

Design of Software Process Improvement Model

Ankit Aggarwal

Department of Computer Engineering M.TECH
H.C.T.M Kaithal India (Haryana)

Alankrita Aggarwal

Assistant Professor Department of Computer
Engineering H.C.T.M Kaithal India

ABSTRACT

Software process improvement (SPI) approaches have been designed to produce changes at many levels, i.e. in the strategies, culture and working practices, of software development. Studies have shown that nearly two thirds of all SPI efforts have failed or fallen short of expectations. SPI Software process improvement (SPI) today is based mainly on a perception of software processes as artifacts and this perception has led SPI efforts to focus on perfecting such artifacts as a means to improve the practices of the people supposed to execute these software processes. Such SPI efforts thus tend to view the design of software processes as separate from their use. In this approach process designers are expected to provide process knowledge to software developers, and software developers are expected to provide experiences and problems to the process designers. This focus on software processes as artifacts implies an emphasis on formalization and externalization of process models possibly at the expense of the process knowledge in the heads of the process users.

Keywords

Software Process improvement models, Software engineering Institute, SPI

1. INTRODUCTION

Karl E. Wiegers explains process improvement as simply: “consistently apply the practices that give you good results, and change the practices that cause problems” [5]. 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[6]. 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.

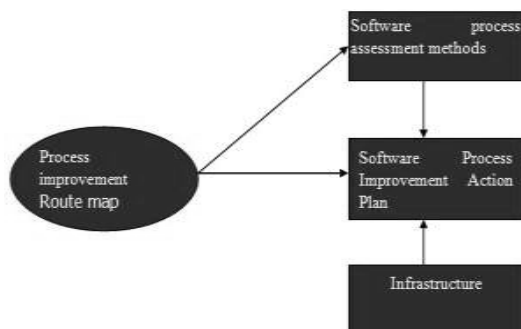


Figure1: Software Process Improvement Framework [7]

2. Software Process Improvement Models

2.1 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) [12].

2.2 Bootstrap

BOOTSTRAP is a European method for software process assessment and improvement that was developed to speed up the application of software engineering technology in the European software industry (Zahran, 1998). The BOOTSTRAP methodology is based on the CMM (discussed in the next section). However, it has been extended and adapted to include ISO 9000 guidelines and the European Space Agency software engineering standard (ESA-PSS-05). Unlike the CMM, BOOTSTRAP does not assume strict adherence to a distinct key practice model and allows the use of alternative approaches (Zahran, 1998). This has been a key factor in its success. In addition, BOOTSTRAP has proven suitable for use by all kinds and sizes of software development organizations. The main features of BOOTSTRAP are:

- Questionnaires for both site and project evaluation
- Uniform procedure and mandatory assessor qualification/training
- Constructive instead of a normative approach
- Open questions
- Immediate feedback and action planning

2.3 Cmm

The Capability Maturity Model (CMM) plays an important role in the software improvement efforts (SPI) of organizations worldwide (Zahran, 1998). The process was developed by the Software Engineering Institute at Carnegie Mellon University in 1986. Its goal is to improve, over time, the application of an organization's software technologies. The model provides a guide for organizations to select software process improvement strategies by facilitating the determination of current capabilities and the identification of critical issues.

The CMM process is made up of five well-defined levels of sequential development: initial, repeatable, defined, managed, and optimizing (Freedman, 2000). These maturity levels provide a progressive scale for measuring the maturity of an organization and its ability to use software technologies. Organizations that depend on formal rules, instead of individual performers, to manage software projects are A related article by Brodman and Johnson (1997) discussed a modified version of the CMM that was more suitable for small organizations and small projects. Problems typically reported with the CCM when used by these organizations were:

- Documentation overload
- Unrelated management structure
- Inapplicable scope of reviews

- High resource requirements
- High training costs
- Lack of need guidance
- Unrelated practices considered to be more “mature.”

2.4 Iso 9001

ISO 9001 is an international standard for quality assurance in design, development, production, installation, and service (Weissfelner, 1999). It is broken down into twenty elements. ISO 9001-3 relates to the development, supply, and maintenance of software. Almost 90 percent of the companies that completed ISO 9001 implementation reported improved internal documentation as one of the most important benefits of registration. Other benefits included higher product quality, greater internal quality awareness, and increased competitive advantage.

ISO 9001 is similar to the CMM in the following areas: emphasis on process, documented processes, practiced processes, address the “what” and not the “how” (Zahran, 1998). Differences between the two approaches occur in the areas of focus, dimensions, assessment and certification, coverage, supplier’s role, and level of detail. For example, an ISO 9001-compliant software organization would not necessarily satisfy the requirements for a level 2 in the CMM. However, it would satisfy most of the level 2 and some of the level 3 goals.

A related article reported the status of implementing an ISO 9001 compliant quality system in a small software organization (Demirors, Demirors, Dikenelli, & Keskin, 1998). Among the challenges encountered during the installation were a lack of guidance, action knowledge, maturity, and quality personnel. In response, ISO procedures were adapted in the following ways: role combination, shorter development cycles, enhanced early communication, simplified procedures, and minimized paperwork

2.5 Personal Software Process (PSP)

The Personal Software Process is a process-based method developed by the SEI for software engineers to use to apply process definition and measurement to their personal tasks (Humphrey & Over, 1997). Most important, the PSP shows developers how to manage product quality, meet commitments, and justify their plans with data. In addition, the PSP follows the concepts of the CMM. The key message of the PSP is that developers should use process management concepts to identify the methods most effective for them. A typical PSP course uses ten software development exercises, a structured sequence of defined processes, and five data analysis exercises to demonstrate the process.

An article by Silberberg (1998) reported on the successful application of the PSP to Ada software development. Examples of improvement made by the process included improved size and time estimating accuracy, reduced project time in the compile and test phases, and improved defect removal yield. The paper also provided a brief introduction to the PSP along with a comparison of the PSP with the CMM.

2.6 Team Software Process (TSP)

TSP is a defined method for a group of software developers to create quality software in an efficient manner (Hilburn, 2000). It provides process scripts, guidelines, tools, and techniques for a team to develop software applications. The process is based on an incremental model that divides effort into “development cycles.” Each cycle involves producing software that satisfies a subset of the total software requirements. In another article, Hilburn and Towhidnejad

(2000) discussed how TSP was used during a team software project in a junior level university course. The process provided students with clear, precise guidance and support, good data collection and analysis techniques, and an environment for building a successful software development team. The report concluded that the TSP was an excellent mechanism to emphasize software quality

2.7 Trillium

The Trillium model was initially designed for use with embedded software systems (e.g. telecommunications) and is based on the CMM (Coallier, Mayrand, & Lague, 1999). Its architecture differs from the CMM in the following ways:

- Architecture is based on roadmaps instead of key process areas
- A product rather than a software perspective
- Wider coverage of capability impacting issues
- Customer focus and a telecommunications orientation

Trillium is comprised of five levels (1-5). These are unstructured, repeatable and project oriented, defined and process oriented, managed and integrated, and fully integrated (Zahran, 1998). Trillium can be used in a number of ways. For example it can be used to benchmark an organization’s product development process against industry best practices or to self-assess and identify opportunities for improvement. In addition, it is useful in pre-contractual negotiations to select a supplier

3. COMPARATIVE ANALYSIS

3.1 CMM VS SIX SIGMA

S.NO	CMM	SIX SIGMA
1.	CMMI is a process improvement mode	Six Sigma is a process improvement methodology
2.	CMMI is a specific domain approach (software and system engineering)	Six Sigma is non domain specific methodology (initially was to improve manufacturing processes)
3.	Lack of standard metric.	Customer –centric.
4.	Define basic process infrastructure	Does not include any process model

3.2 CMM VS ISO

S.NO	CMM	ISO
1.	CMM describe about the software Engineering alone.	ISO describe both software and system Engineering.
2.	CMM more about software development	ISO work for software and hardware both.
3.	CMM is a process improvement approach, which have 5 maturity levels.	ISO (International organization for standardization) is the world largest developer for standard.
4.	CMM is a way to communicate capabilities.	The ISO is a way to communicate the process.

3.3 ISO VS SIX SIGMA

S.NO	ISO	SIX SIGMA
1.	ISO 900 is a quality management system. Which includes specialized quality management standards for specific industries	Six Sigma not a Quality management System. Such as ISO 9001 quality certification system.
2.	ISO 9001, with guidelines for problem solving and decision making, requires a continuous improvement process in place but does not indicate what the process should look like.	Six Sigma is a statistically-based process improvement methodology that aims to reduce defect to a rate of 3.4 defect s per million defect opportunities by identifying and eliminating cause of variation in business process.

4. PURPOSED WORK

LINEAR MODEL OF SOFTWARE PROCESS IMPROVEMENT

Linear model of Software Process Improvement is an approach pr method or both by which process improves and give better result rather than a normal process .By software process improvement a better and high quality product can be found within budget and time. The researcher suggests a software process improvement model .This model is a linear model.

This model consists of eight steps. There are following

1. Initial Step
2. Diagnosing Step
3. Analysis
4. Team Building and Meting
5. Plan Implementation
6. Inspection
7. Configuration
8. Documentation

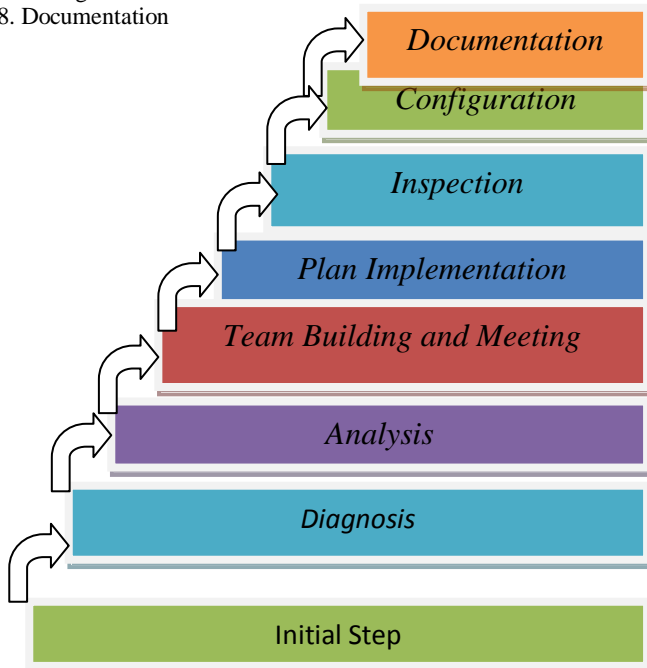


Figure 2: Linear Model of Software Process Improvement

4.1. Initial Step

At the initial level (level 1), an organization can be characterized as having an ad hoc, or possibly chaotic, process. Typically, the organization operates without formalized procedures, cost estimates, and project plans. Even if formal project control procedures exist, there are no management mechanisms to ensure that they are followed. Tools are not well integrated with the process, nor are they uniformly applied. Change control is generally lax and senior management is not exposed to or does not understand the key software problems and issues. When projects do succeed, it is generally because of the heroic efforts of a dedicated team rather than the capability of the organization

4.2 Diagnosing Step

In this phase, the SPI action plan is initiated in accordance with the organization’s vision, strategic business plan, lessons learned from past improvement efforts, key business issues faced by the organization, and long-range goals. Appraisal activities are performed to establish a baseline of the organization’s current state. The results and recommendations from appraisals and any other base lining activities will be reconciled with existing and/or planned improvement efforts for inclusion into the SPI action plan

4.3 Analysis

During the Analysis phase, the issues that the organization has decided to address with its improvement activities are prioritized; strategies for pursuing the solutions are also developed. The SPI action plan draft will be completed in accordance with the organization’s vision, strategic business plan, lessons learned from past improvement efforts, key business issues facing the organization and long-range goals. During the Establishing phase, measurable goals are developed from the general goals that were defined in the Initiating phase; these measurable goals will be included in the final version of the SPI action plan

4.4 Team Building and Meeting

Teams usually work best with a small group of people. Smaller numbers make team administrative tasks easier, such as deciding where and when to meet. Meetings are generally shorter when fewer people need to speak. Small size also makes it easier to develop a common purpose with mutual goals and mutual accountability, which is so important for good teamwork. In addition, a small group of people avoids the "herd" mentality of large groups. In a large group people tend to go along with popular opinion rather than thinking for themselves. In general, the larger the group of people, the harder it is for the group to work well together. Complimentary skills are also important for a team. Necessary skills include: technical, problem-solving, decision-making and interpersonal skills. Technical skills are needed to provide the expertise to meet the goals of the team. The other skills are necessary for the team to work effectively together.

4.5 Plan Implementation

When the plan establish according to problem, then create the solution according to plan. First make a proper solution on basic of experience and knowledge as well as skill of assessment team. Then again gives presentation to the higher authorities. if they give some suggestion for modification in the solution ,then modify the plan accordingly and if they are comfortable for solution than start the implementation. They choose a better solution. Before implementing the plan it should be disuses with the senior management and organization. When they are agree on solution, start working on the solution, all solution does not apply in one step: it is a step by step process.

4.6 Inspection and Configuration Management

An inspection is generally an organized examination or a formal evolution exercise. It involves the measurement, test and gauges applied to certain characteristic in regard to an object or activity. The result is usually compared to specified requirements and standards for determining whether the item or activity is in line with these targets. Inspection is usually non-destructive. It is not guarantee that the work plan and implementation is correct. For inspection a team is formed, in which members are moderator, reader and inspector. A formal meeting is conducted. Role of the moderator is to conduct the meeting and ensure the subject of the meeting. Role of the moderator is like a leader of the inspection team..So moderator should be highly qualified and skilful. Moderator tells the team how the inspection starts and leads them. Inspection team takes the interview of assessment team.

4.7 Documentation

Documentation is an umbrella activity in this model. This is going on side by side throughout process. In each phase documentation is needed for the future work so that anybody can reuse it as well as future work can be done. The document associated with a software project and the system being developed has a number of as Role of the moderator is associated requirement:

They should act as a communication medium between number of the developed team.

They should be a system information repository to be used by maintenance engineer.

They should provide information for management to help them plan, budget and schedule the software development process.

5. COMPARISON WITH SPIM TO OTHER MODEL

5.1 Comparison between CMM and LMSPI

S.NO	CMM	LMSPI
1.	The main problem with the CMM model is that CMM give the maturity level. That an organization across the level then it's enter the next level. But CMM does not specify the implementation.	In LMSPI model implementation is defined and a proper document is prepared for every process.
2.	CMM works on software	LMSPI works on software and organization
3.	CMM is a goal. Being just as stamp of approval.	LMSPI model is a method.
4.	CMM is works only a repeating task.	But LMSPI model is not only repeating task but also whole. If problem is change then this model work doe to cyclic model.

5.2 Comparison between SIX SIGMA and LMSPI

S.NO	SIX SIGMA	LMSPI
1.	SIX SIMA i is a statistically-based process improvement methodology that aims to reduce defect to a rate of 3.4 defect s per million defect opportunities by identifying and eliminating cause of variation in business process	LMSPI is a planned methodology of continuous improvement Methodology
2.	SIX SIGMA is a iterative methodology reduced the defect one by one	LMSPI model is a linear model that improves the process one by one step.
3.	SIX SIGMA methodology works on two approach DMAIC .DMADV	LMSPI works as a full flash model

6. CONCLUSION

The proposed linear model of software process improvement (LMSPI) improves the process in a traditional way. This model is an iterative model. The LMSPI model does the process improvement in a stepwise way. It covers user requirements software quality assurance, and organization point of view. Many of the factors can be found in the organization from the LMSPI model like management commitment and teamwork were strengthened.

SPIM model cover the some limitation of existing model (CMM, SIX SIGMA).For example, the main limitation of CMM is key practice describes. "What to do "but does not prescribe "how to do". LMSPI model describe the implementation and prescribe how to do. The LMSPI model does not necessary to work for the repeatable task .When the new problem come it will work for that also. LMSPI is a flexible model. If there is a change in the process, LMSPI covers all the aspect of the changing of process due to linear model.

7. FUTURE WORK

The proposed model in the future can be implemented after finding key process area (KPA), then creating the templates and forms after conducting exhaustive survey. Success of this model depends on the depth of the survey. The Key process areas and capability area of LMSPI model can also be defined and extended ion future LMSPI model reduces the risk as much as possible but it has also some limitations. It takes so much time in presentations and conduct the meeting. In the future work a lot of work can be to remove the timing problem of the LMSPI model

8. REFERENCES

- [1] Brodman, J., & Johnson, D. (1997). A software process improvement approach for small organizations and small projects. Paper presented at the International Conference on Software Engineering, Boston, MA.
- [2] Coallier, F., Mayrand, J., & Lague, B. (1999). Risk Management in Software Product Procurement. In K. Emam. & N. Madhavji (Eds.), Elements of Software

- Process assessment and Improvement (pp. 23-44). Washington, DC: IEEE Computer
- [3] Dr. Jyotsna Sengupta, Rupinder Kaur, “Software Process Models and Analysis on Failure of Software Development Projects”, *International Journal of Scientific & Engineering Research*, Feb 2011.
- [4] M.P. Thapliyal, Pratibha Dwived, “Software Process Improvement in Small and Medium Software Organizations of India”, *International Journal of Computer Applications*, 2010
- [5] McFeeley, IDEAL SM: “A User’s Guide for Software Process Improvement. Handbook” CMU/SEI- 96-HB-001. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PE, USA.
- [6] Donna K. Dunaway, Steve Masters, “CMM SM -Based Appraisal for Internal Process Improvement (CBA IPD): Method Description”, April 1996.
- [7] Watts S. Humphrey, “Introduction to Software Process Improvement”, 1992.
- [8] B. Curtis, W. Hefley and S. Miller, *People Capability Maturity Model*. Addison-Wesley: Boston, MA, USA, 2001.
- [9] B. Franken, How to Explain Six Sigma by Using the Profit Triangle .Retrieved May, 2007 from www.iSixSigma.com/library/content/.
- [10] F. Coallier, R. McKenzie, J.F. Wilson and J. Hatz, *The Trillium Model, Version 3*, Bell Canada, December, 1994.
- [11] G. Eckes, *The Six Sigma Revolution, How General Electric and Others Turned Process into Profits*. USA: John Wiley & Sons, Inc., 2001
- [12] McGuire, E., & Randall, K. (1998). *Process improvement competencies for IS professionals*. Paper presented at the Conference on Computer Personnel Research, Boston, MA.
- [13] M.B. Chrissis, M.D. Konrad, and S. Shrum, *CMMI for evelopment: Guidelines for ProcessIntegration and Product Improvement*, Third Edition, 2011.
- [14]B.P. Gallagher, M. Phillips, K.J. Richter, and S. Shrum, *CMMI-ACQ: Guidelines for Improving the Acquisition of Products and Services*, 2009.