A Review on Software Process Improvement Methodologies for Small and Medium Enterprises

M. Suganya
Research Scholar
School of Information Technology, Department of Computer Center, Madurai Kamaraj University, Madurai

Dr K. Alagarsamy
Associate Professor
School of Information Technology, Department of Computer Center, Madurai Kamaraj University, Madurai

Abstract

Most of the companies around the globe which are involved in software development are small and medium scale enterprises. In the world economy these organizations are considered as the back-bone. As per history, it is found that more than 85% of the companies in US, China, India, Finland, Ireland etc are small and medium scale enterprises during the year 2008. These small and medium enterprises at some time have realized that their process and working methods should be improved and is crucial for their business. They felt difficult to implement because of the lack of knowledge as well as resources. Successful software process improvement implementation is a herculean task for these small and medium enterprises since they are not capable of investing the cost of these programs. They are unaware of the effective innovations as well as the factors that can influence the adoption of software process improvement in small and medium enterprises. Most of the small software organizations are not adopting existing standards. The reason is that they perceive them as being oriented towards large organizations. The studies have shown that small firms’ negative perceptions of process model standards are primarily driven by negative views of cost, documentation and bureaucracy. In this paper, we present a brief description about the software process improvement as well as a review on the current software process improvement methodologies for small and medium enterprises.

Keywords: Software Quality, Small and Medium Enterprises, Software Process Improvement, Software Process Improvement Methodology

I. INTRODUCTION

A. Process:

Process can be defined as “The organization of people: automated support, procedures and standards into work activities designed to produce a specific end result” [1]. The process integrates the people, tools, and rules, as shown in Figure 1, a process require people who are skilled, trained, and motivated and are able to use tools and equipment. Process should be based on defined rules and methods in order to transfer the raw material (input) to the product faced customers (output). Three main elements of process:

- People: The people who have the skills, training, and motivation.
- Rules and method: The rule and method to implement task.
- Tools and Technology: Techniques and tools must be needed.

B. Software Process:

According to Watts Humphrey (1993) definition “Software process is the Set of Activities, Methods, and Practices that guide People in the Production of Software [2].” Software process can be also be defined as “The set of activities, methods and transformation that people use to develop and maintain software and the associated products, for example: product plans, designs documents, code, test cases and user manuals” [1].
A well-defined software process can help organizations to improve their capability and maturity level throughout the business system. Better quality of process will help organization in standardizing software development, improve reusability and consistency of project teams, perform better practices in inter-industry, improve the quality of products, reduce extra cost and save time. It is important to understand the actual position of organization and what are the business goals in order to start process improvement. That is why; companies should identify their process strengths and weaknesses, improvement areas, potential risks and opportunities [3].

Nowadays, the research for the software process can be broadly divided into the following three aspects:

- Software process modeling language
- Software process support
- Software process management and improvement.

C. Software Process- Objective:

If the process is not well defined, organized and well-disciplined for example uncontrolled requirements, unmanaged activities, poor project planning, uncontrolled quality, no project tracking and no risks and mitigation plans, uncontrolled release will result in a low quality software product or sometimes even a fail product. We can say that quality of the software product is managed by the quality of the software process and if the process is well defined, managed, controlled, measurable and effective the end result will be according to the expectations. Organizations find ways to improve their process capability in order to jump from a pre-maturation level to next level of maturity. Much of the effort is put on development tasks, testing and documentation, nobody wants to take responsibility in improving overall development process hence without focusing on quality of the process there will be no improvement in developed software products. It is the responsibility of the top level management to think about process improvement programs inside an organization.

D. Software Process Improvement:

Karl E. Wiegers explains process improvement as simply: “consistently apply the practices that give you good results, and change the practices that cause problems” [4]. The motivation behind SPI is to achieve business goals and develop quality products through a mature process and practices. The objectives of software process improvement is to set methods in order to improve the development process including project management, eliciting and managing requirements, decision making, measuring performance, planning the work, handling the risks and many more [5]. It’s important that organization should analyze the organizational structure and process and identify the main reasons behind their failed and successful projects and take measures to make improvements wherever needed.

Software organization are using SPI product suites, frameworks, models and self-made techniques in order to improve process capability, and trying to advance from a pre-maturation level of process to an improved and mature process. Many Software Process Improvement Framework designed by experts provides broad and comprehensive interpretation for the software process improvement. The Software Process Improvement Framework, such as Figure 2 explains the process improvement route map and different components involved in continuous improvement.
V.M. Paula et al [6] states that the existing process models do not support informational and behavioral aspects of the software development process. To ensure the quality of the product, reduce costs and maximize productivity, every software firm needs a well-understood and managed software development process.

According to Johnson and Broadman [7], tailoring is needed in specific areas, such as documentation, management, review, resources and training. According to R.S. Pressman [8], the software process is a framework of tasks to build high quality software. Somerville [9] stated that the software process is a structure of activities to develop software systems and pointed out that software process consists of the four activities: (1) Software Specification; (2) Software Design and Implementation; (3) Software Validation; (4) Software Evolution.

E. Importance of SPI:
Siakis characterizes the “software crises by an inability to develop software on time, on budget, and within requirements”. Software development has become a complex job in recent decade due to increasing number of projects, competitive environment and the complex development tasks. The competitive market has challenged software companies to complete the projects and develop quality products in desired time and allocated budget. Managing and handling the quality of projects and products is not easy especially when technology changes rapidly. In order to address these issues, it is essential for companies to focus and improve their organizational practices, and processes. The more capable and mature an organizational process the better and quickly they will achieve business results.

![Software Process Improvement Framework](image)

Fig. 2: Software Process Improvement Framework

It is important to understand that Process improvement means a change in the entire process to get a better output rather than original output. There are lots of factors that will affect the project and product quality, and their role is very important in software process, such as:
- Project should be within time;
- Under budget;
- Better quality;
- For long term business;
- Reuse.

The process improvement methods can help organizations manage their resources effectively, organize people and procedures, update technologies, reduce cost and time and do job in much faster, efficient, quick, safer and easy ways. Below is the list of some companies and organizations that started to work on SPI and the results they got from the process improvement.

1) Motorola Personal Software Process (PSP) Results: Motorola leads the way with the successful industrial implementation of the Personal Software Process (PSP). A Motorola Division in Boynton Beach, Florida used the PSP to achieve zero defects in use over 18 software projects, removing over 76% of their defects before testing began [10].

2) IBM Defect Prevention Results: Following the publication of Philip Crosby’s book, “Quality is Free,” an IBM division in Research Triangle Park, North Carolina successfully pioneered, mastered, and used the Software Defect Prevention Process, achieving 50% quality improvement the first time used, and up to 99% quality improvement in other instances, without product appraisal activities, that is Inspection and Test [11].
Motorola Capability Maturity Model (CMM) Results:
Motorola is a world leader in applying Statistical Process Control (SPC) to hardware design and manufacturing processes. Only recently has Motorola’s software development management begun to match their world renowned quality management performance in hardware, electronics, and communications [12]. From above examples of SPI, we made this conclusion that software industry is focusing more on software development process, and by improving and taking right actions at right time on right process can produce high quality process which can deliver high quality products.

F. Software Process Improvement Models:

1) Six Sigma:
Six Sigma is a process improvement management framework to achieve bottom-line results, and customer’s loyalty. In short the objective of Six Sigma is the implementation of a measurement based strategy that is focused on process improvement and variation reduction (Bendell 2000) [13]. Six Sigma proposes software quality improvement approach called DMAIC which stands for define, measure, analyze, improve and control. Six Sigma helps in improving capability of an organization by providing various tools and methods which help companies to achieve their goals by setting a business strategy.

2) TickIT Guide ISO 9001:
TicketIT project started in 1991 with the purpose of stimulating software system developers to think about process quality, how to achieve the desired quality and how to manage this quality in order to perform continuous improvement. TicketIT also provides a practical framework for the management of software development quality [14]. The basic purpose of this framework was to define and design guidelines for the development teams to follow the ISO 9001 requirements and standards.

3) Spice:
Spice was established by an organization in 1993 and SPICE stands for software Process Improvement and Capability Determination. Main purpose of SPICE framework is to analyze and assess the software process and on these assessments, provide information about process strengths, weaknesses and capability to achieve its goals. The definition of the term “process assessment” is “A disciplined evaluation of an organization’s software processes against the process model or variant model described in this International Standard.” [15]

4) Bootstrap:
BOOTSTRAP was originally designed by using SEI model as starting point and extended on features from ISO 9001 guidelines [16]. BOOTSTRAP is a European method for software process assessment and improvement that was designed and developed to speed up the application of software engineering technology (Zahran, 1998)[17]. The interesting feature of BOOTSTRAP is its suitability for all kinds of software development organization whether they are small or big.

5) CMMI:
CMMI is a framework that provides a process improvement setup for the software engineering and product development. CMMI combined best practices and basic principles and interlinked them in order to improve the process inside an organization. [18].

II. Literature Review

Russ and McGregor [19] observed that software development process can be just as critical to a small project’s success as it is to that of a large one due to the number of external dependencies per team member. They further argued that its goal is to produce high quality and timely results for today’s market without imposing a large overhead on a small project. Paquin [20] identifies assessments, project focus, documentation, required functions and maturity characteristics as process issues for small organization. Abbott [21] identifies six keys to software process improvement in small organizations:

- Senior management support
- Adequate staffing
- Applying project management principles to process improvement
- Integration with ISO 9001
- Assistance from process improvement consultants
- Focus on providing value to projects and to the business

Johnson and Brodman [22] identify seven small organization/small project challenges:

- Handling requirements
- Generating Documentation
- Managing Projects
- Allocating Resources
- Measuring Progress
- Conducting Reviews
- Providing training
Larsen and Kautz [23] also viewed that these organizations are afraid of the initial expenses which they assume are large both with regard to direct costs for process assessment, training and tools, but also due to indirect costs for personal and time resources when implementing improvement actions.

Kuvaja et al. [24] further supports that it is quite difficult for any SME to choose an improvement approach, and to apply it in their organization without the help of external consultants or substantial investment in the time of their software managers. Cultural issues like resistance to change from the employees or the management areas, who regard the extra work required for quality assurance as a useless and complicated burden put on the developing team.

Dyba [25] also found empirically that small organizations implemented SPI as effectively as large organizations, and in turn, achieved high organizational performance. According to his study, the main lesson to be learned is that to implement SPI at least as effectively as their larger counterparts, small software organizations should capitalize on their relative strengths in employee participation and exploration of new knowledge.

Kautz [26] further supported the view that even small organizations with little more than two developers can profit from some basic formal routines. According to his research project conclusion if procedures are defined, concisely described and tested, feedback from these tests can be used as feedback to improve the procedures and routines.

In other words, tacit knowledge of software is socialized. At the same time, the business of software is also reaction on its support elements. “Sky-Earth-People” is the “Three-Cai” of Change of Science, which is origin of Chinese traditional philosophy, and “Shi-li Wu-li Ren-li” is put forward by Prof. Gu jifa and Prof. Zhu zhichang. “Soft-ware Psychology – Software Management – Software Economics” is adopted from software terms.

In J. N. Warfield’s words [27], we could manage the complexity of software product through process design. Processes reduce personal mindbugs (cognitive barriers) primarily through the way in which information is sequenced. If members of groups are given rein to choose the topic of their discussion at random, and if several members speak at the same time, sensible discussion that leads to some organized product is hard to obtain. But if the process is designed so that the subject is broken down into a series of carefully designed questions which are presented under computer control for group discussion and resolution, the dialog becomes focused and the products of the dialog can be aggregated and organized with ease. Such a design will meet with group approval, provided the group is assured that important subjects will not thereby be excluded, and that their contributions will ultimately be incorporated on all matters perceived by group members to be relevant.

Staples and Niazi conducted a motivation-oriented review at the organizational level of CMMI-based SPI [28]. Their conclusion is that organizations are motivated by business benefits: either an increase in software quality or an increase in productivity. They show that only 5% to 10% of CMMI-based SPI initiatives are motivated by human factors (knowledge, leadership, motivation, and environment). They claim that “people-related reasons [are] rarely given by organizations.” This conclusion supports the hypothesis that the impacts of CMMI-based SPI on developers are often overlooked by organizations, and might be a reason for some of the SPI failures.

A less formal review by Sulayman and Mendes [29] shows that developer satisfaction is an important success factor. Their primary study shows that the initial resistance to the SPI initiative was mitigated through the application of an incremental approach, with measured and visible paybacks [30]. This case study shows that a focus on developers can be essential to SPI success.

Our review was also inspired by Beecham et al.’s systematic review of motivation in software engineering [31], which includes Baddoo's extensive work in the domain. Their main finding is that there is no obvious pattern of developer motivation. They note, however, the importance of ‘job fit’, or the degree of coherence between the needs of developers and those of the organization. A mismatch between the two implies a poor job fit, and therefore a lack of motivation. Compatibility with previous methods is another factor highlighted by this review.

Biocchi et al. point out some of the challenges faced by software engineering researchers [32], among them the difficulty of separating the impacts of a technique from those of other factors. Weaknesses in bias evaluation can exacerbate this problem by adding the effect the researcher has to the study data.

In [33] case study investigates SPI model or techniques for software improvement in web software companies. Software process improvement used in AGILE method is proposed in [34], Process Meta model is proposed in [35] after investigating the challenges, advantages and disadvantages of existing models. MECA [36] is introduced which provides complete monitoring for software process and help in providing the ground basis to carry on improvements in existing model. For small organization which mainly focuses on the quality rather processes which actually emboss that quality. But there are many factors involve in small organizations. In [37] describes different small organizations reviewed to improve the literature in small organizations. In [38] identification techniques are proposed for SPI in terms of Quality Functional Deployment (QFD). QFD is basically methodologies that meet the requirements of the customers. In paper [39] takes the voices of customer according to the CMMI reports. In [7] suggests different ways to improve the process and build an environment to achieve. Resistance factors are obstacle in the SPI, in [40] author identifies the resistance factors when implementing the software process improvement. EBM model is proposed in [41] which helps integrates several concepts which contributes a thoroughly understanding of software engineering concepts. Also training plays a very vital role in implementing software process and its improvement. So in [42] shows that training can be very useful to support SPI. Process Quality Measurement Model (PQMM) [43] is introduced which is used to analyze the quality for software organizations.

Six Sigma process models [44] contain theories and tools with software process to improve and produce high quality products after implementing continuous improvement in SDLC. A gradual approach (OWPL) for software process improvement is
introduced to check various gradual assessments which allow gathering comprehensive knowledge depending on software in Small Enterprise Organizations (SMEs).

ABC (Activity Based Costing) model [45] used to reshape software process according to goals and objectives of the organization and secondly evaluate the performance of software process accordingly. IDEAL model is proposed in [46] based on the experience of large organizations which helps in the software improvement in small organizations.

El Emam et al's results show a correlation between higher maturity and meeting schedules, meeting budgets, improving quality, improving productivity; improving customer satisfaction; improving staff morale [47]. El Emam et al's more recent study confirms other desirable maturity features where high process capability is found to be associated with better project estimation and performance [48].

Paulk acknowledges that although the CMM directly addresses the human dimension only in training, it is not, “outside the scope of management responsibility or outside the scope of organizational needs” [49]. He adds that an effective process can empower people to work more effectively.

REFERENCES

[1] SEI (2010); http://www.sei.cmu.edu/about/about.html
[18] CMMI (2010); http://www.sei.cmu.edu/cmmi/