Bringing Research Evidence into Software Industry Practice

A Study on Evidence-based Practice in the Software Industrial Setting

Carol Passos
DCC-UFBA
Salvador, Brazil
mcarolpassos@dcc.ufba.br

Abstract—Research findings can influence decisions at many levels, but only if one knows how to translate them into actionable and practical information. One of the major concerns of empirical software engineering research is in making a real impact on software engineering practice, bridging the gap between academia and industry. The area calls for effective approaches to the dissemination of empirical evidence into practice and there are no simple answers for this problem. In this context, the main objective of this proposal is to develop a methodology to increase the influence of empirical results on software practices and project team behavior. The main contribution is to deepen relevant knowledge and experience on how to expand the use of empirical results in the software development process, including the selection and use of technologies and practices in software industry context.

Keywords – belief; technical folklore; organizational culture; team practices; evidence-based and empirical software engineering.

I. INTRODUCTION

Empirical studies have been used to provide confidence in assertions about what is true and not true in the software engineering domain [32]. Nowadays, the academic community is concerned with addressing the question of how empirical results can influence the community in practical terms [17]. The area calls for effective approaches to the dissemination of empirical evidence into practice and there are no simple answers for this problem [4]. A theoretical basis for this process through formal experimentation needs to be developed [17]. Practitioners can have difficulty to make informed decisions about whether to adopt a new technology because there is little objective evidence to confirm its suitability, limits, qualities, costs, and inherent risks [10]. Busy professionals need to access relevant and reliable knowledge easily. They also need to trust and believe this information. Although the people involved in research have worked hard to produce empirical evidence that is useful to the software community, it is still difficult to penetrate the business culture and transfer these findings to practitioners. Researchers still struggle for guidelines on how to get their findings into practice, positively impacting business development. They have theorized the application of a cultural perspective to understand software implementation and use, but few have applied this to the real-world workplace itself [13].

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 that evidence-based software engineering can provide the mechanisms needed to assist practitioners to adopt appropriate technologies and to avoid inappropriate technologies [19].

Much like societal cultures, beliefs and values have manifested in norms that form behavioral expectations; companies have cultures that form and give members guidelines for the way to do things around here – an organizational culture and belief system [30]. This phenomenon must be studied so one can better analyze and understand how beliefs can impact group practices.

Belief is the psychological state in which an individual holds a proposition or premise to be true. Beliefs could be defined as conceptions, personal ideologies, and perceptions of the world and values that shape practice and orient knowledge. The concept of belief implies the existence of an individual and mental state with intentionality, interacting with goals and influencing ordinary actions [1].

Folklore consists of legends, music, oral history, proverbs, jokes, fairy tales and customs that are the traditions of some culture or group of people or community. Folklore can also serve to validate a culture, as well as to transmit a culture’s morals and values [15]. Importantly, folklore (technical or not) are dependent on surrounding context – culture, region, community or organization – and, at first, it does not have formal trial basis; its adoption often leads to context and validity problems.

Folklore and beliefs are strong in the software engineering community [37] due to the habitual use of mailing lists, social networks and communities of practice as a way of fast and effective knowledge transfer in the area. The process of establishing technical folklore and beliefs is the mode of operating of the software community (mode of thinking and acting). The use of knowledge in software engineering (SE) is also context dependent, so software practitioners need to get the right information at the right time and contextualize it to their needs.

When examining the practices of SE teams, the mindset of practitioners, when making a decision, can be very much influenced by the tools and techniques in use at that time by the software community. The underlying and implicit belief system is inevitably taken on board for the adoption of a new technology, for example. The differences in approach of software engineering practices and their actual benefits possibly obtained by changing from one technology to another...
are poorly understood [37]. This lack of understanding may be due in part to a failure to investigate, understand and document the nature and effect of the belief system underlying current SE theory and practice.

There are some scientific studies in other areas about the impact of beliefs on team or group practices, either individually or in combination and synthesis [14][1][3][23]. Research has documented that practitioners’ beliefs related to software process have a significant influence on their practices. This influence on practices plays out in interesting ways. For instance, Kim and Park [20] reported that there is a connection between a belief system and an organizational culture in the software engineering context; while Wernick and Hall [37] provided an understanding of the common core of beliefs underlying SE practices.

However, the majority of existing empirical findings about beliefs/knowledge and practices/attitudes are not suitable to guide the industry in the best possible new technology introduction. Studies we found are on other areas, like educational [23][18] or clinical medical [17][5][2]. Therefore, there is a need to understand how each practice relates or is influenced by beliefs and how these relationships affect software project results.

This document presents a research plan to characterize the use of folklore-based and evidence-based practices in the software industry. The project presented here intends to characterize how beliefs are incorporated into practices and how can empirical results be incorporated as useful beliefs in software organizations. Our research plan involves qualitative research to investigate the use and impact of beliefs on software development practices. Our main strategy is to develop case studies to understand how the phenomena occur in practice. Participant observation, interviews and ethnography will be used to develop a ground theory on belief usage and impact in the software industry.

This rest of this document is organized as follows: after a brief introduction, section 2 gives an overview of the background. Section 3 presents the research goals and expected contributions. Section 4 explains the research approach, the methodology adopted and project phases, while section 5 presents the data analysis methods used. In section 6, some validity issues are addressed. A discussion of issues on which we would like to get the most advice on and further work are presented in the section 7. And finally, references are listed in the last section.

II. RELEVANT PRIOR WORK

This section provides an overview of the foundations for this work. The conceptual framework [24] of this research is shown in Fig. 1. 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-based practice represents the influence of knowledge from empirical research into team practice. This influence is based on empirical evidence and also affects beliefs and perceptions about SE practice.

Thus, some related works and more relevant studies are presented below.

A. Technical Folklore, Organizational Culture and Project Context

This section intends to analyze the aspects that influence beliefs under the perspective of the software engineering community folklore, the culture of software organizations and the software projects contexts.

Like any other scientific community [37][37], SE community is characterized by the unified thinking of its members. They have similar education and professional initiations. In this process they have absorbed the same technical folklore and drawn many of the same lessons from it. Due to the open communication among the community members, they can easily share assumptions, beliefs, paradigms and approaches. This phenomenon can foster their relationships, increasing the existing folklore as much as helping in creating new ones.

The process of establishing technical folklore is the mode of operating of the software community. It can transcend national, regional or organizational cultures. Software developers applying SE practices are influenced by community’s folklore that is inevitably taken on board in the adoption of a particular technology or practice. This is sustained by a culture that permanently anticipates technological advances and solutions.

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 [30]. So, culture is the learned result of group experiences.
The significance of culture is particularly strong in the corporate world, where constantly new assumptions and beliefs are created, discovered or developed by a given group [30]. This is related to the institutionalized way of thinking and acting of people in a company.

The essence of a company is expressed by the way it does business, the way it deals with customers and employees, and the level of autonomy it grants to their members [35]. Organizational culture is manifest as an informal and hidden force, which arises over time during a company’s history. This force exerts tremendous influence on the models of behavior of its employees and how they act and justify their actions [20].

Thus, organizational culture forms the context in which the 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, behavior, attitudes and actions. The project context should be placed in the center of all relevant software project discussions [26]. We want to highlight it in order to better analyze and understand how beliefs can impact group practices.

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, a comprehensive description of the project context has great relevance to characterize how each practice relates or is influenced by beliefs and how these relationships affect software project results.

Within the scope of this research, the technology transfer process is the process through which an organization, group or team selects and decides to use a set of technologies and practices in a context of a software industry project. Practitioners are faced regularly with technology decisions and are guided by aspects of efficiency and effectiveness. On the other hand, researchers must evaluate the effects of a technology in a particular context and build a body of evidence to help this decision process and to predict if the use of technology will lead to real improvement. Therefore, it is known that compelling evidence is not enough to ensure technology transfer. In addition, the technology must be packaged and supported so as to make it friendlier and easier to understand and use [27]. In this scenario, it becomes more important to understand the beliefs of people involved in the transfer process, how they are incorporated in a software project context and how empirical results can influence them.

There are important empirical studies in the software engineering area that consider the influence of organizational culture on the technology transfer process [20][13][21][31][35]. Our intention is to expand this knowledge through an approach that allows the capture and representation of beliefs and how they impact practices, processes and decisions in software industry projects.

The development of an organization requires a deep understanding of the overall dynamics of processes and activities. So, these new insights can provide guidance to help companies to improve their processes and practices.

B. Evidenced-Based Practice

We also draw upon research devoted to evidenced-based practice, because of our assumption that there should be closer links between research and practice, so that research could be more relevant to practitioners’ needs and practitioners will be willing to participate in research projects. Better approaches of involving practitioners in research projects are needed to ensure that research questions are appropriately framed and tested in relevant contexts, using mechanisms that can be replicated in the conditions of day-to-day practice [17].

Research evidence can be interpreted and used in a variety of ways and should play a more vital role in project team practices, since it can provide a reference against which decisions and choices may be evaluated. 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 [10].

Evidence-based practices can define how SE knowledge can be implemented. According to Haines and Donald [17], there are four main elements necessary for the successful implementation of knowledge from empirical research into practice: (i) the research findings packaged in a digestible form; (ii) a credible dissemination body containing influential members; (iii) a supportive practice environment, including, in this case, tools and other packaging and support mechanisms to aid the technology’s use; and (iv) the concept of local knowledge – the local practices, values, and beliefs into which new knowledge must usually be integrated.

When it becomes common practice, companies will have successfully undertaken evidence-based practice, from particular issues to the development of guidelines and improvement of local practices. Thus, software organizations will be able to adopt best practices more quickly and with fewer risks, improve the quality of products, and reduce the risk of project failures [10].

Regarding this background, there is space for new initiatives that can contribute to increase the influence of empirical findings on SE practices through identification of their relationships and mapping of the origins and sources of beliefs in this context.

III. RESEARCH OBJECTIVES

The main goal of this project is to contribute to the expansion of the use of empirical results in the software development process and use of technologies and practices in the industrial context. We expect to produce innovative results by examining the significant relationships between empirical results and software engineering practices and project team behavior using a qualitative approach. To achieve this goal, we will seek to understand the impact of beliefs on software engineering team practices, and their implications in software projects.

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

- **RQ1**: What is the impact of technical folklore and beliefs on software engineering team adoption, ac-
ceptance, inception and use of technologies and practices?

- **RQ2**: How are beliefs incorporated into practices in software organizations?
- **RQ3**: How can empirical results be incorporated as useful beliefs in software organizations?

In order to answer question RQ1 it is necessary to first understand and find ways to identify and capture technical beliefs of the software engineering area, their origin and sources and how they actually influence the industry software projects. We expect to bring forth empirical evidence about the impact of beliefs on software engineering practices and work processes, contributing to a better understanding of the challenges of the software industry and providing evidence of interest to academic research.

Upon obtaining qualitative evidence that explain how beliefs influence software industry projects, it will be possible to answer RQ2. This will be done through the study and measurement of software engineering practices that are incorporated as beliefs in project teams. In this context, it is also important to understand the process of building relationships of beliefs.

From the identification and characterization of significant relationships between empirical results and software engineering practices, it is possible to investigate the answer to the last question (RQ3). It will be also necessary to point to the right time and to know how to introduce empirical results in software projects context. In this scenario, we intend to present a methodological proposal to increase influence of empirical results in software practices.

Thus, these three research questions lead to new knowledge and evidence that will be important to help and support software industry processes and academic environment. At the end of the research, we expected the following results:

- Identification and characterization of significant relationships between technical folklore/beliefs and team practices in software engineering organizations.
- Representation of beliefs and how they actually impact practices, processes and decisions in software industry projects.
- Guidelines to improve team practices that are influenced by technical beliefs in the software engineering processes.
- Insights on how to increase the influence of empirical results on software engineering practices presented as a methodology.

### IV. Research Approach

In this project, the main idea involves performing a qualitative research of the impact of beliefs on software development practices, developing secondary case studies from project software tracking, such as primary case studies, toward an understanding and knowledge of beliefs and phenomena related to team practices like represented in Fig. 2.

This research has an essentially qualitative design, where the central research method is case study. The data collection has involved interviews, observations and documentary analysis, according to [29][11][16]. To focus and bound the collection of data, a conceptual framework was built (Fig. 1). It explains the main aspects to be studied, their key factors, constructs, variables and relationships among them in order to arrive at a balanced and comprehensive understanding of beliefs and their impact on team or group practices. It is a direct step from conceptual framework to research questions design. Conceptual frameworks also can be particularly useful in cross-case analysis [24].

In selecting a particular instrument, it is important to clearly know its strengths and weakness to reason about its fit within a case study design. After a literature review and an evaluation of research objectives and questions, we have opted for one interview-based qualitative data collection technique called *storytelling* and we used a specific story form *the war story* [22].

**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 [25]. 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 important in this sort of research.

We plan to use an iterative approach in which we define a questionnaire, used it for a set of interviews, analyze the data, and improve the instrument and process for the next round of interviews. The first step of the cycle is therefore to define the interview questionnaire. We tried to keep it as simple as possible and still cover the typical questions of a war story questionnaire.

War story questionnaires usually have warm-up, past experiences, lessons learned and reaction questions. We have a few of each type.

- **Warm-up questions** aim to put the interviewee on the right set of mind, to focus them on the interview subject. Our first version of the questionnaire has two warm-up questions that ask about the participants’ background and experience, present position, and the main challenges of the 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 has four of these questions. They ask about the respondents’ past experiences, involving software process, communication plan, software engineering technical practices and project monitoring.
- **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 have three war story questions about lessons learned. They aim to understand the basis of how beliefs, personal values, and paradigms of the participants came to be and also how they determined the participants’ choices and actions during the project.
- **Reaction questions** ask about the participants’ reactions and personal opinions on the impacts of the use of technolo-
gies, methods and process on the software project. The questionnaires have two reaction questions that directly ask about project impacts of new technologies and unexpected effects of known and new technologies.

Besides the typical war story questions, we added a few questions to identify beliefs related to practices that affected project productivity, quality, time and cost. Those attributes are usually quantitatively measured in software organizations, so they help to link quantitative and qualitative findings (Metric-based Questions). We have four of them in the first version of the questionnaire.

As described, our war story questionnaire goes beyond asking questions that allow the respondents to generalize on their past experiences, it asks them to retell and revive specific and directed stories that illustrate the experiences we were trying to capture. The resulting data contains considerable amount of contextual information, which enables connections between different but related stories. The metric-based questions link quantitative with qualitative data obtained from the participants’ other responses. It helps to corroborate findings via triangulation with the organization metrics.

Following our iterative approach, we conducted four interviews to start exploring the research questions. These interviews were conducted with two people on the front lines of a project that introduced agile programming on an organization very experienced on traditional software process practices. Each person was interviewed twice, once at the beginning and once at the end of the project.

Sampling is critical for later analysis [24]. The choices made place limits on the conclusions and study results, which are connected directly to the research questions. We considered these four interviews a representative sampling in the context delimited by the three research questions previously defined.

These four interviews are being complemented by meeting observations and documental analysis. Up to know, we have attended and recorded 40 project meetings involving the participants interviewed. At this point we are analyzing all the information collected via cycles of coding – tagging key words, phrases, and paragraphs. After that, we will identify relationships and patterns in it.

The interview transcription and coding has being time consuming. We are spending an average of five hours of transcription and one hour of preliminary coding for each hour of interview recorded. The collected, transcribed and coded data have already produced some results and conclusions that will be presented in later publications.

While the data collection is still ongoing, we began early data analysis, because it helps data-gathering cycle back and permits the production of interim reports. Our intention is to anticipate what is going on, how things are proceeding and why things occur as they do and driving the work as grounded theory (Fig. 2). We expect that this will facilitate the understanding of this complex process by showing how their attributes can be related according to rules and practices and presenting them as a theory.

Considering the upcoming data analysis, we intend to employ a combined approach of meta-ethnography [25][5][2] and grounded theory [7][6] in a qualitative cross-case analysis [24] like showed in Fig. 2. The qualitative method approach makes it possible to uncover and understand how practices are actually applied by teams and how their beliefs impact the process.

As mentioned, case studies definition and design, data selection and gathering, cross-case qualitative analysis, results interpretation and discussion and, if necessary, development of metrics for evaluation and confirmation of results are part of the scope of this work. Another important part is a methodological proposal to increase the influence of empirical results on software engineering practices.

In the scope of this work, we are collecting data from three software projects with an average duration of 6 months each. We are conducting case studies mostly using direct and participant observation and semi-structured interviews for data collection and early analysis. It involves constant meetings and practice observation and the establishment of frequency and scope of the interviews based on project dynamics and methodology. Also, we will review project artifacts, including its quantitative metrics, for documentation analysis and to link qualitative findings with quantitative data.

This project has been developed within the doctoral research of the applicant, under the advice of Prof. Manoel Mendonça from The Federal University of Bahia (UFBA), and supervised by Dr. Daniela Cruzes from The Norwegian University of Science and Technology (NTNU).

A. Phases of the Project

The present project was planned in four phases, one of them to be explored during the research visit to NTNU in Norway. Table 1 shows the phases of this project.
TABLE I – PROJECT PHASES

<table>
<thead>
<tr>
<th>Activity</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Planning and Initial Start-up</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Phase I (RQ1) Case Studies on the origins and impact of</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical beliefs/ folklore in software development practices.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Phase II (RQ2) Case Studies on how technical beliefs/</td>
<td>X X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>folklore are incorporated in software development practices.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Phase III (RQ3) Development of framework to enhance the</td>
<td></td>
<td></td>
<td>X X</td>
<td>X X</td>
</tr>
<tr>
<td>use of empirical studies in industry.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Publications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case Studies on the origins and impact of technical beliefs/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>folklore in software development practices.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development of framework to enhance the use of empirical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>studies in industry.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Report and Finalization of the Project (Dissertation and</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Defense)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V. DATA ANALYSIS METHODS

Considering that qualitative research may be conducted in dozens of ways and there are a number of empirical qualitative methods available, researchers need to understand the goals underlying each method and possess some knowledge about alternatives to select the appropriate method to work. The choice of methods depends upon how closely the methods align with the research questions that have been posed. It is often necessary to use a combination of methods to fully perceive the research problem.

Because of the importance of human activities in software development, many of the research methods that are appropriate to software engineering are drawn from disciplines that study human behavior, both at the individual level and at the team and organizational levels [12]. Each method might fail differently. So, a viable research strategy can use multiple methods, chosen in such a way that the weaknesses of each method can be addressed by the use of complementary methods.

As for data analysis method, we chose a combination of three complementary qualitative approaches – meta-ethnography and grounded theory in a qualitative cross-case analysis. The cases are being treated like a series of independent empirical studies that confirms or not emerging conceptual theories. In each case, we will incorporate into the analysis captured sources of beliefs and their evident impact on software engineering practices. The case study method [29][11][16] has contributed to elucidate a variety of practices that allow us to understand how individuals interact in a software engineering team.

It has become quite common to adapt ethnographic traditions in order to conduct case studies. One precondition for an ethnographic study includes research questions that focus on cultural practices of a particular community [12]. For software engineering, ethnography can help to understand how technical communities build a culture of practices and communication strategies that enables them to perform technical work collaboratively. This method can provide a framework for studying the culture of the software engineering industry setting, and for uncovering knowledge, beliefs and values which affect activity.

Once the analysis of each case is completed, cross-case analysis begins. We are categorizing, tabulating and analyzing the individual cases as windows to compare and give insight for the cross-case analysis. In that analysis we will consider in what areas the cases suggest the same points, where they differ, and where the cases conflict. Until now the comparison of the cases brought no relevant difference, but we found that the cases complement one another, and additional observations arose.

As for development of secondary studies, we have decided to adopt a meta-ethnography approach [25][5][2]. The meta-ethnographic research approach is a valuable starting point for a consideration of the philosophical basis of interpretive case studies [36]. It compares and analyzes evidence, creating new interpretations in the process that studies can relate to one another in one of three ways: (i) they may be directly comparable as reciprocal translations; (ii) they may stand in opposition to one another as refutational translations; or (iii) taken together they may represent a line of argument.

We have studied practice by immersing ourselves in the area under study for several months, documenting what takes place via a range of means that include field notes, audio recordings of discussions and meetings, access to various documents and artifacts, and records of interviews with practitioners. Through the synthesis of all these evidence, we intend to make sense of what they are saying and to provide interpretative explanations about them.

Our next step is to include the grounded theory approach [7][6] to form initial insights inductively on the basis of multiple case study data. During this analysis by the grounded-theoretical means of constantly iterating between data, literature and emergent theoretical framework, we will be able to reach a more complete understanding of software engineering practices in a project team context that are influenced by technical folklore of the community and beliefs of the individuals. This approach is suitable since we were studying the impact of beliefs on team or group practices for which existing theories appeared to provide limited answers until now.

VI. VALIDITY THREATS

This section discusses threats to the validity of our approach. In our case studies, we are collecting data by interviews, observations and analysis of documentation relevant to practice.
We have some conclusion validity threats as we are working with a small sample (three projects), limited number of interviewees (two per project), and interviews (two per interviewees). We address this issue by triangulating data from interviews, participant observation, and document analysis and focus groups. The data sources are reliable. We are working with real world project data and the participants chosen for the interviews are experienced project managers or technical leaders. They are capable of accurately and reliably answering the interview questions. The frequency and scope of this collection is based on the project dynamics and methodology. So, we believe we cover the most relevant project moments with our approach.

Regarding, internal validity, one factor that could affect the reliability of the interview answers is the time elapsed since the end of the project. So, we decided to interview participants at the beginning of the project and before its end to collect more reliable data and reduce information loss.

The relationship between researchers and our participants has been particularly significant in two respects. 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 have an active, but secondary role in the target projects, because we act as quality assurance and process control in the company investigated. Being a member of the community under study has both challenges and advantages for the researcher. The main advantage of being a member of the same culture as the researched revolves around acceptance and common vocabulary, so the participants can relax and focus on working in their natural way [28]. 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 interviews notes and observations with other researchers and with the participants.

External validity may be a problem, as we are currently working with just one organization. The participants are professionals using typical development technologies in a typical working environment. This makes the results more generalizable. Nonetheless, the three case studies’ environment is context specific, since the participants belong to the same company. They can introduce some bias due to the organizational culture. Time allowing, we intend to include other companies in the scope of this research.

It is known that interpretation and generalization of findings should be tempered by several considerations. The validity treatments of this research seek to improve the generalization of these findings.

VII. DISCUSSION AND FURTHER WORK

This section presents a summary of the current status of the research, planned next steps and describes the issues or points on which we would like to get the most advice on.

The research is now in the phase II of the project plan (section 4-A). We have conducted interviews with participants of the three projects in scope. We are compiling information of the first four interviews; the data are being analyzed via cycles of pattern coding. These coded data have already produced some results and conclusions and as far as we can see the resulting data contains considerable amount of useful information, especially for the organization itself. While the data collection is still ongoing, we have already begun data analysis, because it provides feedback on the data-gathering cycle and enables us to produce interim reports.

The next step is to do cross-case analysis. Through the synthesis of all this evidence, we intend to provide interpretative explanations about them using a meta-ethnography approach. This involves the use of the grounded theory approach to form initial insights inductively on the basis of multiple case study data (phase III).

Besides the validity issues and the treatment for them, we want to discuss points related to the scope and relevance of the research, the positioning of the work in the existing literature and data collection and analysis methods proposed.

The use of qualitative methods for SE brings advantages for the development of relevant theories and can also bring new knowledge to the academia for better understanding the phenomena in focus. This research intends to adopt a combined qualitative approach, which we expected to help in capturing realistic scenarios for the study. One advantage of our approach is to work with real world projects as primary case studies. Such studies can, therefore, be limited to a unique organization culture. Our project addresses this limitation by planning to include other companies in the scope of this research.

ACKNOWLEDGMENT

I would like to thank my supervisors Manoel Mendonça and Daniela S. Cruzes for their precious advices, and the reviewers of IDoESE for their comments.

The author and her advisors are grateful to all involved in this study, specially the interviewees for their insights and cooperation and to the SOLUTIS organization for supporting this work.

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES), funded by CNPq, grant 573964/2008-4.

REFERENCES


