A Case Study of Software Process Improvement Implementation

Daniela C. C. Peixoto¹  Vitor A. Bastista²  Rodolfo F. Resende¹  Clarindo Isaías P. S. Pádua²

¹ Department of Computer Science – Federal University of Minas Gerais, Brazil
² Synergia, Department of Computer Science – Federal University of Minas Gerais, Brazil
Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG – Brasil
{cascini, vitor, rodolfo, clarindo}@dcc.ufmg.br

Abstract—Managing a successful Software Process Improvement (SPI) is a challenging issue that many software companies face today. Many companies have invested huge amount of money in improving their software processes. This can be confirmed through several papers that present the results of SPI programs. However, as pointed out by the literature, many of these programs have encountered difficulties to achieve the desired benefits. This is usually not caused by incorrect new processes, but due to inadequate SPI implementation and, particularly inadequate SPI adoption. This paper evaluates an SPI program, discussing its implementation problems with emphasis on adoption issues. The analysis was carried out as a case study in a software development organization. Our findings suggest that other than finance, technology and other issues, several implementation aspects, in particular effective adoption strategy, are needed to achieve a successful SPI program. Our main contribution is to give evidences that an SPI implementation process can be assessed and improved using objective measurements and available methods and practices. In particular, we measure the adoption of an SPI practice, analyzing the evolution of the improved practices rework during the development of two software projects.

Keywords- Software Process Improvement - SPI; SPI Implementation; SPI Adoption; Rework

I. INTRODUCTION

Many organizations experience a successful start on their Software Process Improvement (SPI) initiative and after some steps in their improvement process they realize that the overall engagement to change weakens significantly after the initial excitement [1]. Previous studies show critical factors that can be sources for this problem e.g. resistance to change, previous negative experience, lack of the evidence of benefits, imposition, resource constraints and commercial pressures to meet customer demands [2], [3].

Many advances have been made in the development of the SPI standards and models. However, we cannot observe equal advances in the SPI implementation or adoption processes. This has resulted in limited success of many SPI programs [4]. Herbsleb and Goldenson [5] showed that 67% of SPI managers prefer guidance on how to implement SPI activities, rather than additional SPI (what) activities to implement.

Despite the importance of SPI implementation, little empirical research has been carried out analyzing in detail the implementation steps of SPI new processes, specifically in evaluating the SPI adoption.

The purpose of this article is to describe, via a case study, the context of the SPI implementation, with emphasis on adoption issues. We show that the challenges of a successful SPI program are not limited to the identification and creation of new processes. In addition, we verified that there are many barriers to SPI which can be well characterized after the assessment of the SPI implementation. We also present an objective measurement to evaluate the SPI adoption. This measurement can provide for organizations a better understanding of where they are, when considering the SPI implementation process. Considering the successful results of this initiative, we believe that it should be easily adopted by many software development organizations.

The focus of this paper is based on the following research question:

How SPI implementation can be assessed and can incorporate empirical measurements?

To answer this question we conducted our case study in a software development organization, evaluating an SPI initiative that started in 2007. During two years, we observed that to be successful, an SPI implementation needs to recognize human, social and organizational aspects with the same or higher consideration given to finance, technology and other issues.

The remainder of this paper is structured as follows. Section 2 describes the background of this work. Section 3 describes the case study and some findings. Section 4 presents the final remarks and future works.

II. BACKGROUND

A. SPI Adoption and Implementation Process

A recurrent problem reported in the literature for SPI is a lack of an effective strategy to successfully implement SPI programs. We observed that much attention has been paid to what activities to implement instead of how to implement these activities. In this context, Niazi et al. [4], [6] developed a framework that resulted from SPI literature research and empirical studies. In the design of the framework, Critical Success Factors (CSF) and Critical Barriers (CB) for SPI were identified and extended. In addition, interviews were conducted with Australian practitioners in order to select and analyze the factors that
play positive or negative roles in SPI organization. The framework is composed of three components:

• Maturity stage dimension: The IMM (Implementation Maturity Model) is adapted from CMMI [7] and presents four maturity levels. Level 1, “Initial”, indicates that the organization is not aware of SPI implementation process. Level 2, defined as “Aware”, indicates that managers and practitioners are becoming aware of SPI implementation programs and their benefits. Level 3, “Defined”, is the level where processes were documented, standardized and integrated into the organizational process. Level 4, “Optimizing”, is the level where the organization establishes structures for continuous improvement.

• CSF and CB dimension: Instead of the Process Areas (PA) as adopted by CMMI, IMM adopted CSF and CB which are organized into three categories “awareness”, “organizational” and “support”. Each category provides a list of related CSF and CB. The categories are divided among the levels of the IMM.

• Assessment dimension: This dimension provides a list of practices that allows assessing how well the factor has been implemented in practice.

Niazi and others [6] conducted three case studies in order to test and evaluate the framework. The results of their experimental evaluation have shown that the framework has potential to assist practitioners in the design of effective SPI implementation initiatives. However, their work focuses only on the process dimension of SPI implementation. We believe that for a complete evaluation of a successful SPI implementation it is also important to recognize human, social and organizational dimensions. This means evaluating the adoption of the improvement by the practitioners: developers and managers.

It is not easy to analyze if the improved processes have been fully adopted. It is almost impossible to measure if processes have really changed the hearts and the minds of the practitioners [8]. Process adoption is generally characterized as two orthogonal dimensions: infusion and diffusion [8]. Infusion describes how deeply the new process has been adopted by the target population. Diffusion describes how broadly the target population has adopted the new process [9].

Adoption of a change typically follows a number of stages. In our work, in order to evaluate the human dimension of the SPI implementation process, we applied the seven-phase adoption curve discussed by Conner and Patterson [10]. The seven-phase consists of: contact, awareness, understanding, trial use, adoption, institutionalization and internalization.

B. Organizational Context

Here we describe a work that was performed as a collaborative study with Synergia. Synergia is a laboratory for software and systems engineering, hosted in the Computer Science Department at Federal University of Minas Gerais, Brazil. Synergia is internally organized as a commercial software development organization, but it also retains important academic characteristics [11]. Synergia maintains 85 people in its staff composed by undergraduate and graduate students and non-student graduated professionals, most with a Computer Science background.

Synergia uses a tailored version of the Praxis model-driven software development process [12] in its software and systems engineering projects, called Praxis-Synergia. Although the Praxis process has been designed and applied primarily for education and training in Software Engineering, it provides tailoring guidelines, which must be interpreted according to the current situation of the development organization. The Praxis material is available in a book and kept updated in the author’s Web site.

One important characteristic of Praxis, which is maintained in the tailored version, is that it models the development of each product use case as a sequence of development states or control marks. The name of each state evokes how far the use case has advanced towards complete implementation and acceptance. The states are: Identified, Detailed, Analyzed, Designed, Specified, Realized, Implemented, Verified, Validated and Complete. Pádua [11] discusses these states and how each state is associated with one or more quality control procedures such as inspection, test and management review.

III. STUDY DESIGN AND EXECUTION

We performed a case study in conjunction with a SPI initiative conducted at Synergia. This case study analyzes the rework effort and the number of defects detected in a specific test execution task in different projects at Synergia. The aim is to provide more evidences about the results achieved with the process improvement implementation assessment and the empirical SPI adoption measurement.

This is an embedded case study [13] in one company with two units of analysis: two major projects. For every use case of each project we collected the following information: the percentage of rework related to the test execution (% of rework = rework / (work + rework)) and the number of defects.

A. Process Improvement

In this section we summarize the SPI initiative that was conducted at Synergia. A detailed description of this program can be found in Peixoto et al. [14].

During 2007, Synergia managed its improvement effort by analyzing problems in a specific project. The work was organized as a Defect Causal Analysis (DCA) process [15]. The DCA process selects recurring problems with the aim of identifying their principal causes and propose some solutions to avoid (or reduce) new occurrences. At that time, the largest existent project (~ 5,000 Function Points),

1 www.dcc.ufmg.br/~wilson/praxis/ (in Portuguese)
referred here as Project TBI (To-Be-Improved), was chosen to assure that the recommendations would be executed.

The problem discussed ahead was detected while performing the quality procedures of the Implemented state. Before delivering the executable code to the Test team, the developers themselves were required to execute a list of test procedures (a subset of the complete manual test specification), referred here as verification practice. The purpose of this verification practice was to identify errors made by the developers themselves (for example, a nonoperational save command) with the aim of treating them before the execution of the complete manual test specification by the Test team and avoiding rework.

Although in Project TBI all developers were required to execute the list of test procedures, they did not execute it correctly or they did not execute it at all. In a broader evaluation, the non-execution of these test procedures could be the result of time pressure, project cost overrun, and lack of training or experience of the team. But at Project TBI, the developer had enough time to do their task and they knew or were trained in what they had to verify. Trying to find one possible cause, the developers in a DCA meeting were asked about the reasons. It was not possible to identify one specific source of the problem. However, observations during the meetings drew attention to a specific team behavior: they simply neglected this task. But, why this happened? Even if the project manager had required the developers to execute this verification practice they did not execute it! One rationale behind this problem could be resistance to change, or in other words, the resistance to include this practice in the developers list of ‘daily’ practices. As Humphrey observed [16] this is not a trivial problem “particularly because even intelligent people often will not do things that common logic, experience, and even hard evidence suggests that they should”. Trying to understand this resistance to change behavior, we found some studies [17], [18] that show how cultural aspects can affect software process improvement programs.

Considering this hypothesis, one adopted solution was to force the developers to execute the test procedures using a specific tool called IBM Rational Manual Tester, referred here as RMT, which records each step executed and the test result (whether it succeeded or failed). In this way, the developers would provide to the Test team a proof, metaphorically like signing a document, that they had really executed the test procedures.

After five months of implementation, the benefit of this action amounted to 217% of the investment in DCA meetings. At the end of Project TBI, in 2008, there was an average reduction of 5.66 hours per use case of rework to fix and check for defects detected by the Test team, resulting in total savings of 796 hours. The number of use cases without any defects increased from 21% to 53%, with this specific verification practice conducted by the Test team. The average number of defects detected by the Test team reduced from 5.1 to 1.7 defects per use case and the number of critical defects was reduced by 70%.

The other project, referred here as Project IMPROVED, started in 2009 and ended in January, 2010. It had approximately 1,500 Function Points and it presented the same characteristics of project TBI. Almost all developers that worked in Project IMPROVED came from Project TBI.

Comparing the rework originated from the verification practice in the two projects we observed that:

- With 95% of confidence, despite all the rework and number of defect reductions, we cannot say that there is a difference between the number of defects per use case before and after the process improvement in Project TBI (Fig. 1).
- With 95% of confidence, we can say that there is a difference between the number of defects per use case comparing Project TBI and Project IMPROVED. (Fig. 2)

In the next sections, we evaluate the SPI implementation process trying to understand the modifications in the number of defects and rework effort in both projects.

B. SPI Assessment

We assessed the SPI initiative in two phases. First we evaluated the SPI implementation process using the IMM model [4] [6]. In this case, we evaluated only Project TBI, which was the first project to adopt the improved practice (the verification practice). Second, we evaluated the improvement adoption by the Development team in both projects. Then we documented the lessons learned in both phases.
1) SPI Implementation Process Assessment: In spite of Project TBI improvements, some use cases still presented unreasonably high number of defects, though much less critical ones. We conducted a semi-structured interview with each developer that had defects reported for their use case, trying to find the reasons. We noticed that:

• Some developers generated a log without carrying out all the test procedures. RMT has a command that allows users to select all the test procedures and mark them as successful. The developers said that some test procedures were so extensive and detailed that the purpose to be a simple verification was missed.

• Some developers reported that some test procedures specified by the Test team were incorrect. So when the Test team executed them, they reported improper errors.

• Some developers reported that the requirements changed and the test procedures were not properly updated.

To highlight the improvement process problems, the Software Engineering Process Group (SEPG) conducted a post-mortem process assessment of the SPI implementation process. Specifically, they evaluated this verification practice and its implementation problems using the IMM framework. The process maturity assessment method involved assessing the SPI implementation process in the organization and using the results to determine a process maturity level.

The SEPG filled the assessment form together with some of the practitioners that participated of the DCA. Table I summarizes the results of the assessment. The complete IMM practice list can be found in Niazi et al. [4].

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Weak implementation factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2 – Aware</td>
<td>Awareness of SPI, Staff involvement and Senior management commitment</td>
</tr>
<tr>
<td>Level 3 – Defined</td>
<td>Creating process action teams, Experienced Staff, Staff time and resources, Time Pressure and Organizational Politics</td>
</tr>
<tr>
<td>Level 4 - Optimizing</td>
<td>Reviews</td>
</tr>
</tbody>
</table>

Our suspicions that the organization was operating at low maturity levels were confirmed. Considering the results, the organization stands at Level 1, “Initial”, of IMM because three factors of Level 2 were not fully implemented (score less than 7). The Critical Success Factors have an average score of 6. Some considerations of the results are:

• Level 2: Our attention was drawn to the first problem detected during the developers’ interviews which is reflected in the factor “awareness of SPI”. We noticed that the benefits of the SPI program were promoted only before the SPI implementation. In addition, the awareness was not stimulated after the implementation in Project TBI. Related to the staff involvement and senior management commitment, the low score was obtained mainly because some managers were not involved in the SPI process.

• Level 3: The main concern in this level is to have documented, standardized and integrated SPI implementation processes into the organizational process. As expected in a Level 1 organization, most of the initiatives were ad-hoc and non-standardized.

• Level 4: As the process is not documented and integrated, no revisions were planned to be conducted during the SPI initiative. In this way, it was not possible to establish continuous improvement.

We were surprised that the organization successfully used good practices, as formal methodologies (for example, testing an improvement firstly in a pilot project and providing training) in spite of the fact that it did not really have a strong base of fundamental practices. This may suggest that the notion of what is an advanced practice is wrong or that the organization can successfully use such practices without having the other practices implemented beforehand. In addition, we were aware that without quantitative information it would not be possible to promptly identify SPI implementation problems and provide the corrective solutions, as it happens to be very difficult to truly change the behavior of engineers and managers.

Some important lessons learned during this assessment were:

• The IMM provides a good methodology to evaluate the weak and strong implementation factors. Also it provides a way for an organization to improve its practices of SPI implementation process.

• It is crucial the commitment of the people involved in SPI. The SPI literature [19] recognizes that without commitment from all organizational levels the improvement goals will be difficult to achieve. So, people involved with the SPI initiative must perceive the benefits deriving from its deployment, and not only its costs. Also it is important to overcome the resistance of the team developers to adopt the new practices. This involves supporting them to give up established ways of working.

• It is important to take measurements throughout the whole process, including adoption measurement in order to determine a successful adoption of the new practices.

2) Adoption measurement: With the aim of measuring how far the target population has come along the stages of the adoption curve we used an adoption measurement of the SPI implementation. Instead of calculating the number of people who are at a certain stage of the adoption curve and give weighs to each stage, as proposed by Heijstek and van Vliet [9], we evaluated the verification practice rework evolution executed by the Test team. Since we did not have data about the team knowledge in previous projects, we carried out this post-mortem analysis. Based on our historical data and using expert judgment, we determined a percentage of accepted rework for each stage, considering this specific verification (Table II). In this way, we believe
the institutionalization happens when the use cases do not present (significant) rework. A clear understanding of the nature of rework could lead to stronger support for the evaluation of the implementation process.

The reasons for our measurement are:

- The test procedures verified by the developers are detailed and objective. So, if developers carried out this verification, no defects should be detected by the Test team, when carrying out the same test procedures.
- The tool that supports the verification generates a log of the test procedures execution. Before conducting the verification, the tester should verify the tool log. If there are problems, the developer would be notified to solve them before delivering the code.

**TABLE II. MEASURES FOR EACH STAGE OF THE ADOPTION CURVE.**

<table>
<thead>
<tr>
<th>Stage</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>No changes are observed in the rework. Our historical data shows that we have usually 45% of rework during the test procedures verification.</td>
</tr>
<tr>
<td>Awareness</td>
<td>The rework reduced but not expressively (~5%).</td>
</tr>
<tr>
<td>Understanding</td>
<td>The rework reduced, but not expressively (6-10%).</td>
</tr>
<tr>
<td>Trial Use</td>
<td>Because more developers adopted the tool, the rework reduced in 11-20%.</td>
</tr>
<tr>
<td>Adoption</td>
<td>Because all developers adopted the tool, the rework reduced in 21-25%.</td>
</tr>
<tr>
<td>Institutionalization</td>
<td>Reduced rework, between 0-20%.</td>
</tr>
<tr>
<td>Internalization</td>
<td>-</td>
</tr>
</tbody>
</table>

The evolution of the rework percentage computed during the verification practice carried out by the Test team is represented in Fig. 3. The medium value of the rework for project TBI was calculated for each month starting in 2007 and ending in the middle of 2008, and, for project IMPROVED, starting in 2009 and ending in January, 2010. The area labeled “A” shows data before the improvement. This period presented the highest rework percentage. The area labeled “C” shows the lowest rework percentage and corresponds to Project IMPROVED. Fig. 4 presents similar data related to areas “B” and “C” of Fig. 3.

We observed that the percentage of rework presented a value equal or less than 20%, only in areas “C” and “D” of Fig. 4. Besides the usage of RMT tool in Project TBI by all the developers, we cannot observe an expressive reduction of rework. This is due to the SPI implementation problems, specifically SPI adoption problems. As discussed in the previous section, after analyzing the data and after the interviews we observed that two important factors were: the awareness of the SPI (process factor) and also the resistance of the team to change their behavior (human factor).

The human factor plays an important role during the SPI implementation program. We observed that SPI implementation had several flaws, but the resistance to change affected directly the rework percentages.

Some important lessons learned during the adoption measurement are:

- Determining an effective way to evaluate the adoption of a certain technology or methodology is not an easy issue. We need to consider factors that are difficult to measure, as the level of knowledge of each team member in that subject. We observed that in both projects the main source of rework problems were the human behavior.
- It is possible to develop alternative ways of evaluating the adoption, which can show an interesting evolution pattern during some period of time.
- The adoption of a simple change in the process can take a long time to be really effective, as we presented in the previous discussion. Only in Project IMPROVED, after an improvement in the way that the test procedures were specified, was that the practitioners achieved the “Institutionalization” stage.

**C. Limitations**

There are some limitations in our study. First, our data come exclusively from Synergia. In addition, this study needs to be replicated in other organizations to provide a better confidence in the general application of the results.
However, the SPI implementation and adoption problems seem to be similar to that of comparable companies.

Second, we showed that it is possible to develop alternative ways of evaluating the adoption. However, this can have external influence of other factors, for example, team turnover, and requirements and processes change. We tried to reduce this threat to validity triangulating the data with other sources such as interviews and management evaluations.

Finally, it is possible that the team did not express their real opinions during the interviews and meetings. This threat was limited by participants having guaranteed anonymity and being shown the benefits that one single improvement can bring to the organization.

IV. Conclusion

Our work revealed some fundamental difficulties of assessing process maturity and adoption of software process improvement practices.

The IMM model allows the organization to identify factors that have positive and negative impact on implementing SPI, and it also serves as a guide to improve the whole SPI implementation process. Besides the process assessment, another important dimension that needs to be evaluated during an SPI implementation program is the human dimension. In this paper, we applied an adoption measurement approach to evaluate how deeply the new process has been adopted by the target population. We observed that with this measurement it is possible to evaluate when the process improvement really achieve the adequate values expected by the organization.

The key lesson we learned is that while it was reasonably easy to identify areas where improvement is needed, it is much more difficult to understand the main cause of the implementation problems, and how to identify and propose reasonable solutions. Other important observation is that since change is not easily accepted by everyone in a team, the focus in this first phase should be to get people excited about the benefits of the change instead of managing resistance.

In conclusion, we should define assessment methodology with a broader vision of the problems that could affect the SPI implementation. Experiences with other organizations could help us in identifying alternative ways to evaluate the adoption of an SPI implementation process. As a future work, we will evaluate other process improvements implemented in the organization with the aim to compare implementation process problems and the adoption measurement alternatives.

ACKNOWLEDGMENT

We would like to thank Capes Foundation grant BEX 1893/09-2 and Fundação de Amparo a Pesquisa de Minas Gerais (FAPEMIG) grant CEX-APQ-01201-09.

REFERENCES


