Analyzing the Impact of Beliefs in Software Project Practices

Carol Passos
SOLUTIS Tecnologias
Salvador, Brazil

carol.passos@solutis.com.br

Ana Paula Braun
SOLUTIS Tecnologias
Salvador, Brazil

paula.braun@solutis.com.br

Daniela S. Cruzes
IDI-NTNU
Trondheim, Norway
dcruzes@idi.ntnu.no

Manoel Mendonça
DCC-UFBA
Salvador, Brazil

mgmendonca@dcc.ufba.br

Abstract—Folklore and beliefs are strong in the software practitioners’ community. Software engineering is a communication intensive activity. Software engineers are innovation driven and regularly use automated resources to share ideas, new paradigms and approaches to support and improve their practices. This information flow generates technical folklore and beliefs (that do not have a formal trial basis). Software engineers applying practices are influenced by these and they are inevitably taken on board in the adoption of a particular technology or practice. This paper presents an industrial case study, using a qualitative approach, to investigate the origins and impacts of beliefs on software development team practices. Its main contribution is on the understanding of creation and evolution of technical beliefs, and in studying its use for team practices improvement in the software engineering industry.

Keywords-belief; technical folklore and beliefs; industry case study; business culture and values; software practices.

I. INTRODUCTION

Empirical research seeks to explore, describe, predict, and explain natural, social, or cognitive phenomena from the real-world by using evidence based on observation or experience [5][38]. Software engineering (SE) practices and technologies can be formally evaluated in empirical studies to support their adoption by software organizations [22][29][28][30].

Unfortunately, formal experimentation is not wide spread in the software industry. Practitioners deal regularly with technology decisions, and do not have (or can access) trustworthy empirical evidence to support them. Most software development organizations have limited time and money for determining which of the available technologies will be most beneficial on 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[12].

Most of the practitioners rely on technical folklore and beliefs to make their decisions. In spite of not having a formal trial basis, these beliefs and technical folklore are built over a sophisticated set of relationships and communication channels.

Researchers are constantly trying to evaluate the effects of a technology in a defined context to build a body of evidence and to help the generalization of conclusions [30]. The introduction of a particular technology (process model, method, procedure, tool or technique) to common practice in a software engineering project requires a good grasp of the technology, generation and analysis of evidence, good packaging and support, and careful deliberation of the perspective user for the technology [29]. However, the definitive step towards the introduction of a particular technology into an organizational culture is to convince people of the usefulness of this technology 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? This process must be studied and understood by empirical software engineering researchers.

In this paper, we present an industrial case study to characterize a belief system of a project team, and the use of it to motivate the improvement of software development team practices. It involves a qualitative research approach to identify, capture and understand the origins, sources and impacts of beliefs in this context. We use ethnography for that, employing participant observation, interviews and documental analysis for data collection [20][31].

Formally, the main goal of this research was to characterize and understand the belief system of a project team and use it to improve its practices. The research questions were:

- **RQ1**: What are the origins and sources of software engineering beliefs?
- **RQ2**: How do these beliefs impact current software project practices?
- **RQ3**: What practices are being benefited and hindered by those beliefs?

We captured and analyzed this belief system and used it to improve practices and work processes. This sort of approach can contribute both to industry, as it provides new paths to technology adoption, and to the academia, as it improves the understanding on getting empirical evidence into practice.

This paper is organized as follows. Section 2 reviews related work. Section 3 presents the research approach. Section 4 describes the industrial case study context. Section 5 presents the case study execution and analysis. Section 6 presents its results and sections 7 and 8 discuss them. Conclusions and further work are presented in section 9.

II. RELATED WORK

This section provides a brief overview of the foundations for this work. Specifically, how organizational culture, folklore and beliefs and evidence-based approach influence team practices.
A. Folklore and Beliefs in Software Engineering

Beliefs are defined as conceptions, personal ideologies, worldviews 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. 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 [17].

Folklore and beliefs are dependent on surrounding context (culture, region, community or organization) and, at first, they do not have formal trial basis. They are strong in software engineering 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 basis of operation (mode of thinking and acting) of the software community.

There are some scientific studies in other areas about the impact of beliefs on team or group practices [16][1][4][24]. Practitioners’ beliefs related to software process have a significant influence on their practices. For instance, Kym and Park [21] reported that there is a strong connection between a belief system and an organizational culture in the software engineering context [41].

As the majority of existent empirical findings about beliefs/knowledge and practices/attitudes are not suitable to guide the industry in the best possible new technology or practice because they are focused in other areas, such as educational [24][20] or clinical medical [19][6][2][25]. Therefore, it is important to investigate how each practice relates or is influenced by beliefs and how these relationships affect software project results.

B. Organizational Culture and Project Context

Culture is the learned result of group experiences. It is defined as a pattern of shared basic assumptions that the group learned by solving problems of external adaptation and internal integration that 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 [33]. The significance of culture is particularly strong in the corporate world, where groups are constantly creating, discovering or developing new assumptions and beliefs [33].

The essence of a company is expressed by the way it does business, the way it deals with its customers and employees, and the level of autonomy it grants to its members [39]. Organizational culture manifests 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 [21].

Thereby, 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. 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.

There are some relevant empirical studies in the software engineering area that consider the influence of organizational culture on the technology transfer process [21][15][22][34][39]. Our intention is to expand this knowledge through an approach that allows the capture and representation of beliefs and analyze how they impact practices, processes and decisions in software industry projects.

C. Evidenced-Based Practice

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. However, practitioners can have difficulty making informed decisions about whether to adopt a new technology or practice because there is little objective and accessible evidence to confirm its suitability, limits, qualities, costs, and inherent risks [12].

Better approaches, involving practitioners, 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 daily practice. Evidence-based practice is able to provide evidence how software engineering practices can be effectively and efficiently used.

According to Haines and Donald [19], 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 [12].

Our intention is to use the experience of this case study to begin an introduction of evidence-based practices in the company in scope and do future expansions of this culture to other companies.

III. RESEARCH APPROACH

The main idea of our approach involves performing qualitative research of the impact of beliefs on software engineering practices, toward an understanding and knowledge of beliefs and phenomena related to team practices.

We performed interviews, observations and documental analysis, according to [32][13][18]. After a literature review and an evaluation of research objectives and questions, we have opted for one interview-based qualitative data collec-
tion technique called storytelling and we used a specific story form the war story [23]. We used an iterative approach in which we define a questionnaire, use 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.

War story questionnaires [23] usually have warm-up, past experiences, lessons learned and reaction questions. We have a few of each type (See appendix A).

**Warm-up questions** aims to put the interviewees on the right set of mind, to focus them on the interview subject. 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.

**Past experience questions** intend to investigate how living experiences can influence the participants‘ current behavior, covering the main aspects of software project reality.

**Lessons learned questions** try to capture the beliefs that emerged or evolved from the project experiences, and how this came to be. 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 technologies, methods and process on the software project.

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). These organization metrics are important to corroborate findings via triangulation. 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.

With the previous methods and data collection, we performed the analysis of the common and conflicting beliefs of the product owner and the SCRUM master in a project. After that, we conducted a focus group with the whole team and some other practitioners from the same company.

IV. **INDUSTRIAL CASE STUDY CONTEXT**

The case study described here is set in a medium sized software company¹, in which projects are applying for the first time agile methodology based on SCRUM² and PMI³ approaches. This organization has one development center with ISO-9001 certification and four offices in important regions of the same country. The company provides services on software development and evolution as well as information technology infrastructure management services for customers of public and private sectors.

The project under study is a software for electrical energy management and with a six months duration. The solution is a web portal system fronting a database with information and measures of electrical power consumption. The project team follows a well-defined process for software development and uses an open source project management application for stories and bug tracking system. They also keep detailed records of the project status reports and the backlog of stories and changes to the software. Other important documents, including quantitative measures are used for project tracking and monitoring. This practice is clearly influenced by past experiences of team members in a CMMI certification program (See tables 1 e 2 for more details).

### TABLE I. PROJECT PROFILE

<table>
<thead>
<tr>
<th>Attribute</th>
<th>PGE Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Size</td>
<td>5</td>
</tr>
<tr>
<td>Team Members</td>
<td>2 full time developers, 2 part time developers, 1 scrum master, 1 product owner + project manager</td>
</tr>
<tr>
<td>Language</td>
<td>JAVA</td>
</tr>
<tr>
<td>Tool Usage</td>
<td>Open Source Software</td>
</tr>
<tr>
<td>Development Methodology</td>
<td>SCRUM + PMI Approaches</td>
</tr>
<tr>
<td>Non-Functional Requirements</td>
<td>Reliability, Performance, Continuity, Availability</td>
</tr>
<tr>
<td>Reuse</td>
<td>High</td>
</tr>
<tr>
<td>Stability Requirements</td>
<td>Medium stability</td>
</tr>
<tr>
<td>Staff Turn-over</td>
<td>Considered high by the project manager</td>
</tr>
<tr>
<td>Interviews</td>
<td>2 with product owner + project manager, 2 with scrum master</td>
</tr>
</tbody>
</table>

### TABLE II. PROJECT AGILE PRACTICES

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>PGE Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code &amp; Tests</td>
<td>Full</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Full</td>
</tr>
<tr>
<td>Daily deployment</td>
<td>Full</td>
</tr>
<tr>
<td>Daily meeting</td>
<td>Full</td>
</tr>
<tr>
<td>Energized work</td>
<td>Do not use</td>
</tr>
<tr>
<td>Incremental design</td>
<td>Full</td>
</tr>
<tr>
<td>Pair programming</td>
<td>Do not use</td>
</tr>
<tr>
<td>Real customer involvement</td>
<td>Full</td>
</tr>
<tr>
<td>Shared Code</td>
<td>Full</td>
</tr>
<tr>
<td>Single code base</td>
<td>Full</td>
</tr>
<tr>
<td>Sit together</td>
<td>Full</td>
</tr>
<tr>
<td>TDD</td>
<td>Do not use</td>
</tr>
<tr>
<td>Ten minute build</td>
<td>Partial</td>
</tr>
<tr>
<td>Negotiated scope contract</td>
<td>Full</td>
</tr>
<tr>
<td>Planning game</td>
<td>Do not use</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>Full</td>
</tr>
<tr>
<td>Root cause analysis</td>
<td>Partial</td>
</tr>
<tr>
<td>Slack</td>
<td>Partial</td>
</tr>
<tr>
<td>Stories</td>
<td>Full</td>
</tr>
<tr>
<td>Team continuity</td>
<td>Partial</td>
</tr>
<tr>
<td>Weekly cycle</td>
<td>Full</td>
</tr>
<tr>
<td>Whole team</td>
<td>Full</td>
</tr>
</tbody>
</table>

² SCRUM approach: [http://www.scrum.org/](http://www.scrum.org/)
³ Project Management Institute. [www.pmi.org](http://www.pmi.org)
It is not easy to immerse in a competitive industry and get information of products, processes and practices because of confidentiality issues and that empirical research is not yet a high priority to industry \[12\]. The cost and overhead of the data collection activities to investigate any phenomena can make the process not viable. In spite of this, the company in focus here acknowledges the real value of this kind of investigation, due to its innovative profile and the past experience of some key members with CMMI certification program, including Six Sigma approach \[37\]. Therefore, the research performed in this study was well accepted in the company.

V. Case Study Execution and Analysis

Following our iterative approach, we have conducted four interviews to explore the research questions defined. We considered them a representative sampling in the context described. These interviews were conducted with two people on the front lines of a project that introduced agile programming into an organization that is very experienced in traditional software process practices. Each person was interviewed twice, once at the beginning and once at the end of the project.

These four interviews were complemented by meeting observations and documental analysis. We attended and recorded 40 meetings involving the interviewed participants and all the team members.

The interview transcription and coding was time consuming. For each hour of recorded interviews, we spent an average of five hours of transcription and one hour of pattern coding to tag the interviews’ key words, phrases, and paragraphs. The data collected was then analyzed via cycles of pattern coding which grouped summarizing segments of data into sets, themes or constructs.

After reducing the data to a limited number of classes of beliefs, we characterized them in terms of 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 in 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).

The analysis and interpretation of the results of the case study were based on the ethnographic approach \[20\][31], which seeks to capture the point of view of the participants and their own sense of reality. We represented our main findings in a mind map, which is a diagram used to represent concepts, ideas, beliefs, or other items linked to and arranged around a central key phrase or idea \[9\][3]. Mind maps are useful to visualize, structure, and classify the beliefs and to organize related information.

We used the resulting map to guide the discussion and validation of the findings with other researchers and the participants of the study. At the validation session we performed a focus group with professionals involved in the company’s agile projects on a total of seven professionals from three different profiles: Software Architect, Requirements Engineer, Technical Leader and Process Quality Assurance. Since people with different responsibilities in the company would have different perceptions; the responses were analyzed both individually and on an aggregated basis within and across the group. This gave us a big picture of the main challenges, their relative importance, and their causal relationships.

VI. Results

The team under study has conducted the project according to the company’s development approach, which is based on SCRUM and PMI methodologies. Two stakeholders are very important in this approach. The product owner, who represents the stakeholders and the business, outlining work in the SCRUM backlog, and prioritizing it based on business value, and the SCRUM master, who ensures that the process is used as intended, protects the team and keeps it focused on the tasks at hand.

The project had 14 sprints. Planning meetings were made at the beginning of each sprint, after the team had reviewed what was produced in the previous one. The sprints had 15 days duration for the team and 30 days duration for the actual customer delivery. This was a strategic decision from the product owner (PO) and SCRUM master to make sure that no deadlines was overrun.

The daily meetings lasted around 10 minutes. They were organized throughout the project, although these were less rigid in the last sprints. The initial meetings involved three team members plus the SCRUM master and PO. The project owner, which was also the requirement analyst, participated in these daily meeting up to the middle of the project. After that, the SCRUM master decided that it was important to the team to have more autonomy in the decision-making process. About the same time, the team had a turnover of the two members. After each sprint, the team held a retrospective meeting to identify and discuss problems and opportunities that arose during the sprint.

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 SCRUM practitioners call team self-organization. All information about allocated tasks was recorded in an open source project management application\(^4\) used by the team.

All the information collected was analyzed. After creating the mind map of results shown in Fig. 1 and Fig. 2, we performed a follow-up focus group session to validate it.

A. Common Beliefs

The Fig. 1 shows a mind map of 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.

\(^4\) Redmine project management web application. \(\text{http://www.redmine.org/}\)
Figure 1. Common beliefs between SCRUM master and PO
Both the PO and the SCRUM master believe that SCRUM and PMI are good practices. They have this belief from past experiences in previous projects in another company. They agree that this methodology leads to higher levels of team productivity, provides a better mechanism to control of scope, changes and risks, and helps the team to focus on the business value of each delivery. Both of them recognize that the application of this methodology can foster a real customer involvement, what has a positive impact on project practices. The PO said that:

“With a strong communication and risk management, it is possible to improve the outcomes of requirements definition activity as well as the requirements traceability.”

SCRUM approach is not clear on how to establish monitoring in development teams, so combining SCRUM with the practices of PMI approach improved the project monitoring. On one of SCRUM ceremonies based on metrics, the PO said:

“After using quantitative metrics to monitor sprint’s 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 defects causal analysis and can bring more predictability to the work process.”

The PO and SCRUM master agree that good automated software testing approach is essential to the success of any project. The origin of this belief of the PO is the experience in the current project. For her, the absence of this approach had some negative impacts on project practices such as:

- No time left in sprint for bug correction;
- Poor coverage and low effectiveness of the test;
- Absence of real test cases;
- Focus on sprint deadline, not on product quality.

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

- Poor coverage and low effectiveness of inspection practice;
- Low level of testing expertise of the team;
- Greater effort to perform regression testing.

All these impacts cited were discussed in the follow-up focus they agreed with the result from the interview and after a discussion in the focus group the participants decided to start a new initiative to introduce new testing technologies and practices.
About component reuse, the PO and SCRUM master consider it a good and very important practice. They have successful past experiences in previous projects with the exchange of knowledge and experiences and agree that reuse practice can lead to better results in terms of team productivity and product quality. The PO also highlighted that reuse practice can provide a good infrastructure for company’s next projects.

Beliefs related to process management and job rotation were identified in the interview but were discarded in the focus group, as the participants did not judge them relevant for discussion.

B. Conflicting Beliefs

There are also conflicting beliefs between the SCRUM Master and the PO (Fig. 2). The SCRUM master participated in a SCRUM certification course. The first sprint after the course he started with a meeting to discuss some SCRUM methodology issues especially about backlog changes during sprints, time schedule for bug correction, and the closeness between the team and the customer. The project team has considered some of these issues such as SCRUM gaps that can generate conflicting beliefs between team members. In the focus group meeting they had the opportunity to discuss these points.

The SCRUM master believes that requirements changes during sprints can impact negatively the productivity of the team; on the other hand, the PO thinks that there should be some flexibility on this, since the customer must have their requirements and needs met. In the PO’s point of view, it is very important to reserve extra time in sprints to fix bugs found by the customer and related to previous deliveries. For her the team focus has to be on deadlines, but also on client satisfaction.

On 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.”

For the PO, more closeness between the team and the customer can impact positively the team commitment with the quality of the product delivered and sense of business value.

With respect to story and task estimation, the PO advocates on using the planning poker technique, while the SCRUM master thinks story points technique is better. Currently, the stories are estimated by hours and the PO is the responsible for this activity. The SCRUM master disagree with the PO, he believes that the team should be more autonomous to estimate task and stories. As consequence, the SCRUM Master believes that it is important to the team to know the story requirements very well and deeply, but this is not happening in the project because the PO only elicit the requirements in a high level of abstraction. The main negative impacts of the conflict, from the perspective of the SCRUM master, are:

- No proper time for software coding and testing;
- Sprint planning without slack;
- Lower level of estimation expertise of the team.

VII. DISCUSSION

We have described an industrial case study using a qualitative approach to investigate the origins and impacts of beliefs on software development team practices.

Considering research question 1, the study showed that beliefs were originated from two main sources: past and current project experience. Past experiences originated from things that did or did not worked on previous projects, some of them executed in other companies. Current experience originated when the SCRUM master adopted practices demanded by the PO. In these cases, the beliefs were indirectly originated by the demands of senior management.

It is important to point out that ultimately all beliefs originated from personal hands on experience. It is also relevant that the past experiences were taken into account without much consideration for their context.

This brings us to research question 2, how beliefs impact current software project practices. Some beliefs indeed triggered the adoption of new technologies, but the decision of which technology to adopt was mostly based on expert opinion and the community’s folklore. For example, the opening of the testing technology initiative mentioned in section 6-A was based on personal experiences, but their choice of which technology to use, the Selenium tool for functional and regression testing, was based on expert advice and the current software engineering community buzz around it.

With respect to question 3, we found out that there were conflicting beliefs between the PO and the SCRUM master. This definitely hinders the adoption of practices. Some conflicting beliefs are subtle. For example both interviewees, agreed that current effort estimation practice is limited, but digging further we found out that they disagree on which practice should substitute it. They also disagree on the role that should be responsible for the estimation. Our approach of formally capturing and representing the beliefs on a mind map definitely helped to identify and point out those conflicts to the involved stakeholders.

An even more subtle problem was when the interviewees had a common belief, but did not act upon it. We named those semi-beliefs, things that people believe, but survive without the practices recommended by the belief. Like the previous case, the act of formally capturing and representing the beliefs on a mind map helped to discuss those semi-beliefs with the interviewees. This was the case of the applicability of software testing automation for the organization. When confronted by the fact that they believe in testing automation by were not using it, the PO and SCRUM master decide to act upon it, opening an official testing automation initiative in the organization.

Lastly, there were those cases of common strong beliefs. This is the case of component reuse for example. Both, the PO and SCRUM master actively acted on the direction of implementing good reuse practices from the very beginning of the project. The participation of the researchers in the SCRUM meetings and in the focus group meeting helped us to realize that this type of common strong beliefs are stren-

themed and transferred to other people involved in the project.

VIII. VALIDITY ISSUES

We have some conclusion validity threats; we are working with a small sample (one project), limited number of interviewees (two per project), and interviews (two per interviewee). We addressed this issue by triangulating data from interviews, participant observation, document analysis and focus groups. 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 answer the interview questions. And 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 consultant for quality assurance and process control in the company investigated.

To be 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 observations with other researchers and with the participants.

External validity may be an issue, 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 case study’s environment is context specific, since the participants belong to the same company.

IX. CONCLUSION AND FURTHER WORK

This paper addresses the issue of belief origins, usage and impact in software organizations. The case study showed that it is possible to capture and represent beliefs in a mind map. It also shows that this type of formalization helps to discuss those beliefs and the practices that are adopted based on them. The study showed that people act mostly based on their strong beliefs and that those are obtained from personal experience. Even when working in the same project, people have conflicting beliefs. They also may have semi-beliefs, things that they believe, but do not act upon it. The mind map proved to be a useful tool to expose and discuss those conflicting issues with the practitioners involved in the case study.

This research adopted a qualitative approach, which helps in capturing realistic scenarios. Our intention is to increase the relevance of the obtained results to the software industry, and bring new knowledge to the academia for better understanding the phenomena.

Our next step is to include data from two other projects and do a cross-case analysis between them [26]. Next, we will use the grounded theory approach [8][7] to form initial insights inductively on the basis of multiple case study data. Through the synthesis of all this evidence, we intend to provide interpretative explanations about them using a meta-ethnography approach [27][6][2].

ACKNOWLEDGMENT

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

The authors 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.

APPENDIX A. INTERVIEW QUESTIONNAIRE

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.

REFERENCES