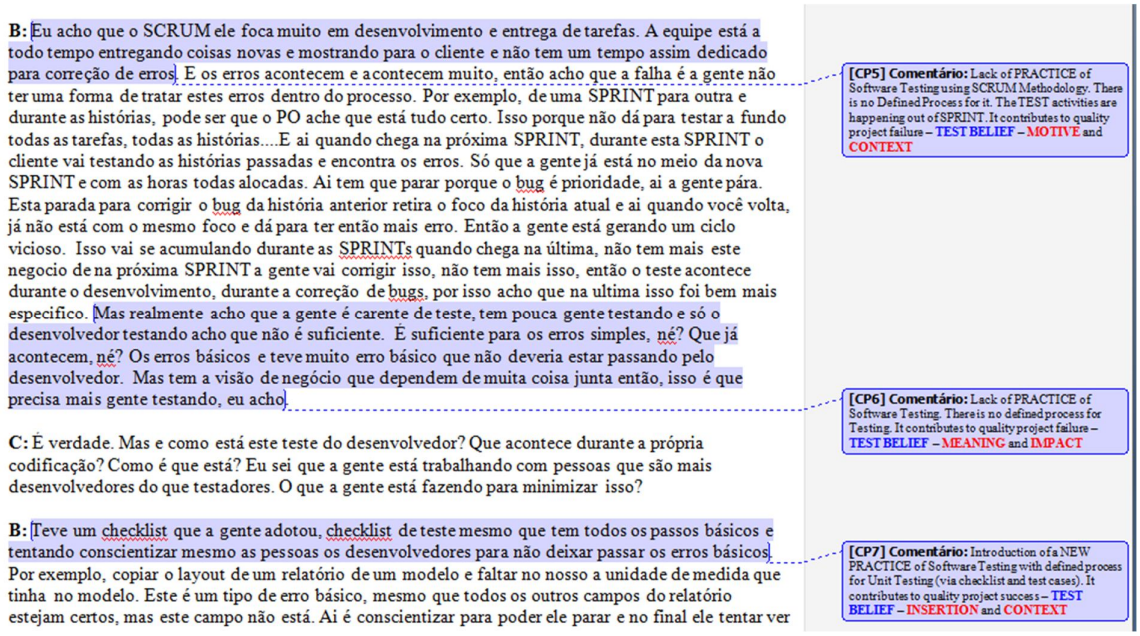

Figure 9 Fragment of Coding Text

After reading the entire text of the transcribed interview at least once after transcription in order to get immersed in the data, we started the cycles of coding. Figure 9 shows how we got to the pattern of code *Software Testing*, which together with the *Test Automation* pattern (from another segment of code) formed the high-order theme *Belief Test* illustrated in Table 13.

Table 13 High-Order Themes

BELIEFS in Common
BELIEFS in Top+ in Impact on Practices
BELIEFS in Conflict
BELIEFS in Major Frequency

			BELIEF'S ATTRIBUTES				
ID	BELIEF TYPE	BELIEF	ORIGIN	CONTEXT	IMPACT	PARTICIPANT ROLE	FREQUENCY
13	TEST	Software Testing	experience in the current project	Lack of PRACTICE of Software TEST (no defined process)	failures in test process; poor product quality; focus on deadline not on quality.	Product Owner	8
14	TEST	Test Automation	experience in the current project	Lack of PRACTICE of Software TEST (no automation mechanism)	poor test coverage; not enough time for bug correction.	Product Owner	7
20	TEST	Software Testing	past experience in previous projects in another company	Strong PRACTICE of Product Quality Control	high effectiveness of tests; high level of testing expertise.	SCRUM Team	10
38	TEST	Test Automation	past experience in previous projects in another company	Strong PRACTICE of Software TEST including Automation	lower regression test effort; higher produtivity during the tests.	SCRUM Team	2

With a clear sense of context, we captured three passages in the text of Figure 9 that exemplify the same theoretical and descriptive idea related to *Software Testing*. Applying a thematic data analytic technique, we established significance by frequency of each theme. In this case, the *Belief Test* theme had three occurrences and represented one of the common

beliefs between two members of the project team. This type of formalization formed an integrated schema for understanding relationships and interactions.

While the data collection was still ongoing, we started a preliminary analysis of the data. A substantial amount of the fieldwork time was devoted to formally analyze and reanalyze the collected data. The ethnographic approach was complemented by a thematic analysis using a within-case strategy. The investigated projects were seen as a series of studies that confirmed or not emerging conceptual theories, related to the key influence factors and presumed relationship between them, represented in our conceptual framework (see Figure 7). During the thematic analysis process, we built a qualitative description of the patterns (themes), using data from our primary case study (the project PGE-1), to form clusters of themes. The within-case analysis supplied the basic material for deduction of implicit theories about the project team behavior and intention. From a natural progression of telling the first story of the first project, to analyzing and condensed all the data collected during the ethnographic study, we were able to summarize the case study results into a more coherent understanding of team beliefs origins, sources and contexts as well as to represent the common and conflicting beliefs and their impact on team practices in a software organization.

We represented our main findings in two mind maps exhibited in Figure 10 and Figure 11. These maps were very important in the organization, categorization and classification of the origins and impacts of common and conflicting beliefs on software development team practices. Excerpts from dialogues, verbatim quotations and descriptions of scenarios and events investigated in the project PGE-1 are described shortly after.

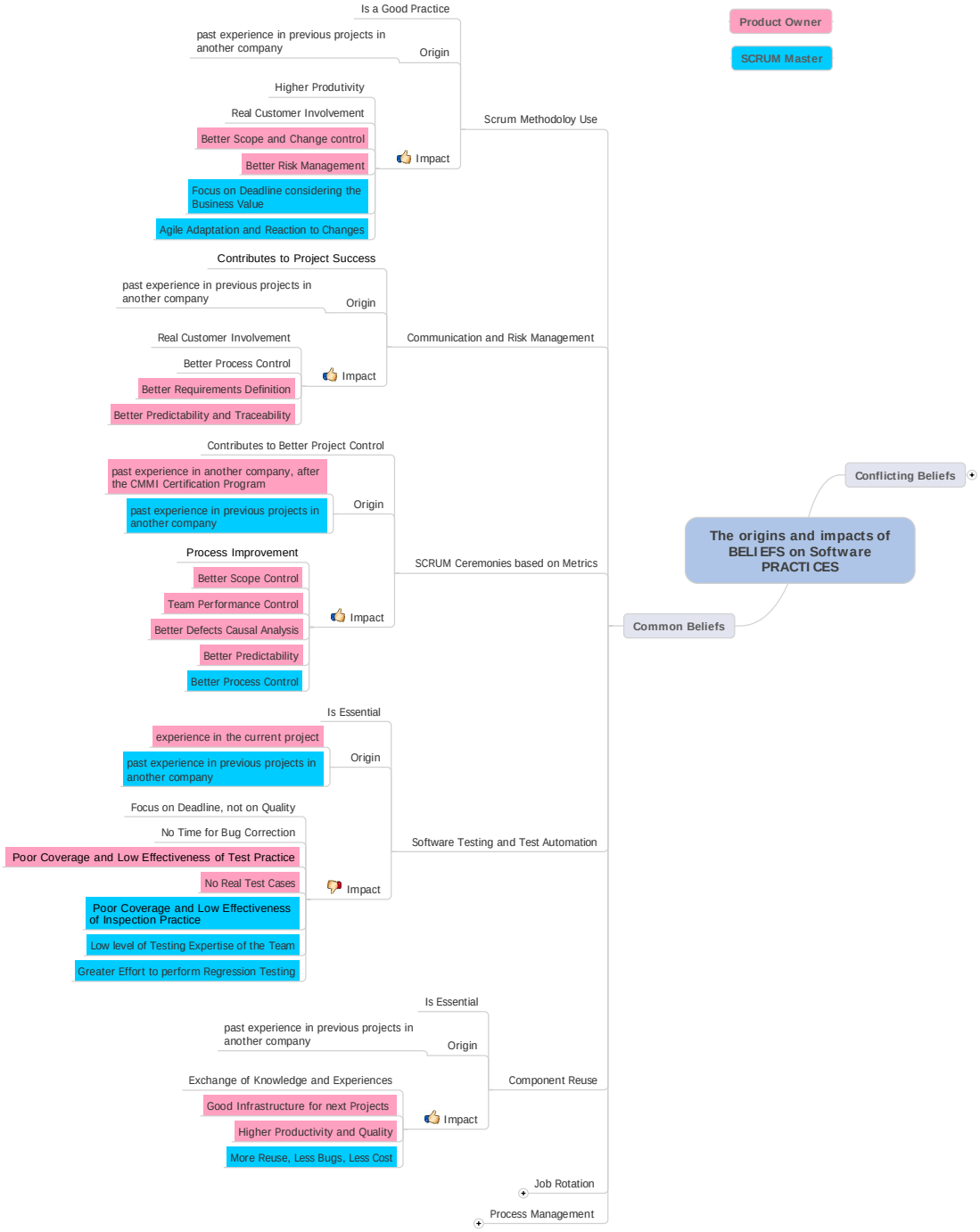

Figure 10 Common Beliefs between Project Team Members

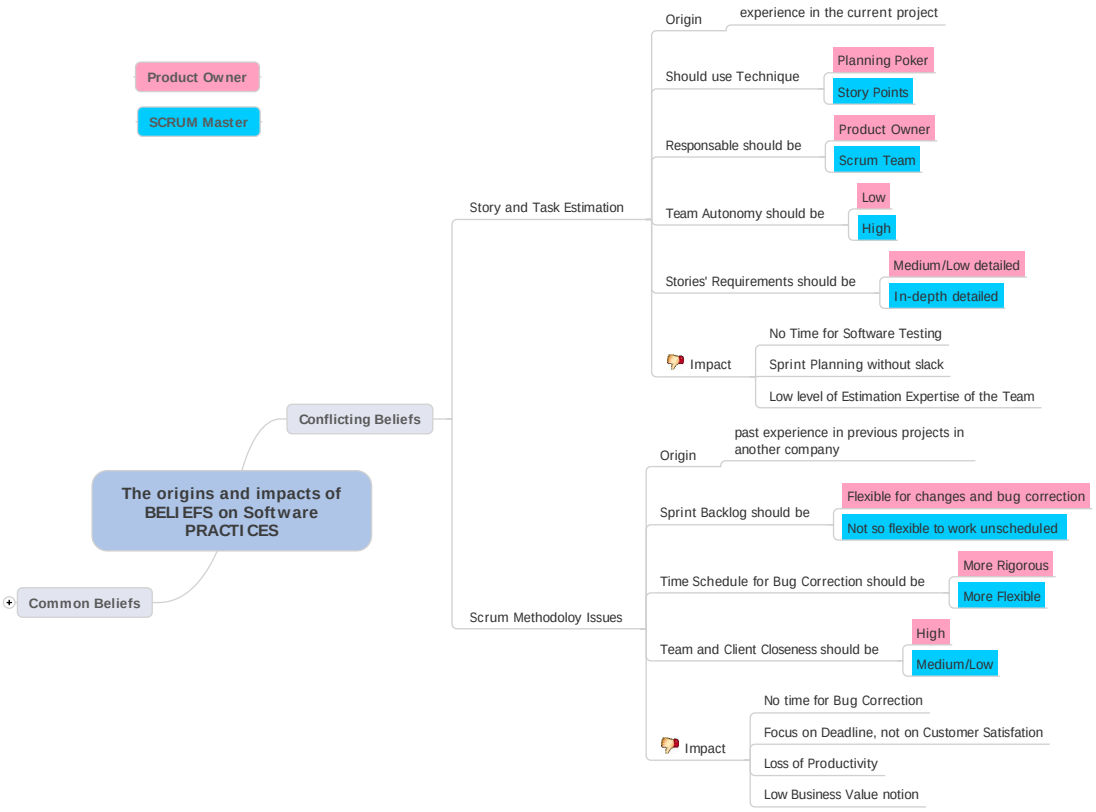

Figure 11 Conflicting Beliefs between Project Team Members

Project PGE-1 was conducted according to the company's agile development approach, which is based on the SCRUM methodology. Two roles are very important in this kind of approach. The Product Owner (PO), who represents the business stakeholders, outlining work in the project backlog and prioritizing it based on the business value, and the SCRUM Master, who ensures that the process is used as intended, guiding the team and keeping it focused on the tasks at hand. Figure 10 and Figure 11 show the beliefs shared by the PO (pink color) and The SCRUM Master (blue color). Nodes without color represent beliefs of both the PO and SCRUM Master.

Project PGE-1 lasted six months. The release planning meetings were made at the beginning of the each sprint, after the team had reviewed what was produced in the previous one. After each sprint, the team held a retrospective meeting to identify and discuss problems and opportunities that arose during the process.

Each team member worked on a chosen task from the beginning to the end of each sprint. This allowed many tasks to be completed in parallel, keeping their ownership well defined and understood, in what the practitioners call team self-organization. All information about allocated tasks was recorded in an open source project management application used by the team.

The daily meetings lasted around 10 minutes. The initial meetings involved two team members plus the SCRUM Master and PO. They were organized throughout the project, although these were less rigid in the last sprints. The PO, which was also the requirement analyst, participated in these daily meetings up to the middle of the project. After some time, the SCRUM Master decided that it was important for the team to have more autonomy in the decision-making process without the PO's presence.

Agile development based on SCRUM is a good practice. The SCRUM Master and PO had this belief from past experiences in previous projects in another company. They agreed that this kind of methodology leads to higher levels of team productivity, provides a better mechanism to control scope, changes and risks, and helps the team to focus on the business value of each delivery. Both of them recognized that the application of this methodology could foster a real customer involvement, which had a positive impact on the project. In this context, the PO said:

> *"With a strong communication and risk management, it is possible to improve the outcomes of requirement definition activity as well as the requirement traceability."*

However, the SCRUM approach is not clear on how to establish monitoring in development teams, so combining SCRUM with the practices of the CMMI model improved project monitoring practice. During a release planning meeting based on metrics, the PO commented:

> *"After using quantitative metrics to monitor sprint progress, I am sure that they can contribute to process improvement and a better team performance and scope control. The measures can lead to an effective defect causal analysis and bring more predictability to the work process."*

The PO and SCRUM Master agreed that a good automated software testing approach is essential to the success of any project. The origin of this belief of the PO was

related to the experience in project PGE-1. For her, the absence of this approach brought some negative impacts on project practices, such as:

- No time left in the sprint for bug correction;
- Poor test coverage with absence of real test cases;
- Low effectiveness of software testing;
- Focus on sprint deadline, not on software quality.

The belief in test automation of the SCRUM Master had its origin in past experiences in another company. He agreed with the PO and listed three more negative impacts:

- Low level of testing expertise of the project team;
- Low effectiveness of inspection code practice;
- Regression testing was demanding and time consuming.

About component reuse, the PO and SRUM Master considered it a good and very important practice. They have had successful past experiences in previous projects with the exchange of knowledge and experiences and agreed that reuse practice could lead to better results in terms of team productivity and product quality. The PO also highlighted that reuse practice could provide a good infrastructure for the company's future projects.

Thus, it was clear that common strong beliefs were reflected into practices that the project team effectively adopted. In addition, these beliefs were easily strengthened and transferred to other people involved in the project. Beliefs related to process management and job rotation were identified in the interview but were discarded after some observations, also the participants did not judge them relevant for discussion.

There were also conflicting beliefs between the SCRUM Master and the PO (see Figure 11). The SCRUM Master participated in a SCRUM certification course. He started the first sprint after the course with a meeting to discuss some SCRUM methodology issues, involving three main aspects: backlog changes during sprints, time schedule for bug correction, and the closeness between the team and the customer. The project team had considered some of these issues such as SCRUM gaps that could generate conflicting beliefs between team members. In the focus group session they had the opportunity to discuss these points.

The SCRUM Master believed that requirement changes during sprints could negatively impact the productivity of the team; on the other hand, the PO believed that there should be some flexibility on this, since the customer had high expectations about its requirements and needs. In the PO's point of view, it was very important to save extra time during sprints to fix bugs found by the customer in previous deliveries. And also, more closeness between the project team and the customer could impact positively the team commitment with the quality of the product delivered, and increase the sense of business value. For her, project deadlines were important, but without losing the focus on customer satisfaction.

About the closeness between the team and the customer, the SCRUM Master said:

"I think the team should be protected from direct contact with the customer, but must participate in the construction of the product backlog since the beginning."

With respect to story and task estimation, the PO advocated the use of the planning poker technique, while the SCRUM Master preferred the story points technique. Since the beginning of project PGE-1, the stories were estimated by hours and the PO was the responsible for this activity. The SCRUM Master disagreed with the PO, he believed that the team should be more autonomous to estimate task and stories. The SCRUM Master said that it was important to the team to know the story requirements very well and deeply, but this practice never happened in the project because the PO only elicited the requirements in a high level of abstraction. The main negative impacts of this conflict, from the perspective of the SCRUM Master, were:

- Tight time schedule for task implementation;
- Not enough time for software testing;
- Sprint planning not including slack time;
- Low level of estimation expertise of the team.

All these beliefs and impacts cited were discussed in a validation session in a focus group format and everybody agreed with the results showed in the mind maps and extracted from the interviews and observations.

Regarding the research questions posed for the first cycle of the research, we were able to answer them, at least partially. We investigated the origins and sources of team beliefs

embedded in the software development context to identify what practices were benefited or hindered by those beliefs.

With respect to research question RQ1.1–*"What are the origins and sources of Software Engineering beliefs?"*, beliefs were originated from two main sources: past and current project experience. All beliefs that emerged from personal hands-on experiences had a stronger influence on software development practices. These experiences were taken into account without much consideration for their original context or empirical results of their benefits. The attitude of a project team when adopting a new practice should be considered in the context of the influences arising from the practices in use at that time and the beliefs that came from their past experiences, which means some beliefs persist over time as they form the foundation of many of our intentions and practices. Thus, the decision of which practice to adopt was mostly based on the beliefs with respect to the person who is the main reference on the subject and the community's folklore, and the current software engineering community buzz around it.

This brings us to research questions RQ1.2 –*"How do these beliefs impact current software project practices?"* and RQ1.3 –*"What practices are being benefited and hindered by those beliefs?"*. Some beliefs indeed triggered the adoption of new practices. For example when the participants had a common belief, but did not act upon it. We named those semi-beliefs, things that people seem to believe, but do not have the predisposition to act upon them. Beliefs alone did not drive project teams to practice, it was also important to take into account other factors, such as aspects of organizational culture, to explain the consistency and inconsistency between belief and actual practice. Also, even when working on the same project, people can have conflicting beliefs and they definitely hinder the adoption of new practices. Thus, beliefs can be contradictory or compete for priority, and even the most salient beliefs can fall apart. Lastly, there were cases of common strong beliefs. Practices adopted based on common strong beliefs always yielded positive results and spread out among new practitioners.

The following is a table that presents the salient beliefs identified and characterized in Company 1 considering the seven projects investigated. Based on the variables represented in the conceptual framework designed for the first cycle (shown in Figure 7), we organized and tabulated the beliefs most frequently mentioned by the

participants; their respective class of beliefs; their origin in past or current experiences; the belief strength as semi-belief or strong belief; the conflicting part and the associated impact; and also the impact type as positive (+) or negative (-). Table 14 summarizes the main findings of the first cycle of the research.

Table 14 CYCLE 1 – Salient Beliefs of Company 1

Belief Class	**Origin**	**Strength**	**Organizational Aspect**	**Conflicting Part**	**Impact**	**Type**
Communication and Risk Management	past	strong	communication as a cultural value.	-----	better requirement definition and traceability. greater customer engagement.	+
SCRUM using quantitative metrics	past	strong	senior management pressure for quantitative results.	-----	better scope control. increased product quality. higher process predictability.	+
Project Management	past	strong	deadline pressure.	About the type of documentation required for project management.	poor product quality. higher rework. worse scope and cost mgmt.	-
Software Testing	past / current	semi	innovation as a goal.	-----	high level of testing expertise. higher effectiveness of testing. better product quality.	+
Component Reuse	past	strong	the company's start-up as a time for planning and structuring.	-----	higher team productivity. better product quality.	+
Agile Methodology	past / current	strong	IT market pressure.	About backlog flexibility and rigor on time schedule.	better change management. higher team productivity.	+
Task Estimation	current	semi	deadline pressure.	About the best task estimation technique and team autonomy.	low level of estimation expertise. negotiations on bad terms using tight time schedule.	-

The ethnographic approach adopted helped in capturing realistic scenarios and representing them in a mix of ideas and insights into narrative accounts and field information. During this first cycle of the study, we made use of techniques that proved themselves to be useful, like a combination of interviews, observations, and document analysis. No doubt, the use of an interview technique contributed to meet the challenges involved through the balance of the practices of participant observation and participant listening, and the provision of a better perception of the participants' stories and their connection to the observations.

3.1.5 Summary of Results

The first cycle of the research mainly focused on the issue of belief origins, sources and usage in a software organization. After reducing the data to a limited number of classes of beliefs, we characterized them in terms of origins and sources. This cycle of study showed that beliefs were originated from two main sources: past and current project experience. These experiences were formed by a series of events lived with direct personal participation or observation, which framed unconscious patterns of behavior through frequent repetition. All beliefs emerged from personal hands-on experiences that had a strong influence on software development practices. Based on the findings, it became clear that beliefs alone do not lead project teams to action and behavior. Other factors should be considered and further studied.

The research questions RQ1.1, RQ1.2 and RQ1.3 of the first cycle led to new knowledge and evidence that is important to the SE industry. At the end of this cycle, we obtained four important results:

- Identification and mapping of team beliefs origins, sources and contexts;
- Characterization of the influence of beliefs on software industry projects;
- Representation of *common* and *conflicting* beliefs and their impact on practices, processes and decisions in the software industrial context;
- Better understanding of how beliefs do influence software industrial settings.

After, approximately, one year of ethnographic research focusing on the social interactions, communications, and relationships that arise as an intrinsic part of adopting new software development practices, we evolved the conceptual framework aiming at providing a more suitable one to understand the influence of belief systems on team practices in organizational environments. Thus, for the second cycle of this research, we worked on the evolution of our initial conceptual framework into a more theoretical framework, based on our previous research and behavioral theories models, in order to help us to analyze the relevant experiences of the software team members, predict team behavior intention, and also

document the inconsistencies between declared beliefs and real practice in software projects. The case studies of the second research cycle are presented in Chapters 4 and 5.

Regarding methodological issues, the benefits of using a collaborative ethnographic approach, including a participatory action research strategy, showed that it is a better choice to support and collaborate with the participating companies, rather than just study them. However, applying collaborative ethnography to study software organizations is not a simple task. Several challenges were identified in this first cycle of our work. These challenges, and associated lessons learned are presented in Appendix D. They represent an important contribution of our research (Passos et al., 2012).

Chapter

4

This chapter describes how we conducted the first phase of the second cycle of the research. It starts by explaining why and how the TRA model was adopted from the results of the first cycle and the benefits of applying a behavioral theory in the context of software development practices. The chapter then describes the research methodology and process involved in this phase of the research, and presents its main results.

4 UNDERSTANDING TEAM BEHAVIOR INTENTIONS

The findings of cycle 1 of the research led to an evolution of our initial conceptual framework (see Figure 7), which was developed on the basis of our perceptions about the main aspects to be studied in a belief system, into a new framework based on the variables and relationships of the TRA model.

Based on the findings of cycle 1, it became clear that beliefs alone do not lead project teams to action and behavior. Factors such as attitude toward behavior have a significant influence on practice. Thus, we applied TRA to agile software project teams of the same company we had worked with during the first cycle of the research, focusing on the influence factors that actually impact on software practices in industrial settings.

The Theory of Reasoned Action (TRA) (Fishbein and Ajzen, 1975) seeks answers to the fundamental question of why people behave a certain way, from the study of attitude and behavior, and has received considerable attention within fields related to social behavior. TRA provides a suitable framework for conceptualizing human behavior and support the determination of behavioral intention of participants. Its theoretical framework is shown in Figure 5.

In the context of our research, TRA was helpful in generating rich and detailed accounts of software project teams, the interactions between their members, and, especially, the actions oriented toward certain software practices. It allowed us to study the way beliefs and attitudes are formed and their relation to behavior and practice.

4.1 CYCLE 2 – PHASE 1 – CASE STUDY APPLYING TRA

In this phase, we refined our goals and research questions based on the knowledge acquired in the real-life situations of cycle 1 and a significant literature review on behavioral theories. We also made changes to our research method and instruments and proposed an evolution of our initial conceptual framework.

4.1.1 Conceptual Framework

The conceptual framework idealized for the first phase of the second cycle is shown in Figure 12. As already mentioned, in the second cycle, we focused on the social interactions, actions and communications that arise as an intrinsic part of adopting new software development practices. In this scenario, the Theory of Reasoned Action caught our attention because it can help to predict people intention and behavior quite well, and provide a simple basis for understanding the relationship between beliefs, attitude and subjective norms.

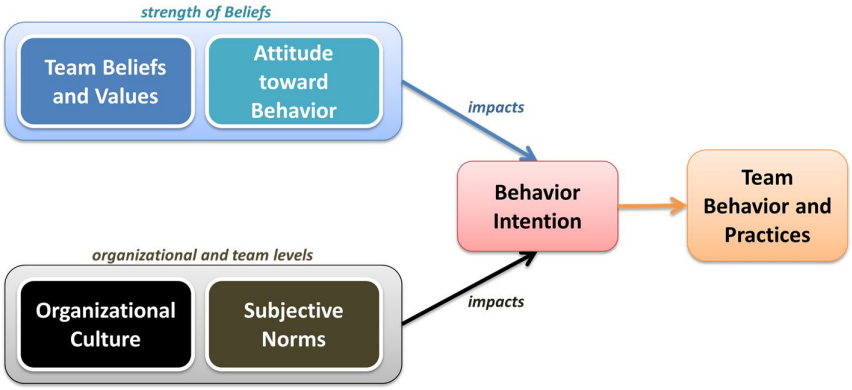

Figure 12 CYCLE 2 – Conceptual Framework (based on TRA)

As illustrated in Figure 12, the values and beliefs of a project team, in addition to the attitude toward its behavior, represent the strength of beliefs. It motivates people toward a behavior intention. So, a subject or matter that a project team believes has a great deal of weight and a significant impact on a team attitude, regardless of its validity. Taking in consideration that attitude is a predisposition to act in a positive or negative way toward an object; it exerts significant influence on the team's behavior intention, which, in turn, influences the team's practices. Another important factor, represented by subjective norms added to organizational culture, can also impact the team's behavior intention and, consequently, the team's practices. It is believed that people consider the implications of their actions and act based on a reasonable assessment of those implications. Therefore, the behavioral intention is affected by what others think and the strength of their opinion on the organization in context.

4.1.2 Research Goals and Questions

The main goal of the second cycle of this research was to better understand team behavior intention applying behavioral theories and addressing the influence factors, such as beliefs, attitude, organizational culture and values, and subjective norms, that actually impact on software practices in industrial settings. Our intention was to provide guidelines for

organizations, based on conclusions drawn from a case study applying TRA theory to study SE practices, recommending how to improve the work practices in a problem-solving paradigm. It was also a pertinent goal to contribute to a refined understanding on how to apply behavioral theories in SE contexts.

After the results obtained in the first cycle and the introduction of TRA in the research scope, new sub-research questions were framed to direct the first phase of this second cycle, as follows:

1. **RQ2-1.1**: How do past experiences influence attitude toward software practices in software development teams?
2. **RQ2-1.2**: How do organizational culture and subjective norms influence the adoption of new practices in software teams?
3. **RQ2-1.3**: How do common and conflicting beliefs impact positively or negatively on software practices?

We applied TRA to agile software project teams of the same company we had worked with during the first cycle. The projects were applying, for the first time, an agile methodology; so we focused on capturing origins, sources and impacts of beliefs on self-management development practices as a new practice in the company. In a 12-month period, we conducted focus group meetings and a set of interviews with professionals from different profiles and who were involved in the company's projects. This study confirmed origins, sources and impacts of beliefs on new development practices evidenced in the first cycle of the research. The results indicated a strong influence of past experiences or repeated behavior on team behavior intention. The results also showed that organizational support and culture were essential to achieve self-management team effectiveness.

4.1.3 Research Methods

During the first phase of cycle 2 of the research, our main strategy was to conduct a new exploratory case study in the same company studied in the first cycle (Company 1) to

confirm the previous findings. Basically, we adopted an interview-based qualitative data collection approach, performed focus group meetings with professionals involved in software projects, and drew maps of project context. Our conceptual framework (see Figure 12) were used to focus and bound the collection of data of phase 1, and guide the understanding of the influence of belief systems on team practices.

We conducted the case study applying TRA to software project teams during a period of 12 months. We started by performing an initial focus group session with seven practitioners directly involved in the main agile projects of Company 1. In order to define the interview questions, we opted to keep our interview-based qualitative approach based on the War Story technique (Lutters and Seaman, 2007), but we built a different script for this phase of the research (see Appendix C).

Data Collection

In Company 1, the data collection involved two rounds. The purpose of the first round was to capture the current state of the self-management team practice in the company. To achieve this goal, focus group meetings were held with practitioners involved in agile projects. Before the first meeting, a literature review was performed, which was designed to address the key references related to the subject and we also prepared an initial presentation to raise some issues we intended to investigate. This presentation included topics related to self-management team pillars, its main challenges and expected benefits. Thus, some relevant aspects were discussed during the following meetings, such as: team autonomy, shared leadership and collective decision-making, communication and collaboration, among others. The discussion helped to identify and map the current situation of self-management practice and the main challenges involved. The second round involved a set of interviews with all the participants of the focus group sessions to explore the research questions defined. Each person was interviewed on-site just after the meetings with 30 minutes of duration, on average.

After a literature review and an evaluation of the most relevant aspects of the self-management team practice, we defined a questionnaire and used it for the interviews. The purpose of this round was to obtain a deep understanding of the overall dynamics of processes and activities, comprising the main aspects of the software projects reality. At the same time, it was important to cast light on the respondents' past experiences, beliefs emerged or evolved from these experiences, impacts of new agile practices on projects and unexpected effects of

known and new methods or techniques. Afterwards, we used the interview findings to guide a validation session with the participants of the study in a focus group format. The purpose was to check if they would agree about the preliminary findings or invalidate them.

During the interviews, after two warm-up questions that asked what the participants knew about self-management team practice and if they had had any previous experience with self-management team practice, we continued with three questions about these past experiences related to self-management team practice. We asked them to cite these experiences, to assess the experiences as positive or negative, and to list the main challenges of implementing self-management team practice. The three lessons learned questions aimed to understand the basis of how useful these past experiences were to their current project and what situation or risk they had tried to prevent or mitigate, in the current project, taking in account past experiences related to shared leadership and collective decision-making practice. As reaction questions, we asked about the degree of autonomy of their agile teams, how communication and collaboration practice was and the cross-functional skills of the team members, and how the organizational support for self-management team practices was in the company. We also asked in which aspects they thought team members needed to improve to reach the self-management team benefits. Our intention was to assess the current state of the self-management team practice in the company and identify beliefs related to this practice that affected project results and outcomes. And finally, a metric-based question tried to address which project metrics could demonstrate the impact of the self-management team practices on project success.

We spent an average of five hours of transcription and one hour of preliminary coding for every 30 minutes recorded. The transcripts of the seven interviews produced more than 56 pages of text and an average of 15 codes per transcription. The audio of the validation session was not transcribed, but tagged with seven codes. In this process, we transformed qualitative data into belief classes as themes. We built about 15 high-level patterns of code that we translated into 6 main themes.

Table 15 CYCLE 2 – PHASE 1 – Case Study Data

Attribute	Data
Duration	12 months
Software Projects involved	3
Participants	7
Interviews	7
Interview Duration	30 minutes
Interview Transcription	56 pages of text
Coding Transcription	15 codes per transcription
Patterns of Code	15
Main Themes	6

The interviews were complemented by document analysis of project artifacts, such as: software development policies and procedures, project charter, project plan, risk plan, resource plan, project metrics, status reports, requirements specification, and software testing records, among others. We triangulated data from different sources to generate a more comprehensive understanding of the social environment and its subtleties in the context.

Data Analysis

We chose to maintain our data analysis approach based on thematic analysis and within-case analysis. We analyzed all the information collected via cycles of transcription and coding to identify relationships and patterns, and group summarizing segments of data into segments, codes and themes in order to create a quantitative description of the frequency of themes and form clusters of themes most frequently mentioned.

Figure 13 represents our coding process and evolution in a thematic analysis approach. We began with transcriptions, then we identified specific segments of text, tried out coding classes of beliefs on them, recognized relationships between beliefs and practices, then moved to translate this coded information into themes and trends, and finally, we tested our hunches and findings to integrate data into an explanatory framework derived from the conceptual framework initially designed.

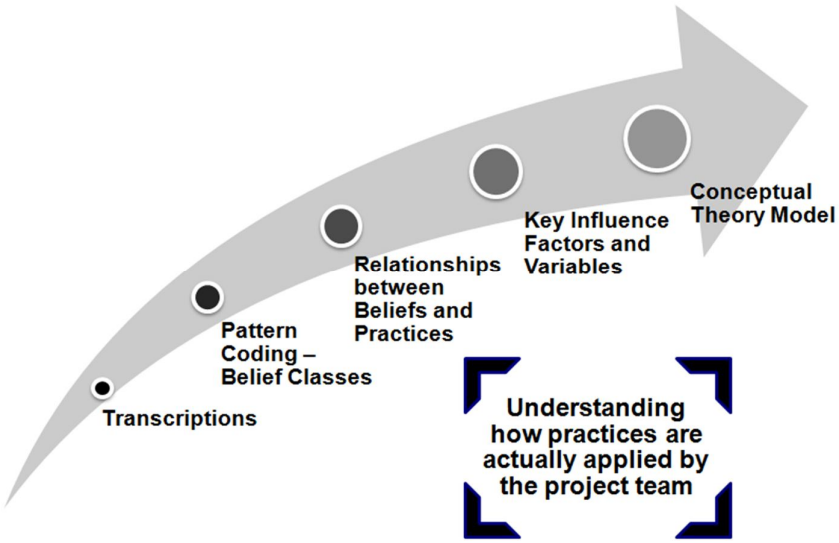

Figure 13 Thematic Analysis Evolution

The purpose was to obtain a deep understanding of the overall dynamics of processes and activities in projects context, focusing on the relevant activities, communications and incidents occurred in the past and current projects that might exert influence on the teams' behavior and confirm whether the hypothesized relationships in the theoretical framework were empirically true.

4.1.4 Case Study Context and Description

The first phase of cycle 2 of the research involved a case study set in the same software company studied in the first cycle, in which projects were constantly introducing new software practices. The company has one development center with ISO-9001 and MPS-Br certifications in the scope of agile software projects based on SCRUM[3] approach. We conducted the case study by applying TRA in the context of software development practices during a 12-month period. It involved a total of three projects and seven participants as

[3] SCRUM approach. http://www.scrum.org/

illustrated in Table 16. The profile information served as a context to better understand the points of view of each participant connected to the beliefs found.

Table 16 Participants Profile

Software Practitioners			
Company	***Project***	***Roles***	***Experience***
1	DMS	Product Owner + Developer	> 03 years
1	DMS	Scrum Master	> 10 years
1	Competency Center	Manager	> 15 years
1	SIIC	Product Owner + Developer	> 10 years
1	DMS	Product Owner	> 15 years
1	PGE-2	Scrum Master + Developer	> 05 years
1	DMS	Software Tester + Quality Assurance	> 03 years

After a literature review of the TRA model and an evaluation of the most relevant influence factors found up to this point of the research, we expanded the variables of TRA, using Montano and Taplin's (1991) work as a reference, in order to better elucidate the main aspects of the study, their key factors and the relationships among them. Figure 14 shows the expanded components of TRA considered in this first phase of the second cycle and the related research hypotheses (H1 to H7).

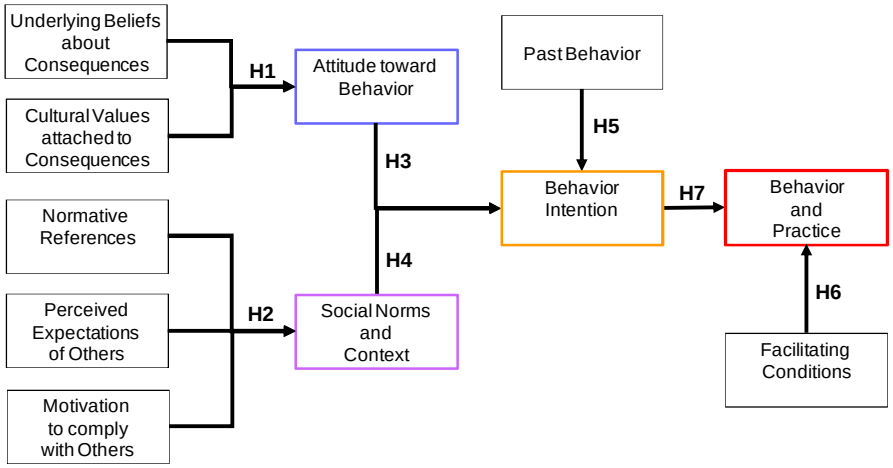

Figure 14 TRA Expanded Variables (based on Montano and Taplin, 1991)

The following table describes how we connected each variable under study with the interview question, research question (listed in Section 4.1.2), and hypothesis explored.

Table 17 TRA Variables Mapping

TRA Variable	Interview Question	RQ	Hypothesis
Underlying Beliefs	1. What do you know about self-management team practice? 4. In your point of view, was this experience positive or negative? Please comment.	RQ2-1.3	H1
Cultural Values	5. In your opinion, what are the main challenges to implement self-management team practice?	RQ2-1.3	H1
Normative References	13. In which aspects do you think team members need to improve to reach the self-management team benefits? 14. Which project metrics do you think self-management team practice could affect positively or negatively? Cite and explain.	RQ2-1.2	H2
Expectations of Others	8. What situation or risk have you tried to prevent or mitigate in the current project taking on account of experiences already lived with shared leadership and collective decision-making practices? Cite and comment.	RQ2-1.2	H2
Incentive to comply with Others	7. Could you tell me about some self-management practice adopted in past projects and not adopted in your current project? Cite and explain. 11. Are team members prepared with cross-functional skills to cover each other in a self-management team?	RQ2-1.2	H2
Attitude toward Behavior	6. What good practices related to your expertise and experience in self-management teams were useful to apply to your current project? Please, tell me about their application. (successful or not)	RQ2-1.1	H3
Social Norms and Context	12. How is the organizational support for self-management team practices in the company?	RQ2-1.2	H4
Past Behavior	2. Have you ever worked within a self-management team before? 3. Could you cite a past experience related to self-management team practice?	RQ2-1.1	H5
Facilitating Conditions	9. How is the degree of autonomy in Scrum teams? 10. How is communication & collaboration practice in Scrum projects?	----	H6
Behavior Intention	all	all	H7

Accordingly, the hypotheses investigated are presented below:

- **H1:**Underlying beliefs and cultural values have an effect on attitude toward behavior;
- **H2:** Normative references, expectations of others and incentive to comply with others increase the strength of social norms;
- **H3:** Attitude toward behavior increases the intention of adoption of practices;
- **H4:** Social norms and context increase the intention of adoption of practices;
- **H5:** Past experiences with a behavior have an effect on the intention of adoption of practices;
- **H6:** Facilitating conditions are direct determinants of adoption of practices;
- **H7:** Behavior intention increases the chance of adoption of practices.

Our intention was to address the participants' past experiences, underlying beliefs and cultural values, normative references and pressures embedded in the software development context, attitude toward new software practices, and facilitating and disturbing conditions related to impacts evidenced in their projects in order to identify the most influential factors in the organizational context and document the inconsistencies between declared beliefs and actual practice. We focused on the aspects mentioned more than once and the top-of-mind beliefs of the respondents.

Results from Company 1

We drew descriptive conclusions about the phenomena in Company 1 and summarized the case study results into a more comprehensive understanding of team behavior intention, guided by the conceptual framework designed for the first phase of cycle 2 (illustrated in Figure 12), the link between the expanded variables of TRA and the related research hypotheses (shown in Table 17) and the research questions posed. The findings from the within-case analysis are presented below, providing a description for key influence factors and their attributes, and including quotations and evidence that support the findings.

The study pointed out the participant's past experience as the most influential factor to predict team behavior. The project team members under study have tried to apply self-management practices in their agile software projects. According to them, the main challenges were the high individual autonomy and the low maturity of the team members. In past companies, developers were used to only worrying about their own schedule and task implementation, which resulted in even more individual autonomy. Most practitioners had experienced self-management practices before in other companies, but never in the context of a team, only individually. The SCRUM Master of the DMS project commented:

> *"I have noticed that usually team members don't put their focus on the delivery but on their individual task. However, there is no meaning in getting your task done, if all others are not ready to deliver to the customer."*

For three of the seven interviewees, the lack of productivity and quality metrics and boundaries made task estimation and the formal definition of *product done* imprecise, which led to a lower quality of the product delivered to the customer due to deadline pressure. Moreover, they reported that imprecise task estimation hindered team autonomy in their

previous projects, which led to negotiations on bad terms with their customers and, consequently, low team-level effectiveness (Team Autonomy Belief). The Competence Center Manager presented his point of view:

> *"The formal definition of product done still needs to be internalized by the project team. Thereby, the team will not give in to the customer pressure to deliver the product until it is actually finished."*

In addition, the interviewees seemed to have a common concern and interest in better defining the work process with the adoption of new metrics for team productivity and software quality (Process Belief), but they did not act upon this issue. Even though they knew that imprecise task estimation could put deadline negotiations at risk with project customers.

One member of the SIIC project believed that the absence of well-defined roles in the project could hinder the project progress (Project Management Belief). For her, the roles defined in agile methodologies, such as SCRUM Master, must be performed by someone trained and aware of his/her responsibilities in the project. Otherwise, the project will not be properly conducted and, as she had experienced before, bad project management would contribute to the failure of the software project.

Some members of the DMS project team believed that through the benefits of team commitment it was possible to be more productive as a self-management team (Commitment Belief). For others, the response time was not related to a self-management team attitude. It was clear that the team members' past experiences were influencing their beliefs and attitude toward agile practices in different ways. This team also believed that the low level of team member seniority could disturb project progress and delay the project (Maturity Belief). For them, this fact was obvious from the very beginning of the DMS project, six months previously.

The study also highlighted the influence factors associated with organizational culture and subjective norms to predict behavior intention. Five of the seven interviewees declared that organizational support and culture were essential to achieve a good team performance. Most of the interviewed practitioners agreed that there was a high level of organizational support for self-management practices in Company 1. In addition, two of them reported that team members with old culture values, drawn from previous projects in another

company, found it difficult to adapt to self-management team practices. The Product Owner (PO) of the DMS project declared:

> *"In this company we have a high level of organizational support and culture for self-management practices. The board of directors gives us freedom to conduct the projects, so they believe in self-managing teams."*

All respondents agreed that some of the key self-management practices were being implemented in the company, but there was still a lot to consider before the project teams could be seen as self-organized. For them, it was important to progress using team-level autonomy, focusing on a holistic project view by preventing each team member from caring only about his/her task and evolving as a team to achieve greater maturity. Thus, the evidence suggested that a good fit between the organizational culture and subjective norms, and the software development context leads to an easier adoption process.

Evidence also showed that there was a significant connection between a team belief system and organizational culture. Our findings indicated that the attitude toward an object was based on how favorable the total set was, because one considers each belief about that object and its evaluation according to the organizational and project context. For example, most members of the DMS project team agreed that achieving team-level effectiveness through team commitment was a good idea. Commitment was identified as a strong value for Company 1 and for the DMS customer because this project involved the obeying of laws and their respective deadlines. On this subject, the Software Tester remarked:

> *"I think when everyone commits, the delivery happens. In general, the problems are found and solved quickly and we build small deliverable pieces faster."*

Regarding the Team Autonomy Belief, the interviewees seemed to have a common concern and interest in better defining the work process with the adoption of new metrics for project monitoring. However, nothing happened, although they were aware that imprecise task estimation could disturb team autonomy and jeopardize negotiations with project customers, which led to low team-level effectiveness. Another point is related to the Flat Organization Belief. It is a common belief for four of the seven participants, however, no action was planned to reduce the hierarchical distance between teams and senior management of the company. There was no consistent initiative, besides some isolated training, to improve

the team members' maturity and experience level. All these cases have illustrated situations where we identified common semi-beliefs.

The evidence showed that there were cases of common strong beliefs. Apart from team communication, whose practice was considered effective in all software projects investigated, there was the case of job rotation in the DMS project team (Job Rotation Belief). The entire team reported that this practice improved software quality in the current project so much that they decided to have functional redundancy in the team for upcoming projects. Also, there was the Commitment Belief. As mentioned, commitment was identified as a strong value for the company and it was a present value translated into many practices from the very beginning of the DMS project. Thus, all these cases have represented beliefs that the project teams actively acted on, and with positive results.

There were also conflicting beliefs between the stakeholders involved in the software projects. On the DMS project, only one member of the team believed that self-management team practices, by themselves, could improve software quality (Team Autonomy Belief). The other members thought that agile practices were not enough to ensure software quality, but it was necessary to combine them with the adoption of quality metrics and boundaries linked to the formal definition of *product done* (Team Autonomy Belief). The Scrum Master commented:

> *"For a self-management team to ensure the quality of the final product, everything should be well defined, such as: software test metrics, process quality metrics, and so on."*

On the other hand, the same member of the DMS project disagreed with the other three members about the impact of self-management practices on team productivity. For him, the response time was faster due to the Commitment Belief shared by the whole team and it was not related to a self-management team attitude. However, for all the other team members this level of commitment was only achieved with the practice of a self-management team. In this scenario, these conflicting beliefs indeed hindered the adoption of new practices in the DMS project, such as shared leadership and collective decision-making, which contributed to the decrease in the level of team autonomy.

Tabulating the Results of Company 1

In Table 18, we list the respective class of salient beliefs, according to topics related to the self-managed team practice; their origin in past or current experiences; the belief strength as semi-beliefs or strong beliefs; the conflicting part and the impact produced by each belief; and lastly, the impact type.

Table 18 CYCLE 2 – Salient Beliefs of Company 1

Belief Class	*Belief*	*Origin*	*Strength*	*Conflicting Parts*	*Impact*	*Impact Type*
Process	The lack of productivity and quality metrics makes task estimation and the formal definition of "product done" imprecise and jeopardizes deadline negotiations with project customers.	past	semi	----	Low quality	-
Job Rotation	The functional redundancy practice improves software quality.	current	strong	----	Better quality	+
Commitment	To achieve team-level effectiveness through team commitment is a good practice.	past	strong	about the impact of self-management practices on team productivity.	Better productivity	+
Team Autonomy	Imprecise task estimation disturbs team autonomy and reduces performance.	current	semi	about the impact of self-management practices on software quality.	Low team-level effectiveness	-
Maturity	The low level of team member seniority disturbs project progress and can delay the project.	past	semi	----	Slow project progress	-
Flat Organization	big hierarchical distance between the team and senior management hindered team autonomy and reduces performance.	current	semi	----	Low team-level effectiveness	-

Regarding hypothesis H1 linked to research question RQ2-1.3 –*"How do common and conflicting beliefs impact positively or negatively on software practices?"*, we investigated the origins, sources and contexts of team beliefs embedded in the organizational context to confirm that conflicting and semi-beliefs indeed hinder the adoption of new practices, and common strong beliefs have a positive effect on attitude toward behavior. The Flat Organization Belief, Maturity Belief and Team Autonomy Belief were examples of common semi-beliefs, representing something that people believe, but do not demonstrate the attitude toward the behavior or practice. The impacts of these beliefs on SE practices are always negative, decreasing the intention of implementation of key practices. Also, some conflicting beliefs hampered the adoption of new agile practices. This type of situation broke the unity of the project team and reduced its performance. Some of these conflicts were subtle and our approach of formally capturing and representing team beliefs definitely helped to identify and point out those conflicts between the involved stakeholders. Also, there were cases of common strong beliefs as present values reflected into many practices that the project teams actively act upon and always with positive results, for example the situations related to

the Commitment Belief and Job Rotation Belief. These beliefs are usually implemented since the very beginning of the project, based on the confidence that these particular practices will lead to positive consequences.

With respect to H2 linked to RQ2-1.2–*"How do organizational culture and subjective norms influence the adoption of new practices in software teams?"*, we identified the project stakeholders (customer contacts and other team members) and the context around the project as potential sources of influence on team decisions. The normative references of the project context and perceived expectation of important others added to how strongly the team member is motivated to comply with the expectation can place in evidence the social norm in question, especially if the norm fits into the context of team beliefs and values. Thus, hypothesis H2 is supported by evidence related to events in which there was an easier assimilation of a new practice or behavior. In the same line, hypothesis H4 related to RQ2-1.2 was accepted. The stronger the social norms are, the more they become determinants of behavioral intention. The interviewed practitioners declared that organizational support and culture were essential to achieve a good team performance in terms of software practice. In this scenario, social norms are considered as indirect predictors since they are considered to influence behavior through their effect on behavior intention.

Hypothesis H3 linked to RQ2-1.1 –*"How do past experiences influence attitude toward software practices in software development teams?"* investigates team attitude influenced by past experiences. Whereas each person is able to process information in a systematic manner forming opinions and comprehension, attitude is essentially information people have obtained about someone or something, that will form an opinion or predisposition to act. It means people behave based on past events and experiences, and also their attitude toward the given act. As well as social norms, attitude was found to be a significant indirect predictor of behavior through the impact on intention. Most interviewees had experienced self-management practices before in other companies and had previous information and opinion about this practice. Very likely, they had expectations about the results of the practice and some idea of the value associated with these results.

When considering hypothesis H5,somehow linked to research question RQ2-1.1 – *"How do past experiences influence attitude toward software practices in software development teams?"*, an interaction between repeated behavior and intention was found such

that practitioners with large experience in using metrics for project monitoring were more likely to carry out their intentions of using them to foster team autonomy. So, we confirmed that satisfactory past experiences with a behavior increases one's intention to repeat the same course of action. Also, the absence of well-defined roles in agile projects was identified as a critical issue based on previous experience related to project management using SCRUM methodology. Thus, on more than one occasion, past experiences proved to be a strong influential factor and predictor of behavior intention, rendering support for hypothesis H5.

Even with little evidence we can presume that hypothesis H6 was accepted. Some facilitating conditions were mapped as characteristics of the project teams and contexts that make it easier or more difficult for an individual to carry out his/her intention. For example, the low level of training and maturity of the project teams can delay the progress of the project and impact, negatively, the project results. Both can be seen as factors that influence the likelihood of completing a task or practice with success. Thereby, facilitating conditions are additional determinants of probable behavior.

According to expectations, behavior intention significantly increases the chance of adoption of practices, supporting hypothesis H7. Behavioral intention reflects how hard a person is willing to try, and how susceptible he/she is to engage in a given behavior. The progression from intention to practice in organizational environments is a consequence of all the key influence factors and their relationships reviewed so far. These factors can cause people to go against their initial intention, sometimes when the conditions are unfavorable. However, people usually do what they intend to do, confirming that, in theory and practice, intention is the most proximate predictor of behavior. In this case study, we perceived behavior intention as a plan of behavior in response to specific situations elaborated in terms of how, when, where and other specifics. Thus, we sought to provide descriptions of relevant events, facts and organizational scenarios, including verbatim quotations with sufficient evidence to support the findings and allow a proper perspective of the context of the people under study.

In summary, this case study highlighted the strong influence of past experiences on self-management practices of agile teams in the software organization (Company 1) and produced new insights about repeated behavior in discussing the role of past experiences in the context of SE theory and practice. The study also confirmed that beliefs alone do not lead

project teams to practice. Other relevant influence factors, represented by the expanded variables of TRA illustrated in Figure 14, should be considered and deeply investigated.

4.1.5 Summary of Results

In this first phase of cycle 2, the central point of the research was the application of TRA in the context of software development practices in order to produce new insights into team intention and behavior. Thus, we developed a case study to characterize team belief systems by applying the TRA theory to investigate the origins, sources, and strength of these belief systems, and point out their influence on team software practice in industrial settings.

The research questions of the first phase of cycle 2 led to findings that confirmed a significant connection between organizational culture and subjective norms and project team behavior intention as the TRA model indicates. Evidence to support this was found in the statements declared by the participants about how much organizational support and culture were essential to achieve a good team performance and how much other cultural influences, resulting from previous projects, could set back or advance the adoption of software practices.

Thus, at the end of the first phase of cycle 2, we obtained the following results:

- Characterization of team belief systems and their impacts on the adoption of new software development practices, providing rich narrative accounts for software engineering research.
- Guidelines to help companies to improve project team practices through the knowledge of team belief system and its impacts.
- Practical knowledge on how to apply behavioral theories to study software practices, enabling further examination through confirmatory studies.

Overall, the results of this case study ratified and explained the influence of team belief systems on software practices. However, during the use of the TRA framework, it became clear that team behavior is not entirely under an individual's volitional control. The

main aspects to be studied in SE contexts should go beyond the influence of team beliefs on project team practices. A conceptual framework should also cover factors such as perceived behavioral control. The TPB model was then adopted as a more complete framework, for dealing with the complexities of the influence of a belief system on project team practices, and for conceptualizing human behavior in organizational contexts. The next chapter presents the description of the case study conducted by applying TPB in three different software companies.

Chapter

5

This chapter describes how we conducted the second phase of the second cycle of the research. It provides a description of beliefs and the key influence factors, including quotations and evidence that support the findings. A summary of the results of the second cycle are discussed. This chapter also presents the research approach and design, including the conceptual framework, questions and goals, methods and instruments used in this phase of the research.

5 MAPPING THE INFLUENCE FACTORS ASSOCIATED TO TEAM BELIEFS AND PRACTICES

Based on the findings of the last phase, it became clear that belief, attitude and subjective norm alone do not completely predict intention and behavior in software industry contexts. Other factors such as team perceived autonomy and confidence should be considered. Thus, we applied TPB to software project teams in three different companies.

The Theory of Planned Behavior (TPB) (Ajzen, 1985) fits in the context of our research within the software industry, because it allows us to study the way belief, attitude, culture and formed sense of self-efficacy, and their relationships with intention and behavior. TPB is expected to perform better than TRA in organizational contexts.

5.1 CYCLE 2 – PHASE 2 – CASE STUDY APPLYING TPB

5.1.1 Conceptual Framework

The conceptual framework for the second phase of cycle 2 was derived from an adaptation of the TPB model and is shown in Figure 15. It goes beyond beliefs, attitudes and social norms to consider team confidence and team autonomy influence on intention and behavior.

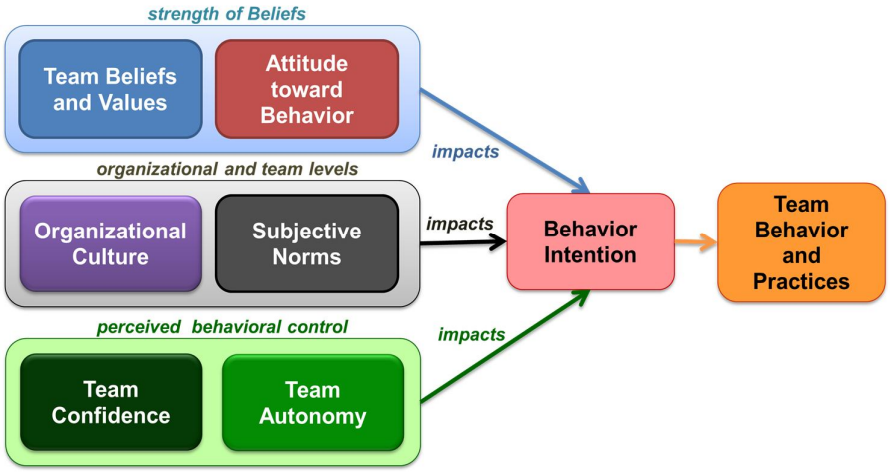

Figure 15 CYCLE 2 – Conceptual Framework (based on TPB)

Two new elements were added to the framework. The element of *Team Confidence*, that denotes people's perception of the degree to which they are capable of performing a given behavior, and the element of *Team Autonomy*, that indicates how much control of the situation people perceive. Both elements compose the *perceived behavioral control* component as a measure proposed by TPB. These elements interact with behavioral intention to affect behavior.

5.1.2 Research Goals and Questions

The main goal of the second phase of cycle 2 of this research was to understand and map new influence factors associated to team beliefs and practices, such as team confidence and autonomy, impacting on software practices in industrial settings. Also, we expected to provide support to others who want to apply behavioral theories to study software practice and valuable guidelines for managers and researchers as they seek to understand the main influences and the role of organizational culture on the adoption of software development practices.

The sub-research questions were adapted to the adoption of TPB model, as an extension of TRA, by including a measure of perceived behavioral control. They evolved as follows to meet the requirements of this theory and represent some of the findings the first phase.

1. **RQ2-2.1**: How do beliefs and attitude influence team practices in software organizations?
2. **RQ2-2.2**: How do organizational culture and subjective norms influence software team behavior and practices?
3. **RQ2-2.3**: How do team autonomy and confidence impact software practices?

We applied the TPB model to software project teams in three different software companies in order to better understand and analyze a belief system of software project teams under a more practical perspective, and use it as a base to foster the improvement of software development practices. It involved a set of interviews and information gathering about the context of the projects for six months.

We observed that organizational culture influence reinforced the strength of the team beliefs and the confidence that a particular behavior will lead to positive consequences. Thus, for this second phase of cycle 2, we emphasized the research question related to the organizational culture and normative pressure embedded in the software development context to gain new insights into expressive relationships between culture and practice in terms of the

three levels of culture (basic underlying assumptions, espoused beliefs and values, and artifacts).

5.1.3 Research Methods

The second phase of cycle 2 of the research had the same qualitative approach using case study as the central research method. Our main strategy was to develop an exploratory case study, adding new settings to the research, to confirm and delve deeper into previous findings. We adopted an interview-based qualitative data collection approach, asked insightful questions, drew maps of project context, and collected some project artifacts. Our conceptual framework was used to focus and bound the collection of data of phase 2 (see Figure 15), and expand the explanation about the main influence factors associated to project team beliefs and practices.

Data Collection and Analysis

We conducted the new case study applying TPB in three different software companies with very distinct profiles for six months. For each company (Companies 2, 3 and 4), we started by examining and mapping about 35 context facets using the same checklist of the first cycle to identify and capture the company and project contexts. We kept the same interview approach, but we had a new script for this second phase of cycle 2 (see Appendix C).

We performed nine interviews in total. The main target was to map and analyze the relevant experiences of the software team members and uncover the beliefs which could hinder or benefit the adoption of new software practices. Using TPB as a guide, we prepared interview questions to capture the issues we intended to investigate. We focused on the influence of organizational culture, subjective norms, team confidence and perceived autonomy embedded in the software development context.

We performed only one warm-up question that asked about the main challenge of the participants' projects in terms of productivity, quality, deadline, cost or other metrics. The past experience question investigated some bad or good experiences of the participants related to projects without well-defined software development methodology, risk analysis, planning or defined schedule, and how this scenario impacted the project results. The lessons learned questions aimed to investigate the process of adoption of new practices in the companies. So, we asked the participants how they perceived the adoption of new practices in terms of usefulness and importance, how they introduced their own new practices, originated from their past experiences, in their current projects, and how the acceptance and results of this introduction was. We also asked about mandatory practices, i.e., practices that they adopted just because it was required to use it in the organization, but they did not recognize its usefulness. We intended to understand why they did not believe in these mandatory practices. And lastly, we tried to capture the current state of the companies' software processes, asking about the recent introduction of new software practices and in which aspects these practices impacted the projects.

The interviews were conducted with professionals involved in the main software projects of the companies. Each semi-structured interview was performed on-site lasting for 15-40 minutes. Each person was interviewed once, shortly after a meeting with the researcher responsible for the interviews. Unfortunately, we did not have a chance to conduct validation sessions after the interviews.

We spent an average of five hours of transcription and one hour of preliminary coding for every 30 minutes recorded. The transcripts of the nine interviews produced more than 44 pages of text and an average of seven codes per transcription. In this process, we transformed qualitative data into belief classes as themes. We built about 10 high-level patterns of code that we translated into 4 main themes.

Table 19 CYCLE 2 – PHASE 2 – Case Study Data

Attribute	Data
Duration	6 months
Software Projects involved	5
Context Mapping	+35 facets
Participants	9
Interviews	9
Interview Duration	15-40 minutes
Interview Transcription	44 pages of text
Coding Transcription	7 codes per transcription
Patterns of Code	10
Main Themes	4

In all three companies, the interviews were complemented by document analysis of project artifacts, the interviews were compared with the collected information about the organizational contexts and documents. The resulting nine interview audios were analyzed using the thematic analysis approach, in order to crosscheck and compare the answers to find behavioral confirmation and disconfirmation as well.

5.1.4 Case Study Context and Description

As shown in Table 20, the second phase of the second cycle involved a set of interviews and information gathering about the project contexts in three different software companies.

Table 20 Companies Profile

Company	Age	Operation	Office	Segment	Personnel	Standard Process	SW Process Certification
2	11 years	Brazil	3	Full IT Service Provider	800	Agile Methodology	ISO-9001 CMMI Level 2
3	03 years	Brazil	1	Software Product Line	15	*not defined*	---
4	19 years	Global	>200	Full IT Service Provider	42000	CMMI Process	ISO-9001 CMMI Level 5

Each organization has at least one development center, two of them with ISO-9001 and CMMI certifications. In Company 2, the project team follows a well-defined process for software development and they recognize the benefits of using productivity rate as a metric for project monitoring. Company 3 is the newest and has started to develop the FD project whose sponsor is a Brazilian government fund. The FD project involves a web portal that brings together service delivery to facilitate the routine of busy people who are looking for convenience. The FD team members did not follow an agile process, but they decided to adopt some of the agile key concepts, such as the role of Scrum Master. And finally, Company 4 is a global technology, consultancy and innovation company, providing solutions and services in more than 120 countries. Based on a combination of electronics, communications and information technology, Company 4 follows a business strategy including solutions and services of high technological content, from the design of a solution, through to its development and implementation, until its operating management.

For each company, we started by examining and mapping the project contexts, then the interviews were conducted and supplemented by document analysis of project artifacts. The study lasted for six months. It involved a total of five projects and nine participants with distinct profiles as illustrated in Table 21.

Table 21 Participants Profile

Software Practitioners			
Company	***Project***	***Roles***	***Experience***
2	STF	Development Center Manager	> 10 years
2	STF	Programming Leader	> 03 years
2	STF	Technical Leader	> 03 years
3	FD	Scrum Master + Developer	> 03 years
3	FD	Developer	02 years
3	FD	Developer	01 year
4	SIG	Project Manager	> 03 years
4	SIGEP	Project Manager	> 03 years
4	DO	Quality Manager	01 year

We applied the TPB model to the five project teams in order to understand and analyze their belief systems. The instruments were adapted to the adoption of TPB model, as an extension of TRA, and the conceptual framework designed for the first phase of the second cycle was evolved to meet the requirements of this new theory (see Figure 15). We expanded the variables of TPB and related them to nine research hypotheses as illustrated in Figure 16.

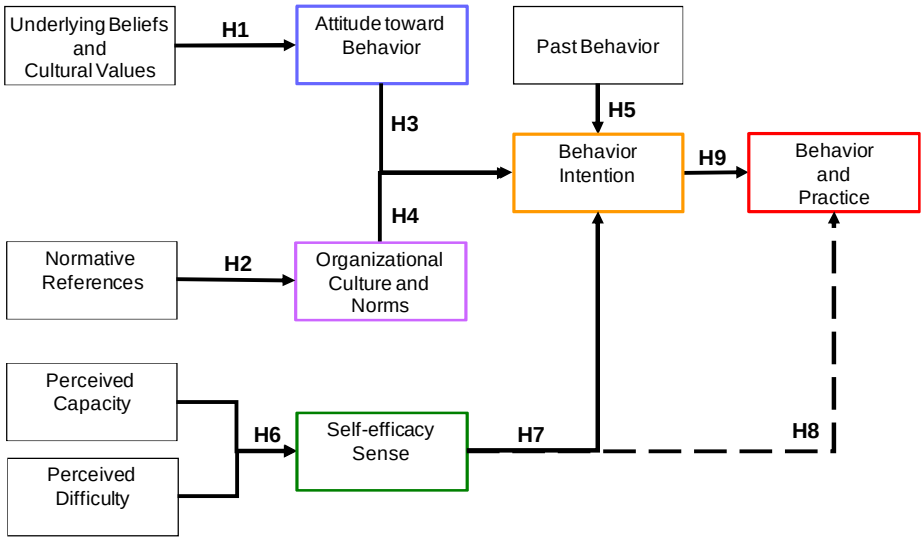

Figure 16 TPB Expanded Variables

Table 22 describes how we linked each variable with the interview question, the research question of the second phase of this second cycle, and the hypothesis investigated.

Table 22 TPB Variables Mapping

TPB Variable	Interview Question	RQ	Hypothesis
Underlying Beliefs and Cultural Values	1. In your opinion, what is the main challenge of your project? (productivity, quality, deadline, cost or other)	RQ2-2.1	H1
Normative References	4. Is there any practice in your current project (or past projects) that you adopted just because it was required to use it in the organization that you work, but you don't see its usefulness? Why do you not believe in this practice?	RQ2-2.2	H2
Attitude toward Behavior	3. Is there any practice in your current project that is new to you? Do you think this new practice is beneficial to your project? Please, explain in what sense. And after using this new practice for a while, have you changed your opinion regarding its usefulness and importance?	RQ2-2.1	H3
Organizational Culture and Norms	6. Is there any new methodology that the organization in which you work is adopting? Do you know the reason for that change? Do you believe in this new methodology? Why? In what aspects is this company's new methodology affecting your current project?	RQ2-2.2	H4
Past Behavior	2. Could you cite a past experience where the project was conducted without planning, a defined schedule, without following a software development methodology and without risk analysis, and how this impacted (positively or negatively) the project results?	RQ2-2.1	H5
Perceived Capacity and Difficulty	4. Is there any practice in your current project (or past projects) that you adopted just because it was required to use it in the organization that you work, but you don't see its usefulness? Why do you not believe in this practice?	RQ2-2.3	H6
Self-efficacy Sense	5. Is there any practice that you have introduced to your current project and that was not used in the organization in which you work? why did you think it was important to introduce this practice? And was there any problem with acceptance? What are the results of this practice in your current project?	RQ2-2.3	H7, H8
Behavior Intention	all	all	H9

Subsequently, the hypotheses investigated are listed as follow:

- **H1:**Underlying beliefs and cultural values have an effect on attitude toward behavior;
- **H2:** Normative references increase the strength of organizational culture and norms;
- **H3:**Attitude toward behavior increases the intention of adoption of practices;
- **H4:**Organizational culture and norms increase the intention of adoption of practices;
- **H5:**Past experiences with a behavior have an effect on the intention of adoption of practices;
- **H6:**Perceived capacity and difficulty has an effect on self-efficacy sense;
- **H7:**Self-efficacy sense has an effect on the intention of adoption of practices;

- **H8:**Self-efficacy sense is a direct determinant of adoption of practices;
- **H9:** Behavior intention increases the chance of adoption of practices.

We summarized the case study results into a more pragmatic understanding of belief systems behind SE practice and how people progress from intention to practice in organizational environments. Our purpose was to develop a new exploratory case study, adding three settings to the research, to confirm and delve deeper into previous findings, so the study sought to cover the key influence factors that actually impact on software practices in industrial settings.

In the conceptual framework designed for the second phase of cycle 2, illustrated in Figure 15, the link between the expanded variables of TPB, the related research hypotheses and the research questions were used to guide the data analysis presented below.

Results from Company 2

In Company 2, people were motivated by the IT market demands to adopt an agile methodology and the Project Leaders declared explicit empathy and claimed to be friendly to this kind of methodology. The Technical Leader commented:

> *"The programming part of the project was divided into three sprints, which was interesting because it allowed us to increase the application into small deliverable pieces. We were able to use the SCRUM burn-down strategy and monitor daily what was happening in the project. It improved communication between team members and made everyone walk together as a team indeed."*

At the beginning of the project, the STF project team showed some resistance to this adoption, but during the project, the team accepted the new practices and started to work very well as a team. At first, they seemed to be afraid of taking a bigger responsibility as a self-management team, since this approach presumes that the team have significant authority and responsibility for many aspects of their work, such as planning, scheduling, delegating, and making decisions. On this subject, the Programming Leader remarked:

> *"We tried to deploy some practices of agile methodology in our project, such as: daily meetings and staff involvement in determining their own activities and deadlines. At first, we had some difficulty in this process due to the team size and*

the long-term deadlines. So, the project team (including me) showed some resistance to adopt agile practices but, afterwards, we accepted the new practices and gained a different and better view about them."

Also, the company's development center manager had an attitude toward the use of an individual productivity rate as a metric for project monitoring. For him, a more precise task estimation had improved team autonomy in his projects, which led to better negotiations with customers and, consequently, higher team-level effectiveness.

Regarding the requirement traceability control in Company 2, it was mandatory to use a bad format and useless artifact for requirements traceability. The Leaders believed that this practice reduces configuration and change management effectiveness. However, it was part of the CMMI certification program, so it was required for every project in the company.

The CMMI certification program of Company 2 culture influenced the adoption of software engineering metrics for project monitoring. The company's Development Center Manager believed that a better task estimation using individual productivity rate could lead to project success. He said:

"As a manager, I felt the need to know my developers' productivity. So, we have set up this practice to use individual productivity as a metric for project monitoring."

The STF project team believed that the adoption of the CMMI practices increased the company's competitive edge and stimulated the company's evolution process bringing more profitable and demanding projects. Nevertheless, some members of the project team reported that the CMMI organizational culture seemed to frame software practices to a more traditional development process in a cascade model, which is not so compatible with agile methodologies. The Programming Leader showed his point of view:

"The CMMI organizational culture frames practices in a traditional development process using the cascade model, which fits well in a Development Center organization, but not so much for agile development models."

The STF project team considered that the compliance with the CMMI practices could increase the chance of success in software projects. The knowledge sharing through software documentation and planning, driven by the CMMI certification program, also increased the chance of success in software projects. For this team, a defined development

process and the existence of adequate planning could contribute to project goals achievement. So, the team confidence level related to this subject was substantially high. The Programming Leader commented:

> *"In the context of the CMMI certification program, we are always focused on knowledge management, which is today one of the main resources that a Development Center has to improve software quality and increase team productivity."*

The same applied to the adoption of agile methodologies for project management. The team had a high level of confidence and also autonomy to use the SCRUM model and its practices to conduct the STF project. The Technical Leader was explicitly sympathetic toward agile methodologies. From his perspective, the main positive impacts were:

- Better product quality;
- Better scope management;
- Better communication within the project team;
- Higher team-level effectiveness and integration.

Results from Company 3

Our findings indicated that the attitude toward an object is based on how favorable the total set of options is, because one considers each belief about that object and its evaluation according to the project context. For example, most members of the FD project team agreed that achieving knowledge sharing through a project tracking blog was a good practice. For them, this practice improved team productivity and integration, which, in turn, led to team-level effectiveness. The Scrum Master of the FD project said:

> *"We are adopting a blog tool to register the activities and events of the project. Thus, any new team member can readily understand the project evolution from an overview of the project knowledge and context."*

In addition, the Scrum Master of the FD project was motivated to adopt an agile methodology to speed up software processing and software product delivery. He believed that a good task estimation practice should be supported by appropriate tools, even though some team members were still resistant to this practice.

The evidence suggested that a good fit between the organizational culture and subjective norms embedded in the software development context leads to an easier assimilation of a new practice or behavior. Company 3 seemed to be motivated by competition to adopt an agile methodology. This methodology became part of its organizational culture and exerted huge influence on the behavior of the employees and how they acted toward an agile practice. The main positive impacts of this adoption from the perspective of the Developers of the FD project were:

- Higher team-level effectiveness and integration;
- Better project response time;
- Better team productivity.

The Scrum Master of the FD project was quite confident that the adoption of a task estimation practice supported by tools could help in the implementation of agile software methodology. He believed that it was a big challenge to predict team productivity without a systematic process for scheduling and task estimation. He said:

> *"The systematic recording of project events and activities using a tool is essential. So, everyone can have an idea of what is going on in the project. It is a real challenge for the project management to take actions to improve productivity and optimize cost without a good task estimation support."*

Some team members declared explicit resistance to this practice, but they were willing to perform it, which denoted a conflicting belief within the team and a low level of team autonomy. In this context, a Developer of the FD project declared:

> *"The Scrum Master asked us to register our time tracking in a tool to support him in monitoring the level of team productivity. I can understand that it is an important thing for him, but I didn't like doing it. Yeah, I'm resistant to the use of this practice."*

Results from Company 4

In Company 4, the project managers overcame the challenge of dealing with geographically distributed teams and deploying team communication practices. They not only adopted new communication tools, but, mainly, they brought past experiences and lessons learned to improve the communication process.

Also, the project managers were motivated to use individual productivity control as part of a task estimation practice for a new change management process during project monitoring. They cared about proposing new work processes which could become a good practice for the whole company.

In this case, evidence also showed that there was a significant connection between a team belief system and organizational culture. According to the participants, the company's senior managers fostered an attitude toward the ISO-9001 processes and procedures because of the company's certification program. The Quality Manager remarked:

> *"The Project Managers and Senior Managers are motivated to follow the ISO-9001 defined processes because they know their work is being seen, monitored and the problems could be escalated to the respective heads during ISO-9001 audits."*

Regarding the adoption of a new software development methodology, at first, some project teams showed explicit resistance, but after observing that the project knowledge could be better shared, they accepted the new practices. The Quality Manager said:

> *"Knowledge management is really important for this company. When people realize that, with the new methodology that we were proposing, it was easier to identify, capture and share the project knowledge, they assumed an attitude toward the new practices."*

With respect to the CMMI certification program, it seemed to fit very well in the context of a Development Center operational model, but not so well in an agile project scenario. The Project Manager of the SIG project said:

> *"There are some specific projects that are adopting this agile methodology, but we are not applying it here in the development center. In other branches, we have some projects in which it can fit into, but never when we need to strictly follow the procedures defined in ISO-9001 and CMMI certification programs."*

In the middle of the CMMI certification program, the project teams of Company 4 demonstrated low autonomy to adapt software practices to their needs. The evidence showed that there were cases of mandatory production of complex, time consuming and useless reports, only to be in compliance to the CMMI model. The Project Manager of the SIGEP project said:

"The project teams were forced to build some kind of monitoring reports that did not influence or help project monitoring. These reports were so complex and so dense that no one read them. In my point of view, reports that require so much effort are practically useless and may hinder the project progress."

In addition, the project managers were not motivated to build a suitable project plan, which indicated a low team commitment. In most cases, they didn't have full autonomy to conduct the project in compliance with the necessary practices for their project context.

Tabulating the Results of Companies 2, 3 and 4

In Table 23, we list the beliefs most frequently mentioned by the participants of Companies 2, 3 and 4; their respective class of beliefs, according to topics related to the software team practice; their respective attitude; the influence of the organizational culture; the level of the perceived behavioral control; the impact of each belief; and finally, their impact type, positive (+) or negative (-).

Table 23 CYCLE 2 – Salient Beliefs of Companies 2,3,4

Belief Class	*Belief*	*Attitude*	*Organizational Culture*	*Perceived Behavioral Control*	*Impact*	*Impact Type*
Process	Project management using SCRUM Methodology.	Toward agile software process	IT market and competition	High confidence	better productivity better response time high team-level effectiveness	+
Task Estimation	Project management using SE metrics supported by tools.	Toward precise task estimation practice	Motivated by fixed and early deadlines	High autonomy	better response time better project monitoring	+
Project Management	Bad project management increases the chance of failure in software projects.	Toward task delivery	High deadline pressure	Low autonomy and confidence	bad quality higher rework bad scope and cost management	-
Knowledge Management	Knowledge sharing practice through software documentation and planning increases the chance of success in software projects.	Toward project information sharing	CMMI certification program	High confidence	better quality better productivity better response time	+

With respect to hypotheses H1 and H3 linked to research question RQ2-2.1 – *"How do beliefs and attitude influence team practices in software organizations?"*, we addressed the participants' beliefs related to some real impact evidenced in their projects in order to identify the most influential belief in each situation. Also, we tried to evaluate the strength of beliefs among employees who perform software practices in terms of the frequency and importance and how these beliefs influenced the employees' attitude. Evidence to support hypothesis H1was found in the statements declared by the participants of all three companies. The interviewees had a common concern and interest in new approaches for

Project Management and they demonstrated attitude toward agile software practices in order to meet IT market demands. Another point was related to the Knowledge Management Belief. As a strong common belief to the participants, it had a positive effect on their attitude toward behavior, which, in turn, interacted with the behavioral intention to promote the adoption of the knowledge sharing practice. The impacts of these beliefs and attitudes on practices were positive and contributed to improve the implementation of them.

According to expectations, hypothesis H5 also linked to RQ2-2.1 was accepted. The data suggested that satisfactory past experiences with a behavior forms a strong condition for increasing one's intention to repeat the behavior in question. Project Managers of Company 4 brought past experiences and lessons learned to improve the team communication process, overcoming the challenge of dealing with geographically distributed project teams. In Company 2, the project team members showed some resistance to the adoption of new practices, particularly when it was not directly related to their past experiences. Accordingly, the relevant actions, interactions and events in the past of the project team members denote the common or conflicting beliefs between them, which can benefit or hinder the adoption of new practices on software organizations.

With regard to hypothesis H2 linked to RQ2-2.2 – *"How do organizational culture and subjective norms influence software team behavior and practices?"*, we identified that normative references of project teams, when in agreement with their underlying beliefs and cultural values, can reinforce the strength of organizational culture and norms and their influence on project team decisions. Knowledge management is really important for Company 4, so the resistance in accepting a new software development methodology was broken when the project team members realized that the project knowledge could be better shared. If it was easier to identify, capture and share the project knowledge, then it would be worthwhile to spend some effort in implementing the new methodology. Thus, hypothesis H2 is supported by evidence that attested the importance of the social normative influence of people who are significant in project team contexts.

By considering hypothesis H4 also related to RQ2-2.2, the study highlighted the influence factors associated with organizational culture and subjective norms to predict behavior intention. In Company 2, the CMMI certification culture influences the adoption of quantitative metrics for project monitoring. Also, some project team members reported that

the CMMI organizational culture seemed to foster a knowledge management culture through software documentation and planning. In the scope of Company 3, the agile methodology approach became part of the organizational culture and began to exert huge influence on the behavior of the employees and how they acted toward agile practices. The main positive impacts of this situation, such as higher team-level effectiveness and integration, better project response time and increased team productivity, were crucial for this to happen, since Company 3 was motivated to adopt agile methodology mainly by IT market competition.

When considering hypothesis H6 linked to research question RQ2-2.3 –*"How do team autonomy and confidence impact software practices?"*, we uncovered the meaning of events in companies 2 and 4 to reveal the effect of perceived capacity and difficulty on the self-efficacy sense. The STF project team of Company 2 reported that the CMMI organizational culture seemed to frame software practices to a more traditional development process in a cascade model, so they have struggled to put the agile model into practice. A very similar situation was identified in Company 4. The Project Manager of the SIG project reported that when it was mandatory to follow the procedures defined in the ISO-9001 and CMMI certification programs, they avoid deploying agile practices in the scope of the development center, because of disturbing conditions related to negative impacts evidenced in the projects.

Hypothesis H7 linked to research question RQ2-2.3 –*"How do team autonomy and confidence impact software practices?"* investigates the influence of self-efficacy sense on behavior intention. Self-efficacy is usually evaluated in relation to specific situations that represent challenges or impediments to successful performance. In Company 3, the Scrum Master of the FD project had the conviction that he could successfully implement the task estimation practice supported by tools. For him, it was a big challenge to predict team productivity without a systematic process for scheduling and task estimation, which could also contribute to a good implementation of agile software methodology in future projects. Some team members declared explicit resistance to this practice, but they were somehow willing to perform it, which denoted a conflicting belief within the team, but also the strong influence of the Scrum Master's self-efficacy sense on team behavior intention.

The analysis of hypothesis H8 also linked to RQ2-2.3 provided sufficient evidence to endorse self-efficacy sense as a direct determinant of behavior. During the data

analysis, two separate but interrelated situations representing self-efficacy and its impact on behavior were identified. In the scope of CMMI certification program of Company 2, the mandatory use of a bad format and useless artifact for requirements traceability is a stronger influence than the beliefs and attitude of the Project Leaders in determining team behavior and practice. In Company 4, the project teams were forced to build complex, time consuming and also useless reports, only to be in compliance to the CMMI model. The low sense of self-efficacy, in this case, had a negative impact on the deployment of the necessary practices for each project context. Additionally, were confirmed that behavior intention increases the chance of adoption of practices, providing support for hypothesis H9. However, behavior intention was found to fully mediate the effect of belief and attitude on behavior, but not the effect of self-efficacy sense, which can directly determine the adoption of practices in several situations, as described here.

5.1.5 Summary of Results

In this second phase of cycle 2, we developed a new case study to characterize team belief systems by applying the TPB theory to investigate the origins, sources, strength and related impacts, pointing out the role of past experiences in the adoption of practices and organizational culture influence on team software practice in industrial settings.

The research questions of the second phase of cycle 2 led to findings that attested that the main aspects to be studied in SE contexts go beyond the influence of team beliefs on project team practices. They should cover all the key influence factors and the relationship among them represented in the TPB model, as well as in the underlying the theory proposed in this dissertation (see Figure 2), which emerged from the synthesis of the findings of the second cycle presented in Chapter 7.

Thus, at the end of this second cycle, we obtained the following results:

- Characterization of team belief systems and their impacts on the adoption of new software development practices, providing rich narrative accounts for software engineering research.
- New insights about repeated behavior in discussing the role of past experiences in currently attitude-behavior models.
- Better understanding of how organizational culture actually impacts industry software projects.
- Guidelines to help companies to improve project team practices through the knowledge of team belief system and its impacts.
- Practical knowledge on how to apply behavioral theories to study software practices, enabling further examination through confirmatory studies.

This study has important implications for research and practice. It provided guidance to academic research directions and practical and useful recommendations for companies in order to strengthen project team attitude toward software practices. The study pointed out some key issues related to the adoption of new practices and contributes to software process improvement. In the next chapter, the recommendations are presented and discussed during a cross-case analysis.

Chapter

6

This chapter presents a timeline of the case studies in the participating companies, their cycles and phases, as well as the main research approach and key findings encountered. Then, it presents a cross-case analysis of our findings, considering the evolution and knowledge gained from the case studies, and lists some practical recommendations for software companies that are adopting new practices. It also revisits the three main research questions RQ1, RQ2 and RQ3 of our research based on the synthesis of the findings of the second cycle. Lastly, this chapter discusses practical implications drawn from the developed research.

6 DISCUSSION OF THE INFLUENCE FACTORS ON SOFTWARE PRACTICES: A CROSS-CASE ANALYSIS

6.1 CASE STUDY TIMELINE

The research described here involved a long-term study essentially composed of three case studies over two cycles, and the second cycle divided into two phases. The first case study was conducted using an ethnographic approach, immersing ourselves in the day-to-day project activities of Company 1 for 18 months. We conducted the second case study by applying TRA in the same company studied in the first cycle during a 12-month period and the third case study by applying TPB in three different software companies with very distinct profiles (Companies 2, 3 and 4) in a 6-month period.

The following figure outlines the timeline of the case studies in the participating companies, their main research approach, as well as the key findings encountered in each cycle and phase.

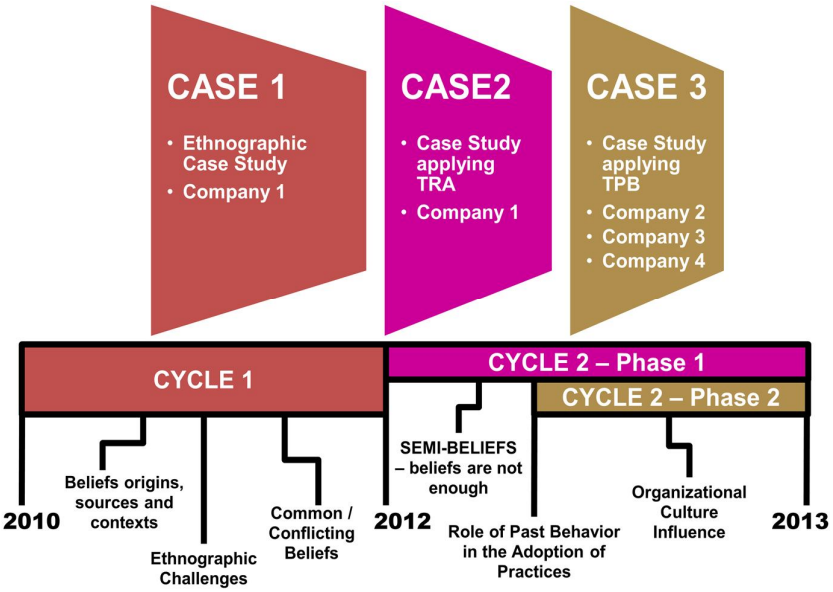

Figure 17 Case Study Timeline

6.2 RESULT SYNTHESIS

In this section, we synthesize the results of the second cycle of our research. This second cycle expanded the framework of the first cycle, and involving the company originally studied in the first cycle (Case 2 of Figure 17) and three other companies with different profiles (Case 3 of Figure 17).

Through the synthesis of two exploratory case studies of cycle 2, we investigated the influence factors related to team beliefs, attitudes, team autonomy and confidence, and the organizational culture and normative pressure embedded in the software development context to understand team behavior intention and practice. We now discuss the findings and synthesize the results in light of our three main research questions (RQ1, RQ2 and RQ3).

The cross-case analysis method was adopted for the results synthesis. It seemed to be an appropriate method to synthesize impacts of an object of study on software development contexts. The aim was to see processes and practices across multiple cases to

understand how they are qualified by local conditions and, thus, to develop deeper descriptions and more powerful explanations. We categorized, tabulated and analyzed the individual cases as windows to compare and give insight for the cross-case analysis. In this analysis, we considered in what areas the cases suggest the same points, where they differ, and where the cases conflict. The comparison of the cases brought no relevant difference, but we found that the cases complement one another, and additional observations arose.

We adopted a mixed strategy, combining the case-oriented and variable-oriented approaches represented in Figure 18. We analyzed each case in depth, looked for themes that cut across multiple cases, condensed the pattern found to permit systematic comparison, and then wrote up the final results using the expanded variables defined by our conceptual frameworks based on TRA and TPB models.

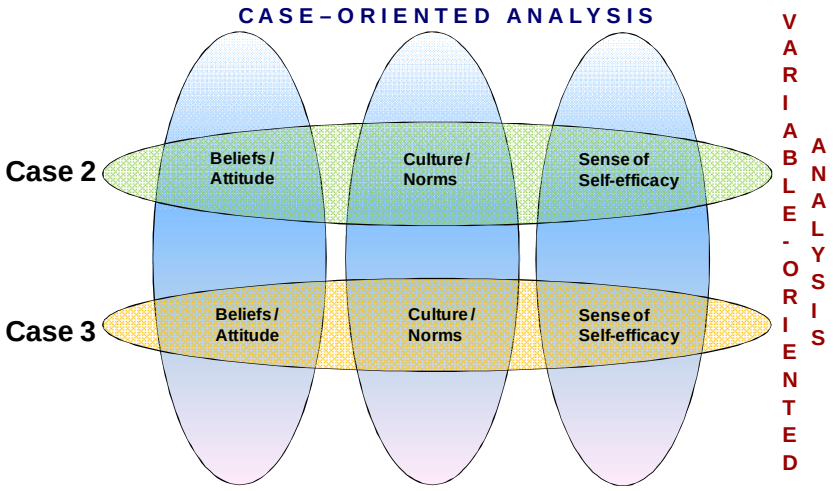

Figure 18 Cross-Case Analysis Process

Our mixed strategy of cross-case analysis, combining case-oriented and variable-oriented approaches, aimed at analyzing each case in a systematic way and representing the final results using the hypotheses on the relationships between the expanded variables of TPB (Figure 16).

Table 24 provides an overview of how the sub-research questions for each research cycle relates to the main research questions in this thesis. The answers and discussion of the three main research questions presented below are rather concise. For a more thorough explanation and evaluation, see Chapters 3, 4 and 5.

Table 24 Relation between Research Questions

Main RQs	Sub RQs
RQ1	RQ1.1
RQ2	RQ1.2 RQ2-1.1, RQ2-1.2 RQ2-2.1, RQ2-2.2, RQ2-2.3
RQ3	RQ1.3 RQ2-1.3

Table 25 provides a synthesis matrix of our results in the form of a table. It summarizes evidence drawn from the two complementary cases of the second cycle. This evidence is coded under thematic headings in order to contextualize each theme by comparing different usage contexts. It lists the salient beliefs of Companies 1, 2, 3 and 4; the respective class of beliefs; the belief strength as semi-beliefs or strong beliefs; the conflicting part of each belief; the level of the perceived behavioral control; the evidence of the influence of the organizational culture; the impact of each belief; their impact type; and finally, the company involved.

Table 25 Synthesis Results – Salient Beliefs

Belief Class	*Belief*	*Strength*	*Conflicting Part*	*Perceived Behavioral Control*	*Organizational Culture*	*Practice Impact*	*Impact Type*	*Company*
Task Estimation	Project management using SE metrics supported by tools.	strong	About the adoption of task estimation tools.	High autonomy	Motivated by fixed and early deadlines	better response time better project monitoring	+	2,3,4
Process	Project management using SCRUM Methodology.	strong	About the compatibility between CMMI traditional development and agile models.	High confidence	IT market and competition	better quality, productivity high team-level effectiveness	+	2,3
Project Management	Bad project management increases the chance of failure in software projects.	strong	About the documentation required for project management	Low autonomy and confidence	High deadline pressure by customers	bad quality high rework bad scope and cost mgmt	-	1,2,4
Knowledge Management	Knowledge sharing practice through software documentation and planning increases the chance of success in software projects.	strong	----	High confidence	CMMI certification program IT market demands	better quality, productivity better response time	+	2,3
Team Commitment	To achieve team-level effectiveness through team commitment is a good idea.	strong	About the impact of self-management practices on team productivity.	High confidence	Self-management team practice adoption	Better productivity	+	1
Team Autonomy	The lack of productivity and quality metrics makes task estimation imprecise, which disturbs team autonomy by jeopardizing deadline negotiations with customers.	semi	About the direct impact of agile practices on software quality.	Low autonomy	High deadline pressure by customers	Low team-level effectiveness	-	1,2
Team Maturity	The low level of team member seniority disturbs project progress and can delay software projects.	semi	----	Low autonomy and confidence	Organizational culture of low operating cost.	Slow project progress	-	1

6.3 IMPLICATIONS FOR RESEARCH

The synthesis of phases 1 and 2 of cycle 2 offered insights into the role that the key influence factors play in the adoption of software development practices. This section summarizes the findings that helped us to answer the three main research questions properly. Thereby, we directly answer question RQ1, and suggest three contextualized responses to RQ2 and two responses to RQ3.

RQ1:What are the main factors that influence software development team behavior intention and practice?

Answering RQ1 resulted in two major contributions (or final results) listed in Section 1.4, R1 and R2, dealing with identifying and mapping the team belief origins, sources and contexts and the significant relationships between team belief systems, including the key influence factors, and team practices in software organizations.

This research aimed at characterizing belief systems of software project teams by identifying the key influence factors that determine the software engineering practice and behavior, and hence, the outcome of software projects. We concluded that the practice of software development depends on a number of factors, particularly the system of beliefs concerning software engineering and its practice; the social and organizational context of the software projects; and the team's level of autonomy and confidence.

The results highlighted the complex nature of the relationship between belief and practice. Only the beliefs that are the most salient can exert strong influence on practice during software projects. Salient beliefs operate as determiners, they usually come to mind when respondents are asked open-ended questions and commonly exceed a particular frequency. Thus, they can be highly predictive of both intention and behavior. Also, normative beliefs with respect to the person who is the main reference were also significantly related to practice.

Furthermore, beliefs are influenced by the interaction within the nested social contexts within which they are situated. They have different levels of strength in different contexts. Thus, beliefs and practices cannot be examined out of context, but should be always situated in a physical setting in which constraints, resources and opportunities may derive from sources at various levels (organizational, team and individual).

Additionally, salient beliefs about behavioral consequences are predictive of practice. It is important to take into account the possible benefits or barriers, and self-efficacy sense to explain the consistency and inconsistency between belief and actual practice. Facilitating or disturbing conditions mapped as characteristics of the project teams and contexts can make it easier or more difficult for the project team to carry out its initial intention. The low sense of self-efficacy has a significant negative impact on the deployment of software practices.

Regarding the origin and sources of beliefs, evidence suggests that beliefs based on personal experience are stronger than beliefs acquired through information from outside

sources and likely to be highly resistant to change, even when these past experiences were taken into account without much consideration for their original context. The attitude of a project team when adopting a new practice is considered in the context of the influences arising from the practices in use at that time and the beliefs that came from their past experiences and repeated behavior, which means some beliefs persist over time as they form the foundation of many of team intentions and practices.

Past experience or repeated behavior proved to be a very strong influential factor and predictor of intention. Our results supported the argument that past behavior performed repetitively acts as a moderating variable of the relationship between intention and practice. Satisfactory past experiences with a behavior are a key condition for trend development as they increase team intention to repeat the same behavior in the next time a similar situation arises, because they believe that this behavior will lead to a satisfactory outcome.

Thus, the main aspects to be studied in SE contexts go beyond the influence of team beliefs on project team practices. They should cover all the key influence factors and the relationship among them represented in the underlying theory proposed in this dissertation (see Figure 2).

RQ2:How do these factors influence the adoption of new practices in software teams?

Response 1 –A good fit between the organizational culture and the basic assumptions of a project team reinforce the strength of the team attitude toward a new practice.

Our findings indicate that the participant's past experience or repeated behavior is the most influential factor to predict team behavior. These past experiences were clearly described in specific and directed stories reported by the participants during the meetings and interviews. They originated from a personal hands-on approach that did or did not work well on previous projects. All the participants' past experiences have framed individual basic assumptions, and brought forth the team beliefs and cultural values derived from these experiences. Usually, the past experiences were taken into account without much consideration for their original context and their suitability for the present time in the project or organization.

In all cases and companies, the study indicated that the team members' past experiences were influencing their attitude toward software practices. With respect to the task

estimation issue, in Company 1 some interviewees reported that imprecise task estimation in their previous projects led to negotiations on bad terms with customers and, consequently, low team-level autonomy. For them, the lack of productivity and quality metrics was the main cause for the task estimation being disorganized, which hindered their autonomy. In contrast, in Companies 2,3 and 4 the task estimation practice was related to positive consequences. In Company 2, the CMMI certification culture influenced the adoption of software engineering metrics for better project monitoring. In Company 3, some team members were resistant to this practice because of bad experiences from the past, but the Scrum Master of the FD project was quite confident that the adoption of a task estimation practice supported by tools could help in the implementation of agile software methodology, so the whole team was engaged in this practice. And finally, in Company 4 the Project Managers were motivated to use individual productivity control as part of a task estimation practice for a new change management process that they proposed. In this company, people are encouraged to propose new process improvements and they are well recognized for doing so.

In the context of the CMMI certification program of Company 2, some members of the STF project reported that focusing on knowledge sharing through software documentation and planning was one of the main resources to improve software quality and increase team productivity. In Company 3, most members of the FD project team agreed that achieving knowledge sharing through a project tracking blog was a good practice because this practice improved team productivity and integration. Also, in Company 4 some project teams showed explicit resistance to adopt a new proposed methodology, but after observing that the project knowledge could be better shared, they realized that the new practices could bring significant benefits. Therefore, the knowledge management practice was in agreement with the basic assumptions of most interviewees.

Accordingly, we observed that when the basic assumptions of team members are in agreement with the organizational culture, they exert greater influence on the team's attitude, because the organizational culture can reinforce the confidence that the particular behavior will lead to positive consequences.

Thus, answering RQ2 over hypotheses H1 and H2 (see Figure 16) derived in two major contributions (or final results), R2 and R4, dealing with the characterization of significant relationships between project team shared assumptions and beliefs, and new

practices in software organizations, and understanding how the organizational culture impacts software industry practice, considering the role of past experience or repeated behavior in this context.

Response 2 –Organizational culture influences software practice through its convergence with team beliefs and cultural values toward behavior intention.

All companies analyzed had agile approaches focused on team-level effectiveness. Practitioners had experienced similar agile practices before in other companies and their past experiences have influenced their beliefs and attitude toward agile practices in different ways. Some team members with old culture values, drawn from previous projects in different companies, found it hard to adapt to new software practices.

In Company 1, the interviewees declared that organizational support and culture were essential to achieve a good team performance. In a scenario of high customer pressure for deadlines, there was a high level of organizational support for self-management practices and some of the key self-management practices were actually being implemented in the company. It was part of the company's strategy to achieve the level of commitment required to meet the project goals.

People in Companies 2, 3 and 4 agreed that good task estimation supported by appropriate tools was a good practice to adopt in software projects. This practice was encouraged by the companies because it could lead to early deadlines.

In Companies 2 and 3, the agile approach became part of the organizational culture and exerted huge influence on the behavior of the team members and how they acted toward agile practices. From the perspective of the Developers and Technical Leaders, there were many positive impacts in adopting this approach such as: better team productivity and product quality, and higher team-level effectiveness and integration.

In Companies 2 and 4, the CMMI certification program seemed to frame software practices to a more traditional development process in a cascade model, which could not be so compatible with agile methodologies. This kind of certification program influenced the company's culture, and sometimes, obstructed the progress of agile practices due to the introduction of complex and time consuming artifacts.

The most notable difference between the organizational culture of Company 1 and the organizational culture of the other three companies was the emphasis of Company 1 on

team commitment. Commitment was identified as a strong value for Company 1 and for its customers, so it was a present value translated into many project practices. On the other hand, for Companies 2 and 3, there was a significant concern about the knowledge sharing through software documentation and planning. In these organizations, this practice improved team productivity and integration, which, in turn, led to a good team performance.

From these results, we confirmed that there is a significant connection between organizational culture, when in agreement with project team beliefs and cultural values, and behavior intention as our underlying theory indicates (see Figure 2). More evidence to support this was found in the statements declared by the participants about how much organizational support and culture were essential to achieve a good team performance and how much other cultural influences, resulting from previous projects, could set back or advance the adoption of software practices, depending on which beliefs or cultural values the project teams had in mind.

Thus, responding RQ2 over hypotheses H1, H2, H3 and H4 resulted in two major contributions, R4 and R5, involving a better understanding of how the organizational culture impacts project team behavior intention, and some guidelines on how to strengthen the companies' culture and norms toward software practices that support the companies' goals and targets.

Response 3 – The self-efficacy sense of a project team impacts on team practice directly.

In all cases and companies, the study indicated that the project team self-efficacy sense was related to how easy or difficult the behavior of interest could be and the possible benefits or barriers.

In Companies 2 and 4, we collected sufficient evidence to endorse self-efficacy sense as a direct determinant of behavior. Both situations were related to the CMMI certification program in which the companies have engaged.

In the scope of CMMI certification program of Company 2, the mandatory use of a bad format and useless artifact for requirement traceability was a stronger influence than the beliefs and attitude of the Project Leaders in determining team behavior and practice.

In Company 4, the project teams were forced to build complex, time consuming and useless reports, only to be in compliance with the CMMI model. They demonstrated low

autonomy to adapt software practices to their needs. The perceived power of the CMMI certification program inhibited behavior performance and caused team members to go against their initial intention which was not spending any effort on so complex and dense reports that no one actually reads. The low sense of self-efficacy, in this case, had a negative impact on the deployment of the necessary practices for each project context. From this perspective, behavior intention not only mediated the effect of self-efficacy on behavior, but also directly determined the adoption of practices, taking into account the facilitating and disturbing conditions of the context.

Thereby, meeting RQ2 over hypothesis H9 led to two major contributions, R2 and R5, dealing with the characterization of significant relationships between team belief systems, including self-efficacy sense as a key influence factor, and team practice in software organizations, and some guideline on how to reinforce the team self-efficacy sense providing the right resources and opportunities.

RQ3:How do common and conflicting beliefs and cultural values impact positively or negatively on software team behavior and practice?

Response 1 – Common strong beliefs and cultural values can become embedded in the organizational culture as they are strengthened and transferred to other practitioners in the company.

There were cases of common strong beliefs as present values reflected into many practices that the project teams actively act upon and always with positive results. These beliefs are usually implemented since the very beginning of the projects and strengthened and transferred to other people involved.

In Company 1, the team communication practice was considered effective in all the software projects investigated. Also, commitment was identified as a strong value for the company and it was a present value translated into many practices. Both situations represented beliefs that the project teams actively acted on with positive consequences.

As already mentioned, in Companies 2,3 and 4, the Knowledge Management Belief was reflected into practices that the project teams actually adopted. Most members of the software project teams agreed that achieving knowledge sharing was a good practice and they acted to reinforce it. Other beliefs related to task estimation began to reflect the values of

the organization and became part of the organizational culture of Companies 2,3 and 4. The adoption of a task estimation practice supported by tools helped in the implementation of agile software methodology, so this practice was encouraged by the companies because it has contributed to bringing forward deadlines.

Also in Company 2, there was a common belief about requirement traceability control. For team members, the mandatory use of a bad format and useless artifact for requirement traceability jeopardized project management effectiveness, incorporating critical issues to the project. And in Company 4, the Project Managers overcame the challenge of dealing with geographically distributed teams by improving team communication practices through their past experiences and cultural values derived from other companies. So, these values also became present in the organizational culture of Company 4.

Answering RQ3 over hypotheses H1, H2, H3 and H4 resulted in two major contributions (or final results) listed in Section 1.4, R3 and R4, dealing with representing common strong beliefs and their positive impact on team practice, and understanding how the organizational culture impacts software industry practice, process and decision, considering the role of repeated behavior in this context.

Response 2 – Conflicting beliefs and cultural values impact negatively on software team practice because they break the unity of the project team as a group with autonomy and confidence.

When applying an underlying theory adapted to the problems and objects of study in the field of Software Engineering, we analyzed the relevant actions, interactions and events in the past of the team members, trying to uncover semi-beliefs or conflicting beliefs between them, which could hinder the adoption of new practices.

In the scenario of Company 1, the Maturity Belief and Team Autonomy Belief were examples of common semi-beliefs, which means beliefs without attitude, and as such, they had negative impacts on team practices. There were also conflicting beliefs between the stakeholders involved in the software projects. The team members of the DMS project disagreed about the direct impact of agile practices on software quality. These conflicting beliefs indeed hindered the adoption of new practices in the DMS project, such as shared leadership and collective decision-making.

Regarding the adoption of agile methodologies to project management in Company 2, there were some conflicting beliefs between the three participants involved in the STF project about the compatibility between CMMI traditional development and agile models. Also in Company 2, the Manager of the Development Center had an attitude toward the use of task estimation practices, which aimed to improve team autonomy.

In Company 3, some team members declared explicit resistance to the adoption of a task estimation practice supported by tools, but they performed it as a mandatory task, which denoted a conflicting belief between team members. Also in Company 4, some conflicting beliefs jeopardized the adoption of new practices. The Project Managers complained about the mandatory production of complex and time consuming reports during the project. In the Quality Manager's point of view, these reports were necessary and could be useful for project monitoring, if they were built properly and right on time.

Some of these conflicting beliefs were subtle and our approach of formally capturing and representing team beliefs and cultural values definitely helped to identify and highlight these conflicts between the involved stakeholders. Accordingly, responding RQ3 over hypotheses H1, H6 and H7 derived in one major contribution (or final result), R3, involving the representation of conflicting beliefs and their negative impact on team practices that proved to be a useful tool to expose and discuss these conflicting issues with the practitioners of the companies involved in order to motivate the improvement of software development team practices.

Therefore, bringing this discussion to a close, all these responses to the research questions RQ1, RQ2 and RQ3 cast light on the influence factors that impact the team's intention of behavior and practice in software organizations. Beliefs about the likely consequences of behavior, about the normative expectations of important others, and about skills, resources, opportunities and other aspects that can facilitate or hinder behavior performance jointly influence the decision to engage or not in the behavior of interest.

Regarding the methodological results (R6 and R7), a descriptive analysis about the benefits of using a collaborative ethnographic approach and the key challenges of applying ethnography to study software practices was performed during the first case study and a summary of the results is listed in Appendix D.

6.4 IMPLICATIONS FOR PRACTICE

This study has important implications for practice. In this dissertation, after conducting a long-term research with the aim of characterizing belief systems in software industrial contexts within a behavioral perspective, we intend to use this experience to motivate the improvement of software development team practices. The research included organizational and team level factors associated to beliefs and practices in order to provide guidelines for the software industry to support the adoption of new practices. We list functional and practical recommendations for professionals with leadership and management profiles and involved in companies' software projects that are dealing with the challenges of adopting new practices on software projects.

Here we present some evidence-based recommendations to foster the adoption of new practices in software organizations. The question is which type of beliefs and cultural values should be built into the organization culture to support the specific goals of the companies and how these cultural values might be associated with certain types of practices. Based on our findings, we recommend the following:

Recommendation #1

Check, compare and confront team beliefs emerged and evolved from past experiences in organizational and project contexts. It is important to understand the ways that these beliefs reflect attitudes at the current time.
Related Contributions: R1 and R2
Clarifications: The team members' beliefs and basic assumptions should be exposed in retrospective sessions, when they will be able to retell and revive specific and directed stories that illustrate their past experiences. Thus, they will express their key values, ideals, morals, customs, and the beliefs that other team members will come to understand and share. All new information about others' experiences has the potential to affect team attitude based on the strength of existing beliefs.

Recommendation #2

Organize integrated learning events to share team's past and current experiences in order to develop team confidence and to increase its capacity.
Related Contributions: R1, R2 and R4
Clarifications: Knowledge Management is today one of the main resources that a software organization has to improve software quality and increase team productivity. Satisfactory past experiences should be shared and placed in the current context, so they can be evaluated. Learning events can support this evaluation, combining short workshops or group sessions with periods of group work-based practice and reflection, including some training or development activities. The learning/training records of staff should be captured by audio, video or writing, and thus, they can be shared with everyone involved.

Recommendation #3

Recognize and value the common strong beliefs reflected into practices that yielded positive results to increase team-level autonomy.
Related Contributions: R3

Clarifications: The more appreciated the positive result of a practice is, the stronger the team intention will be to repeat the practice in question. Increasing team autonomy level will lead to the strong belief that teams can freely make the decision to use the necessary resources and opportunities to perform the practice. It is reflected into practices that team members actually adopt and seek for positive results. Team practices, that yielded positive results, should be mentioned in special meetings and celebratory events of victories and gains.

Recommendation #4

Manage the conflicting beliefs and beliefs without attitude that hinder the adoption of new practices.
Related Contributions: R3
Clarifications: Some conflicting beliefs are subtle. An even subtler problem is when team members has a common belief, but do not act upon it. The team members' conflicting beliefs should be exposed in retrospective sessions and project meetings, when all team members have the opportunity to elucidate and defend their point of view. A moderator in these sessions can help to better understand the causes of conflicts or lack of action, and orient new initiatives to introduce the practice or a variant of it.

Recommendation #5

Put more time and energy into planning and preparation for the adoption of new practices to increase team confidence to deal with software quality and productivity challenges.
Related Contributions: R2
Clarifications: The main idea is that the easier a practice is to use, the more useful it can be and enhance job performance. Thus, a favorable practice environment with support mechanisms will contribute to break the barriers of the confidence of the project team and motivate it to act and seek for benefits.

Recommendation #6

Package the new practice with support materials (such as training guides, help functions and simple interfaces) that make adoption relatively smooth and painless.
Related Contributions: R2
Clarifications: There are four main elements necessary for the successful implementation of a new practice: (i) the new practice packaged in a digestible form; (ii) a credible dissemination strategy involving influential team members; (iii) a supportive practice environment, including, tools and other packaging and support mechanisms to aid the use of the practice; and (iv) a good grasp of the local practices, values, and beliefs into which new practices must be integrated.

Recommendation #7

Promote the balance between individual-level and team-level autonomy with a focus on a holistic project view by preventing each team member from caring only about his/her task.
Related Contributions: R2
Clarifications: The first thing to do is to shift from traditional command-and-control management to collaborative self-managing teams by putting all the liability at the team level, so all team members have responsibility for project tasks and activities. When people work together they are more likely to commit to common goals. Developers should not work alone on a module and the project plans should be realistic with agreements and commitments among those who have to carry out the project tasks.

Recommendation #8

Maintain good communication between senior management and project teams of the company to build trust and a good influence on the team members' attitude.
Related Contributions: R4

Clarifications: Good communication means the door is always open to receive project team requests or complaints. The Senior Managers should seek to transmit good cultural values, such as sharing, openness, and trust that will lead to positive behavior. Internal workshops and events to discuss project progress can be a good opportunity to communicate well and openly, and also talk about project issues and possible solutions.

Recommendation #9

Build trust and commitment between the organization and the project teams. When people work together toward a common goal, trust and commitment follow them. So, make sure both the organization and the project teams know and respect the common goals.
Related Contributions: R4
Clarifications: A misalignment between team expectations and organizational expectations can be very counterproductive. Thus, achieving a good fit between the organizational culture and the project context together with greater team collaboration can reduce this problem. Team members in a joint effort are more likely to commit to a common purpose between them and the organization. If the project teams are aware of the company's plans, their attitude toward a goal will be based on how favorable the total set is and they will consider its evaluation according to the organizational and project perspectives.

All recommendations should be considered by the Project Managers and Technical Leaders of the companies. Furthermore, recommendations 1, 3 and 4 should be considered by software practitioners in general.

Chapter

7

This final chapter presents the conclusion of this thesis, along with its major contributions. This chapter also addresses the limitations and validity issues of the performed work as well as the recommendations for further research.

7 CONCLUSION AND FURTHER WORK

The research carried out in this dissertation has provided valuable insights and conclusions on the research problem and questions, and resulted in eight major contributions (R1 to R8). Below we sum up our main conclusions, make an evaluation of the limitations of this dissertation with respect to both rigor and relevance, and outline possible future work based on our main findings.

7.1 RESEARCH CONTRIBUTIONS REVISITED

The objective of this dissertation was to characterize the belief systems of software project teams. The study addressed the key influence factors associated to project team beliefs, its attitude toward behavior, the organizational culture and subjective norms, and the resulting sense of self-efficacy of project teams to predict behavior intention. We expected to facilitate the understanding of this complex process by showing how their attributes can be related in order to present them in an underlying theory as an instrument for representing team belief systems in SE contexts. Also, based on the research goals and the answers provided to

the three main research questions posed in Chapter 1, this dissertation contributes with practical and methodological knowledge.

7.1.1 Theoretical Model Proposed

The proposed theory, represented in Figure 2, is composed of three main components. The first component represents the strength of beliefs through *team beliefs* and its *attitude toward behavior*. The second component consists of *organizational culture* and *subjective norms*. It can be translated into the expectations of other people in a social environment and also how they will view the behavior in question. The third one is the *perceived behavioral control* component, which denotes people's perception of their degree of capacity and difficulty to carry out their intention or perform a certain behavior.

Another component arose from the results obtained in the studies described in Chapters 3, 4 and 5. *Past behavior* proved to be a very strong influential factor and predictor of behavior intention. Satisfactory past experiences with a behavior forms a strong condition for increasing one's intention to repeat the behavior in question. In addition, three new relationships were identified. The *team belief and values* element, when in conflict situations, has a relationship to the perceived behavioral control component, which means that this element directly affects *team confidence and autonomy*, impacting project team practices in a negative way. Beyond that, *perceived behavioral control* is a direct determinant of behavior. Also, *common strong beliefs* of project teams can increase the organizational culture influence, which affects *team attitude* toward a new practice. All these new elements have the power to influence the model of behavior.

In summary, according to the theory proposed and illustrated in Figure 2, project teams will be motivated to try to perform a new practice and to expend effort and persevere in their attempts, if: (i) they believe they can perform it, because they have capacity and no difficulty; (ii) they have strong beliefs plus information about the behavior and its likely consequences; (iii) they recognize a link between satisfactory past experiences and the course

of action; (iv) they are supported by organizational and team level factors plus normative expectations of others; and (v) they have positive perceived behavioral control and understand the likely facilitators and impediments of their performance.

Besides the limitations of not dealing with factors associated with team personality characteristics, our proposed theory has numerous applications in the SE area from project team planning to software process definition, deployment and evolution.

7.1.2 Practical Contributions

Based on the findings of the investigations in this dissertation, we have made several contributions for practice as follows:

R1. Identification and mapping of team beliefs origins, sources and contexts;

R2. Characterization of significant relationships between team belief systems and practices in software engineering organizations;

R3. Representation of *common* and *conflicting* beliefs and their impact on practices, processes and decisions in industrial contexts;

R4. Understanding of how organizational culture and norms actually impact software industry practice and the role of repeated behavior in this context;

R5. Guidelines to improve software team practices through the knowledge of team belief system.

We expect to introduce innovative results by applying our conceptual frameworks, based on behavioral theories, to study software practices, and contribute to an improved understanding on how to predict team behavior intention and better perceive how people progress from intention to practice in organizational environments.

7.1.3 Methodological Contributions

The main methodological contributions of this dissertation are:

R6. Relevant insights on how to deal with the key challenges of applying ethnography to study software practices;

R7. Presentation of the benefits of using a collaborative ethnographic approach, including a participatory action research strategy – collaborating with the participating companies is as important as studying them;

R8. Clarification on how to apply behavioral theories to study software engineering practices through the proposal of an underlying theory adapted to the problems and objects of study in the field of Software Engineering.

The results of the methodological analysis about the benefits of using a collaborative ethnographic approach and the key challenges of applying ethnography to study software practices is listed in Appendix D, and also, we related our findings to relevant behavioral literature in order to support others who wish to conduct this type of study, making the results more organized to endorse prior conclusions.

7.2 LIMITATIONS

Several limitations were identified in this work, especially during the studies reported in the previous chapters.

This study is one of the first initiatives of applying behavioral theories in the context of SE practices to guide research in software organizations. We do not have a complete list of recommendations and implications for research or practice yet, thus further studies should be performed to point to other possibilities of applying behavioral theories in SE contexts.

Common criticisms to a long-term case study also apply to this study, among them one may list: uniqueness, difficulty to generalize the results, and the introduction of bias by participants and researchers. However, the case study method is suitable for industrial evaluation of software engineering practices, methods and tools because it can avoid typical scale-up problems observed in small experiments. Whereas formal experiments record the variables that are being manipulated, case studies collect information from the variables representing typical and real situations and events. In summary, the most important aim of a case study is to explain the factors in a real-life context that are too complex for the survey or other experimental approaches.

In our study we generalized the findings from empirical statements to theoretical statements, which involved generalizing data from interviews and perceptions by discussing them in accordance with the behavioral literature. In this respect, we related our findings to relevant examples of the application of behavioral theories and compared them with results found in information systems research. We also triangulated data from different qualitative sources: data from interviews and observations, document analysis and focus group meetings, which was important to corroborate findings of the case studies. Interview data was our primary source of the key influence factors associated with organizational culture and subjective norms to predict behavior intention.

Our findings apply to software projects teams within just four participating organizations. However, all the participants were professionals using typical development technologies in a typical working environment, e.g., the natural setting demanded by the case study approach. We worked with real world project data and the participants chosen for the interviews were experienced project team members. They were capable of accurately and reliably answering the interview questions. Additionally, we applied consolidated behavioral theories in the research design of the second cycle in order to improve internal validity.

Thus, regarding external validity, we described the main characteristics of each case and company, including context and settings, data collection, analysis, and synthesis process, as well as quotations with our major findings. This made the results easier to generalize. Nonetheless, the qualitative findings are highly context and case-dependent. As commonly done in-depth qualitative studies, we also had to do a trade-off between the number of participants, the duration and the cost of this study. The number of subjects

interviewed in this context is not quantitatively significant, but gave deeper insights on the issues investigated in this work.

The relationship between researchers and our participants has been particularly significant in two aspects. Firstly, we have chosen a field of study where we belong to the community being studied, so we could reasonably be described as software practitioners or as research scientists. Secondly, we had an active, but secondary role in the target projects, because we acted as consultant for quality assurance and process control in one of the companies investigated (Company 1). Being a member of the community under study has both challenges and advantages for the researcher. The main advantage is that researchers and practitioners in this context share the same organizational culture and so researchers are accepted and have common vocabulary with the participants, so the participants can relax and focus on working in their natural way. The challenges arise from the tension involved in moving between two worlds in the need to be non-judgmental in order to avoid bias in collecting and interpreting data. So, we decided to discuss and validate all interview notes and some observations with other researchers and with the participants.

Even though we applied some aspects of a grounded theory approach to form initial insights inductively on the basis of the case studies data, our proposed theory is still a subjective construction. Besides, the process of interpretation is iterative and the subjective constructions and their associated realities are likely to be seen differently. The question, therefore, is whether the model is informed and elegant as an evolution of our conceptual frameworks adapted to the problems and objects of study in the field of Software Engineering.

Lastly, there is also a risk that our findings could be influenced by factors that escaped our attention. To mitigate this, we chose to have the evidence reviewed by more expert researchers and to perform agreement rounds in order to seek the completeness of the conclusions.

Despite these limitations, this study contributes to the growing literature on qualitative SE research and provides empirical support for the importance of organizational issues in the software practice improvement.

7.3 FUTURE WORK

The findings of this dissertation, its contributions and limitations, indicate several possibilities for further research with relevance for SE theory, methodology and practice.

- **Refinement of the research instrumentation:** Further empirical studies using the research instruments listed here should be performed. This could contribute to evolve and improve the instruments and research process for a next round of qualitative research in SE field.
- **Further testing of the proposed theory:** Further studies should be performed, where new data are collected from other software companies. This could be used to validate our conclusions presented in this dissertation.
- **Further study of the recommendations and guidelines:** Further studies should be carried out to investigate how our recommendations are implemented in the companies, and whether the recommendations have had the intended effect.
- **Further research to adapt our proposed theory to other subjects in SE area:** Efforts should be made to develop new studies to investigate other relevant aspects of SE dynamics and contexts. The nature and effect of other factors in the social environment underlying current SE theory and practice should be further investigated.
- **Investigation of the research question *How can empirical results be incorporated as useful beliefs in software organizations?*:** To answer the question of how to expand the use and influence of empirical results in software industry settings, further studies should be conducted to explore this question in a more extensive research, involving a representative number of participating companies around the world.

There are certainly other directions for further research. However, the value of any such future work depends on the perceived relevance of the research problem of each

particular audience. The results of the empirical research presented here in this dissertation support our claim that organizational and team factors associated to project team beliefs impact practices, processes and decisions in the software industrial context.

BIBLIOGRAPHY

ADAMS, D., NELSON, R., TODD, P. **Perceived usefulness, ease of use, and usage of information technology: a replication**. MIS Quarterly, 16(2): 227–250, 1992.

AGUIRRE, J., SPEER, N. **Examining the relationship between beliefs and goals in teacher practice**. Journal of Mathematical Behavior, 3(18): 327-356, 2000.

ATTRIDE-STIRLING, J., **Thematic networks: an analytical tool for qualitative research**. Qualitative Research 1(3): 385–405, 2001.

AJZEN, I, FISHBEIN, M. **Understanding attitudes and predicting social behavior**, 1st edition, USA: Prentice Hall (Englewood Cliffs, N.J.), 1980.

AJZEN, I. **From intentions to actions: A theory of planned behavior**. In J. Kuhl & J. Beckman (Eds.), Action control: From cognition to behavior (pp. 11–39). Berlin: Springer-Verlag, 1985.

AJZEN, I. **The Theory of Planned Behavior. Organizational Behavior and Human Decision Processes**, 50(2):179–211, 1991.
Available at http://people.umass.edu/aizen/obhdp.html .

AJZEN, I., FISHBEIN, M. **The influence of attitudes on behavior**. In D. Albarracín, B. T. Johnson, & M. P. Zanna (Eds.), The handbook of attitudes, pp. 173-221. Mahwah, NJ: Erlbaum, 2005.

AJZEN, I., GILBERT COTE, N. **Attitudes and the prediction of behavior**. In W. D. Crano& R. Prislin (Eds.), Attitudes and attitude change, pp. 289-311. New York: Psychology Press, 2008.

ALAVI, M., KATWORTH, T., LEIDNER, D. **An Empirical Examination of the Influence of Organizational Culture on Knowledge Management Practices**. Journal of Management Information Systems, 22(3): 191–224, 2006.

BAIN, J.D., MCNAUGHT, C. **How academics use technology in teaching and learning: understanding the relationship between beliefs and practice**. Journal of Computer Assisted Learning, 22 (1): 99-113, 2006.

BANDURA, A. **Self-efficacy mechanism in human agency**. American Psychologist, 37(2): 122-147, 1982.

BECKER, T., RANDALL, D., RIEGEL, C. **The multidimensional view of commitment and the theory of reasoned action: a comparative evaluation**. JOM, 21(4): 617-638, 1995.

BLANK, M., HENNESSY, M. **A reasoned action approach to hiv prevention for persons with serious mental illness**. The ANNALS of the American Academy of Political and Social Science, 640: 173, 2012.

BODEN, A., MÜLLER, C., NETT, B. **Conducting a business ethnography in global software development projects of small German enterprises**. IST-53(9): 1012-1021, 2011.

BOYATZIS, R., **Transforming qualitative information: thematic analysis and code development**. Sage, 1998.

CHARMAZ, K. **Constructing grounded theory: a practical guide through qualitative analysis**, USA: Sage, 2006.

CHEN, G., KANFER, R. **Toward a Systems Theory of motivated behavior in work teams.** Research in Organizational Behavior, 27: 223–267, 2006.

CHEN, G., WEBBER, S., BLIESE, P., MATHIEU, J., PAYNE, S., ZACCARO, S., BORN, D. **Simultaneous examination of the antecedents and consequences of efficacy beliefs at multiple levels of analysis**. Human Performance, 15:381–409, 2002.

COHEN, S. **Designing effective self-managing work teams**. Proceedings of Theory Symposium on Self-Managed Work Teams, Denton, Texas, USA, June, 1993.

CRANG M., COOK I. **Doing ethnographies**, UK: Sage, 2007.

CRUZES, D., DYBÅ, T., RUNESON, P. HÖST, M. **Case Studies Synthesis: Brief Experience and Challenges for the Future**. Proceedings of ESEM'11, Banff-Alberta, Canada, September, 2011.

DAVIS, F. **Perceived usefulness, perceived ease of use, and user acceptance of information technology**. MIS Quarterly, 13(3): 319-340, 1989.

DAVIS, F.,VENKATESH, V. **A critical assessment of potential measurement biases in the technology acceptance model: three experiments**. International Journal of Human-Computer Studies, 45(1):19–45, 1996.

DESHON, R., KOZLOWSKI, S., SCHMIDT, A., MILNER, K., WEICHMANN, D. **A multiple goal, multilevel model of feedback effects on the regulation of individual and team performance in training**. Journal of Applied Psychology, 89:1035–1056, 2004.

DITTRICH. Y., RÖNKKÖ, K., ERIKSSON, J. **Cooperative method development: Combining qualitative empirical research with method, technique and process improvement**. EMSE 13(3): 231–260, 2008.

DOUGLAS, N. AND WYKOWSKI, T. **From Belief to Knowledge Achieving and Sustaining an Adaptive Culture in Organizations** , USA: CRC Press, 2010.

DUBÉ, L., ROBEY, D. **Software stories: Three cultural perspectives on the organizational context of software development practices**. Accounting Management and Information Technologies, 9(4): 223–259, 1999.

DYBÅ, T., KITCHENHAM, B., JØRGENSEN, M. **Evidence-based software engineering for practitioners**. In IEEE Software,1(22): 0740-7459, 2005.

EASTERBROOK S., SINGER J., STOREY, M., DAMIAN, D. **Selecting empirical methods for software engineering research**. Guide to Advanced Empirical Software Engineering, Section III, 285-311, 2008.

ERNEST, P. **The impact of beliefs on the teaching of mathematics. in mathematics teaching: the state of the art**. USA: The Falmer Press, 1989.

FÆGRI T.E., DYBÅ T., M. DINGSØYR, T. **Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work**. IST 52(10): 1118–1132, 2010.

FETTERMAN D. **Ethnography: step-by-step**, 3rd edition, USA: Sage, 2010.

FISHBEIN, M., AJZEN, I. **Belief, attitude, intention, and behavior: an introduction to theory and research**. MA: Addison-Wesley, 1975.

FORSEY M.G. **Ethnography as participant listening**. Ethnography - 11(4): 558–572, 2010.

FUNDA SAVASCI-ACIKALIN. **Teacher beliefs and practice in science education**. Asia-Pacific Forum on Science Learning and Teaching 10(1), article 12, June, 2009.

GEORGES, R., JONES, M. **Folkloristics: an introduction**. Bloomington: Indiana University Press, 1995.

GHORAB, K. **The impact of technology acceptance considerations on system usage, and adopted level of technological sophistication: an empirical investigation**. International Journal of information Management 17(4): 249-259, 1997.

HAINES, A., DONALD, A. **Getting research findings into practice**, 2nd edition, UK: BMJ Books, 2002.

HAMMERSLEY, M. **Ethnography: problems and prospects**, Ethnography and Education, 1(1):3-14, 2006.

HENNESSY, M. **Advancing Reasoned Action Theory** (The ANNALS of the American Academy of Political and Social Science Series), USA: Sage, 2012.

HOLDEN, R., KARSH, B. **The Technology Acceptance Model: its past and its future in health care**. Journal of Biomedical Informatics, 43(1): 159–172, 2010.

IIVARI, J., HUISMAN, M. **The relationship between organizational culture and the deployment of systems development methodologies**. MIS Quarterly, 31(1): 35-58, 2007.

IIVARI, J., IIVARI, N. **The relationship between organizational culture and the deployment of agile methods**. Journal of Information and Software Technology, 5(53):509–520, 2011.

JAN, A., CONTRERAS, V. **Technology acceptance model for the use of information technology in universities**. Journal of Computers in Human Behavior, 27(2): 845–851, 2011.

KARN, J.S., COWLING, A.J. **Using ethnographic methods to carry out human factors research in software engineering**. Behavior Research Methods Journal.BRM-38(3), 495-503, 2006.

KERIMOGLU, O., BASOGLU, N., DAIM, T. **Organizational adoption of information technologies: case of enterprise resource planning systems**. Journal of High Technology Management Research 19(1): 21–35, 2008.

KIRKMAN, B., ROSEN, B. **Beyond self-management: antecedents and consequences of team empowerment**. The Academy of Management Journal, 42(1): 58-74, 1999.

KITCHENHAM, B., DYBÅ, T., JØRGENSEN, M. **Evidence-based software engineering**. Proceedings of ICSE'04, Scotland, UK, 23-28 May, 2004.

KRAFT, P., RISE J., SUTTON, S., RØSAMB, E. **Perceived difficulty in the theory of planned behaviour: perceived behavioural control or affective attitude?**. British Journal of Social Psychology, 44(3): 479–496, 2005.

KYM, H., PARK, W. **The effect of cultural fit/misfit on the productivity and turnover of IS personnel**. Proceedings of SIGCPR '92, USA, 1992.

LASSITER, L. **Collaborative ethnography and public anthropology**. Current Anthropology - 46(1):83-106, 2005.

LEIDNER, D, KATWORTH, T., ALAVI, M. **The Role of Culture in Knowledge Management: A Case Study of Two Global Firms**. International Journal of e-Collaboration, 2(1): 17-40, 2006.

LESEURE, M., BAUER, J., BIRDI, K., NEELY, A., DENYER, D. **Adoption of promising practices: a systematic review of the evidence**. International Journal of Management Reviews, 5/6(3/4): 169–190, 2004.

LEWIS S., RUSSEL, A. **Being embedded: A way forward for ethnographic research**. Ethnography - 12(3): 398-416, 2011.

LIMAYEM, M., HIRT, S., CHEUNG, C. **How habit limits the predictive power of intention: the case of information systems continuance**. MIS Quarterly, 31(4): 705-737, 2007.

LUTTERS, W.G., SEAMAN, C.B. **Revealing actual documentation usage in software maintenance through war stories**. Journal of Information and Software Technology, 1(49):576–587, 2007.

MANSOUR, N. **Science teachers' beliefs and practices: issues, implications and research agenda**. International Journal of Environmental & Science Education, 4(1): 25-48, 2009.

MARTIN, J. **Cultures in Organizations: Three Perspectives**. New York: Oxford University Press, 1992.

MIDDLESTADT, S. **Beliefs underlying eating better and moving more: lessons learned from comparative salient belief elicitations with adults and youths**. The ANNALS of the American Academy of Political and Social Science, 640: 81, 2012.

MILES M., HUBERMAN, A. **Qualitative data analysis: an expanded sourcebook**, 2nd edition, USA: Sage, 1994.

MONTANO, D., TAPLIN, S.**A Test of an Expanded Theory of Reasoned Action to Predict Mammography Participation**. Social Science & Medicine Journal, 32(6): 733-741, 1991.

OMAR, M., SYED-ABDULLAH, S., YASIN, A. **The impact of agile approach on software engineering teams**. American Journal of Economics and Business Administration, 3(1): 12-17, 2011.

PASSOS, C., BRAUN, P., CRUZES, D., MENDONÇA, M. **Analyzing the impact of beliefs in software project practices**. Proceedings of ESEM'11, Banff-Alberta, Canada, September, 2011.

PASSOS, C., CRUZES, D., DYBÅ, T., MENDONÇA, M. **Challenges of applying ethnography to study software practices**. Proceedings of ESEM'12, Lund, Sweden, September, 2012.

PASSOS, C., CRUZES, D., MENDONÇA M. **Applying Theory of Reasoned Action in the Context of Software Development Practices: Insights into Team Intention and Behavior**. Proceedings of EASE'13, Porto de Galinhas, Brazil, 2013a.

PASSOS, C., CRUZES, D., MENDONÇA M. **Beliefs Underlying Teams Intention and Practice: An Application of the Theory of Planned Behavior**. Proceedings of ESELAW'13, Montevideo, Uruguay, 2013b.

PASSOS, C., CRUZES, D., HAYNE, A., MENDONÇA, M. **Recommendations to the adoption of new software practices: a case study of team intention and behavior in three software companies**. Proceedings of ESEM'13, Baltimore, Maryland, USA, October, 2013c.

PESLAK, A., CECCUCCI, W., SENDALL, P. **An empirical study of social networking behavior using theory of reasoned action**. Proceedings of CONISAR'11, North Carolina, USA, November, 2011.

PETERSEN, K., WOHLIN, C. **Context in industrial software engineering research**. Proceedings of ESEM'09, Orlando, Florida, USA, October, 2009.

PFLEEGER, S. **Understanding and improving technology transfer in software engineering**. Journal of Systems and Software, 47, 111-124, 1999.

RANDOLPH, M., PINKERTON, S., SOMLAI, A., KELLY, J., MCAULIFFE, T., GIBSON, R., HACKL, K. **Seriously mentally Ill women's safer sex behaviors and the theory of reasoned action**. Journal of Health Education and Behavior, 36(5): 948-958, 2009.

REEVES, S., KUPER, A., HODGES. B. **Qualitative research methodologies: ethnography**. BMJ, 337(10), 1336-1020, 2008.

ROBERTO, A., KRIEGER, J., KATZ, M., GOEI, R., JAIN, P. **Predicting pediatricians' communication with parents about the human papillomavirus (HPV) vaccine: an application of the theory of reasoned action**. Journal of Health Communication, 26(4): 303-312, 2011.

ROBINSON, H., SEGAL, J., SHARP, H. **Ethnographically-informed empirical studies of software practice**. IST-1(49): 540–551, 2007.

ROBINSON, H., SHARP, H. **Organisational culture and XP: three case studies**. Proceedings of ADC'05, Denver, Colorado, USA, July, 2005.

ROBSON, C. **Real world research : a resource for users of social research methods in applied settings**, 3rd ed. USA: Wiley, 2011.

RUNESON P., HOST M. **Guidelines for conducting and reporting case study research in software engineering**. Empirical Software Engineering, 2(14):131–164, 2008.

SANTOS, P. TRAVASSOS, G. **Action research can swing the balance in experimental software engineering**. Advances in Computers, v. 83, p. 205-276, 2011.

SCHEIN, E. **Organizational culture and leadership**. 2nd edition, USA: Jossey-Bass Publishers, 1992.

SHARP, H.,SOUZA, C., DITTRICH, Y. **Using ethnographic methods in software engineering research**. Proc. of ICSE'10, Cape Town, South Africa, May, 2010.

SHEPPARD, B., HARTWICK, J., WARSHAW, P. **The theory of reasoned action: a meta-analysis of past research with recommendations for modifications and future research**. Journal of Consumer Research 15(3): 325-343, 1988.

SIGFRIDSSON A., SHEEHAN, A. **On qualitative methodologies and dispersed communities: Reflections on the process of investigating an open source community**. IST- 53(9): 981–993, 2001.

SJØBERG D., DYBÅ T., JØRGENSEN M. **The future of empirical methods in software engineering research**. In Future of Software Engineering, IEEE Press, pp.358–378, 2007.

SMIRCICH, L. **Concepts of culture and organizational analysis**. Administrative Science Quartely, 28(3): 339-358, 1983.

STRODE, D., HUFF, S., TRETIAKOV, A. **The impact of organizational culture on agile method use**. Proceedings of HICSS'09, Waikoloa, Big Island, Hawaii, January, 2009.

SUTTON, S., MCVEY, D., GLANZ, A.**A comparative test of the theory of reasoned action and the theory of planned behavior in the prediction of condom use intentions in a national sample of English young people**. Journal of Health Psychology, 18(1): 72-81, 1999.

SUTTON, S., FRENCH, D., HENNINGS, S., MITCHELL, J., WAREHAM, N., GRIFFIN, N., HARDEMAN, W., KINMONTH, K. **Eliciting salient beliefs in research on the theory of planned behaviour: the effect of question wording**. Current Psychology: Developmental, Learning, Personality, Social, 22(3): 234–251, 2003.

TOLFO, C., WAZLAWICK, R. **The influence of organizational culture on the adoption of extreme programming**. Journal of Systems and Software, Journal of Information and Software Technology, 11(81): 1955-1967, 2008.

TOLFO, C., WAZLAWICK, R., FERREIRA, M., FORCELLINI, F. **Agile methods and organizational culture: reflections about cultural levels**. Journal of Software Maintenance and Evolution. 23(6): 423-441, 2011.

VANLANDINGHAM, M., SUPRASERT, S., GRANDJEAN, N., SITTITRAI, W. **Two views of risky sexual practices among northern thai males: the health belief model and the theory of reasoned action**. Journal of Health and Social Behavior, 36(2): 195-212, 1995.

VENKATESH, V., DAVIS, F.**A theoretical extension of the technology acceptance model: four longitudinal field studies**. Management Science 46(2): 186–204, 2000.

WERNICK, P., HALL, T. **Can Thomas Kuhn's paradigms help us understand software engineering?**. European Journal of Information Systems, 13 (3): 235-243, 2004.

WHITE, L. **The concept of culture**. American Anthropologist, 61(2): 227-251, 1959.

YZER, M. **Perceived behavioral control in reasoned action theory: a dual-aspect interpretation.** The ANNALS of the American Academy of Political and Social Science, 640: 101, 2012.

APPENDICES

Appendix A

This appendix presents the instrument used to map the companies and project contexts.

CONTEXT MAPPING INSTRUMENT

Company Context: ID: Company Name: Background: Creation: Location: Segment: Services Offer: Annual Revenue: Number of Employees: Certifications: Standard Process:	**Project Context:** ID: Project Name: Main Goal: Domain Customer : Origin: Customer Type: Customer Segment: Start Date: Size (PF): Productivity(H/PF): Time: Technology Platform: Methodology: Tools: Metrics:
Project Team Context: Member Name: Role: Role Description: Seniority: Experience Time: Competences: Mini-CV:	

Appendix B

This appendix presents the instrument that established specific goals for our observation task and some guided questions to drive the researchers' work.

OBSERVATION INSTRUMENT

Specific Goals to Record:

- personal values, paradigms, and folktales of the project team members.
- interactions between project team members.
- activities actually performed by the project team.
- practices adopted by members of the project team.
- impact of the practices of team members on the project.
- arguments for and against a particular practice.
- influence of a practice on another practice in the project.
- events and exceptions that happened during the work of project team.
- tools used by the project team members.
- time spent on activities performed by project team.
- metrics that show the practices adopted by the team.
- other phenomena relevant to the research.

Guided questions:

1. What practices are actually adopted by the project team?
2. How often these practices are adopted by the project team?
3. Are these practices being adopted properly by the project team?
4. Which profile adopts certain practices within the project?
5. How to understand the beliefs behind the practices in the project team?
6. How to understand the meaning of beliefs for the team?
7. How to understand the origin of beliefs in the team?
8. How to perceive the impact of beliefs on the project?
9. How to understand the influence of beliefs within the project?
10. How to characterize the beliefs behind the practices in the project team?
11. How to track the impact of beliefs on the project? (history, timeline, context, etc.)

Appendix C

This appendix presents the questionnaires used for the interviews.

INTERVIEW QUESTIONNAIRES

QUESTIONNAIRE 1

We present below the final version of the questionnaire used in the first cycle of the research in Company 1.

Warm-up questions:

1. What methodologies, software architectures, technologies, application domain, and types of client and size of projects have you worked with?
2. What are the main challenges of your current project?

Past experience questions:

3. Could you cite a past experience where the absence or presence of well defined work process impacted (positively or negatively) the project's progress?
4. Could you cite a past experience where the absence or presence of risk and communication plans impacted (positively or negatively) the project's progress?
5. Could you cite a past experience where failures in software engineering practices impacted the project's progress?
6. Could you cite a past experience where the absence or presence of project monitoring process by metrics impacted (positively or negatively) the project's progress?

Lessons learned questions:

7. Could you tell me different and similar practices adopted in past projects and in the current project?
8. What best practices related to your expertise and experience were useful to apply to your current project? Could you tell me about their application?
9. What situation or risk have you tried to prevent or mitigate in the current project on account of experiences already lived? Cite and comment.

Reaction questions:

10. In what aspects is the software development methodology of the company impacting your project?
11. Have any unexpected effect or impact happened after your current project started to use this methodology? Cite and explain.

Metric-based questions:

12. What affects, positively and negatively, the productivity of your project? Cite and explain.
13. What affects, positively and negatively, the quality of your project? Cite and explain.
14. What affects, positively and negatively, the time schedule of your project? Cite and explain.
15. What affects, positively and negatively, the cost of your project? Cite and explain.

QUESTIONNAIRE 2

The questionnaire below was used during the first phase of the second cycle of the research in Company 1.

Warm-up questions:

1. What do you know about self-management team practice?
2. Have you ever worked within a self-management team before?

Past experience questions:

3. Could you cite a past experience related to self-management team practice?
4. In your point of view, was this experience positive or negative? Please comment.
5. In your opinion, what are the main challenges to implement self-management team practice?

Lessons learned questions:

6. What good practices related to your expertise and experience in self-management teams were useful to apply to your current project? Please, tell me about their application. (successful or not)
7. Could you tell me about some self-management practice adopted in past projects and not adopted in your current project? Cite and explain.
8. what situation or risk have you tried to prevent or mitigate in the current project taking on account of experiences already lived with shared leadership and collective decision-making practices? Cite and comment.

Reaction questions:

9. How is the degree of autonomy in Scrum teams?
10. How is communication & collaboration practice in Scrum projects?
11. Are team members prepared with cross-functional skills to cover each other in a self-management team?
12. How is the organizational support for self-management team practices in the company?

13. In which aspects do you think team members need to improve to reach the self-management team benefits?

Metric-based questions:

14. Which project metrics do you think self-management team practice could affect positively or negatively? Cite and explain.

QUESTIONNAIRE 3

The questionnaire below was used during the second phase of the second cycle of the research in Companies 2, 3 and 4.

Warm-up questions:

1. In your opinion, what is the main challenge of your project? (productivity, quality, deadline, cost or other)

Past experience questions:

2. Could you cite a past experience where the project was conducted without planning, a defined schedule, without following a software development methodology and without risk analysis, and how this impacted (positively or negatively) the project results?

Lessons learned questions:

3. Is there any practice in your current project that is new to you? Do you think this new practice is beneficial to your project? Please, explain in what sense. And after using this new practice for a while, have you changed your opinion regarding its usefulness and importance?
4. Is there any practice in your current project (or past projects) that you adopted just because it was required to use it in the organization that you work, but you don't see its usefulness? Why do you not believe in this practice?
5. Is there any practice that you have introduced to your current project and that was not used in the organization in which you work? why did you think it was important to introduce this practice? And was there any problem with acceptance? What are the results of this practice in your current project?

Reaction questions:

6. Is there any new methodology that the organization in which you work is adopting? Do you know the reason for that change? Do you believe in this new methodology? Why? In what aspects is this company's new methodology affecting your current project?

Appendix D

This appendix presents a summary of the results of the methodological analysis about the benefits of using a collaborative ethnographic approach and the key challenges of applying ethnography to study software practices.

METHODOLOGICAL ANALYSIS

CHALLENGES OF APPLYING ETHNOGRAPHY

During the first case study we identified five key challenges of using ethnographic methods to study software practices. Some of these challenges have also been discussed in the social science literature and in SE literature. We related our findings to relevant ethnographic theory in order to support others who wish to conduct this type of study, making the results more organized as follows.

1) Collaborative Ethnography:

DESCRIPTION	LESSONS LEARNED
• C**ooperative** and P**articipatory perspective** to ethnography. • Work in **collaboration** with software organizations. • Have **something to offer** to the companies.	• Engage in a **knowledge exchange** with the practitioners. • Give **feedback to the subjects**, so they can **improve** work practices. • Build a **mutually beneficial research program**.

2) Insider-Outsider Dynamic:

DESCRIPTION	LESSONS LEARNED
• A required **insider** and an **outsider** view at the **same time**. • Ethnographers **stand in-between**: • research **subjects** and **context**; • projects **demands**; and • the **ethnographic product**.	• Combine **participation** with **observation**, so: • Get an **inside posture** mixed with an **outsider perspective**; • Do not move from the role of **researcher** to the role o **advocate**. • Avoid to get **over-involved** to **not disturb** the natural setting.

3) Observation and Listening:

DESCRIPTION	LESSONS LEARNED
• The balance between participant **observation** X **listening** • **Engaged Listening** involving: • **interviews**; • **informal conversations**	• **Participant Listening** alongside with **Participant Observation**: • Once you have identified the issues by **interviews**, you can **observe** the actions upon them. • The **interviews** allowed a **deeper understanding** of the obtained data through **direct observations**.

4) Relationship with the Participants:

DESCRIPTION	LESSONS LEARNED
• The relationship **Researchers** and **Participants** in an ethnographical work is a **core element** for **successful research**. • People in a **relationship** tend to **influence** each other.	• Maintain a **non-judgmental orientation** in order to avoid **bias**. • Manage the **tension** involved in moving between 2 **roles** (**researcher** and **practitioner**). • Minimize your **effect** on the **behavior of the subjects** under study, remaining in our roles.

5) Rigor and Contextualization:

DESCRIPTION	LESSONS LEARNED
• The **Rigor in Qualitative Work** means: • **Contextualization** need to be **broad** and **detailed** enough. • **Rigorousness** is essential in order to avoid **bias**.	• Conduct our ethnographic study with an **underlying theory**. • Look for **disproving** instances. • Seek **behavioral confirmation**, not just what people **say**. • Position yourself as the **main instrument** of the ethnographic study. • Use **tools** to deal with: • **substantial amount** of data; • distinct **levels of detail**; • **several iterations**.