ISSN:2320-0790

An International Journal of Advanced Computer Technology

A Systematic Review on the Impact of Metrics in Software Process Improvement

Simran Jaitly, Anil Kumar Mishra, Dr. Latika Singh
PG Scholar, ITM University, Gurgaon, Haryana
Asst. Professor, ITM University, Gurgaon, Haryana
Associate Professor, ITM University, Gurgaon, Haryana

Abstract: Software Process Improvement is an act of changing the ongoing software development and maintenance process to achieve basic business goals. It is a sequence of catalogued activities required to develop and maintain the software within technical and management schema. Software metrics provide a quantitative basis for planning and predicting software development processes and their required improvement strategies. This research paper focuses on the impact of software metrics on software process improvement. Moreover, many metrics and tools have been developed; promoted and utilized resulting in remarkable successes. It also examines the realm of software engineering to see why software metrics are needed and also reviews their contribution towards software process improvement and its quality.

Index Terms: Software Process Improvement; Software metrics; Software quality; CMMI; OOSPICE

INTRODUCTION

I. Software Process Improvement

A software process can be defined as the sequence of steps required to develop or maintain a software aiming at providing a technical and management framework for applying methods, tools, and people to the software task. It is the art and science of changing an organizations software process to build better software, attaining some basic business goals [1]. It provides mechanisms to software organizations for evaluating their extant processes, identifying possibilities for improving as well as implementing the improvements and evaluating the impact of those improvements.

As a matter of course, the ultimate goal of SPI in organizations is to provide a Return on Investment (ROI) for the organization through the improvement activities that yields more resources than is spent on them. Reportedly, ROI has been credited for various SPI achievements, including improved efficiency of the development process and reduction of total software costs, increased quality of the end product, higher predictability of cost and schedule, and increased level of reuse.

The objective of software process improvement is to set methods in order to improve the development process within the organizations framework. It's important that the organization analyzes the process structure and identify the main reasons behind their failure and apply the improvement strategies wherever necessary. Firstly, the objectives of SPI initiative needs to be determined and then, selecting the assessment techniques aligned with tracking the progress towards achieving the set objective is the second task to be done.

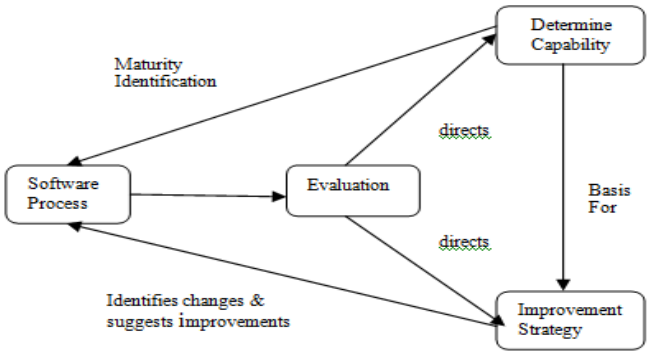

Fig1. Software Process Improvement Framework

SPI Framework assesses the maturity of a software process. It is a set of characteristics that must be present if an effective software process is to be achieved. It is a method for assessing whether those characteristics are present & checks for a mechanism that summarizes the results. It pertains to a strategy for assisting a software organization for implementing SPI.

To improve the quality of software and organizations software development productivity and capability, various approaches

have been developed and deployed in the software fields. With the use of process assessment tools, the process capability can be identified and based on the results from the assessment; enhancements of the processes can be done.
Software Process Improvement Tools have been developed and deployed in various existing fields. The following table provides an insight into the existing tools. In this paper, we have searched and listed down some existing SPI Tools that are into extant in various software organizations.

Table1. Existing SPI TOOLS

NAME	CATEGORY	OBJECTIVES	METHODOLOGY
Agile Assessment Method	Untraditional Approach	determines how the AAS Models help teams and organizations to improve their processes.	methods have been proposed: Assessing Agility; Agile Assessment Approach; Nokia Test; Comparative Study; Other Approaches
SPAILS	web-based tool	Provides gap analysis information and proposes individual SPI measures to achieve the process improvement objectives.	Representation from organization → Representative from role of each project → Affirmation for conflict responding → Generate project report → Generate organization report
RIA Based Tool	Web-based tool	Provides a new process structure & elements to document the process.	• Self Evaluation • MoProSoft Model
ISO Based Tool	Manual Tool	conducts the process assessment as ensuring a project's compliance with specific standards in software development organization.	Retrieval and communication with the skilled in order to offer the way of appropriate process mentor. SPI strategies are based on 2 types of evaluation results: With results of self-evaluation by practitioners & With results of internal assessments.
Six Sigma Tool in PSP/TSP	Team-Based Tool	Deploying PSP/TSP in conjunction with Six Sigma, leads to improved project performance & continuous process improvement.	The identified team-based tools are affinity diagram, SQFD (Software Quality Function Deployment), Kano analysis and SWFMEA (Software Failure Mode and Effect Analysis).
DuoTracker	Process Driven Tool	track and analyze s/w defects for s/w process decision making	classify defects in a manner that makes analysis at both organizational and individual software processes meaningful

II. **Software Metrices**

Software metric is the measure of any property of a piece of software or its specifications. Since quantitative measurements are essential, there is a continuous effort by computer practitioners to bring akin approaches to software development. The intent is to obtain objective, reproducible and quantifiable measurements, which may prove to be beneficial for software assessment. It is a standard for measuring and evaluating a quantifiable entity.

In this paper we present an overview of various software metrics that are currently into use for analyzing the software process and estimating the objectives for process improvement technique to be applied onto the existing process.
1)*Product Metrics*: Product metrics assists developers to better understand the software attributes and to assess its quality based on some clearly defined rules. It benefits the software engineers to gain an insight into the design and development of the software. It focuses on specific attributes of software work products resulting from various stages of SDLC. They are also used to identify the system components which are of sub-standard quality.

2)*Process Metrics*: Process metrics can be used for improving software development and maintenance. It provides indictors leading to long term process improvement. Private process metrics are only known by the individual or developing team. Public process metrics enable organizations to make strategic changes to improve the software process. [21]

3)*Object Oriented Metrics* : Object-oriented measurements are being used to evaluate and predict the quality of software. Often, these metrics are used as an early indicator of the externally visible attributes, because they cannot be measured until later stages in the software development process[22].

4)*Project Metrics* : Project metrics describes the project execution and characteristics. It enables the project manager in assessing the status of the ongoing project, tracking potential risks and to uncover problems before they go critical.

5) Customer Satisfaction Metrics: Customer satisfactionis often measured by customer survey data obtained via the five-point scale: Very satisfied, Satisfied, Neutral, Dissatisfied, Very dissatisfied. Satisfaction is obtained with the overall quality of the product and its specific dimensions through various methods of customer surveys.

6) *Cohesion & Coupling Metrics*: Coupling and Cohesion metrics determine whether combining the metrics is supportive in the task of predicting bugs in large open-source software. Object oriented design and analysis is an ongoing technique of software engineering technologies, due to which many different object-oriented coupling and cohesion metrics have been developed.

7) *Design Metrics*: Design Metrics are utilized in measuring

the complexity and goodness of a design. They allow the software engineer to distinguish between designs and pinpoint design weaknesses, particularly with a view to minimize the development effort. A design measure is obtained based upon the information flows between modules that are empirically validated by analyzing a software system.

III. CMMI Based Quality Metrics

The Capability Maturity Model Integration (CMMI) is an eminent and accepted process improvement approach, which also proves to be a process quality measurement. It is widely applicable. The current process quality is estimated by the means of the CMMI Appraisal Method SCAMPI which needs the appraisers' experience to investigate the processes in expanse for assessment results, that are highly accurate. The assessment result represents complacency for all those process areas that fulfill the respective defined goals and practices.

Based on CMMI, a software metrics process model is discussed in this paper. SPI methodology, followed in CMMI can be defined as definition of sequence of steps, tools and techniques to be implemented to plan and implement the improvement activities. They emphasize the importance of planning SPI activities, naming the key activities and concepts and describing their relationships [17].

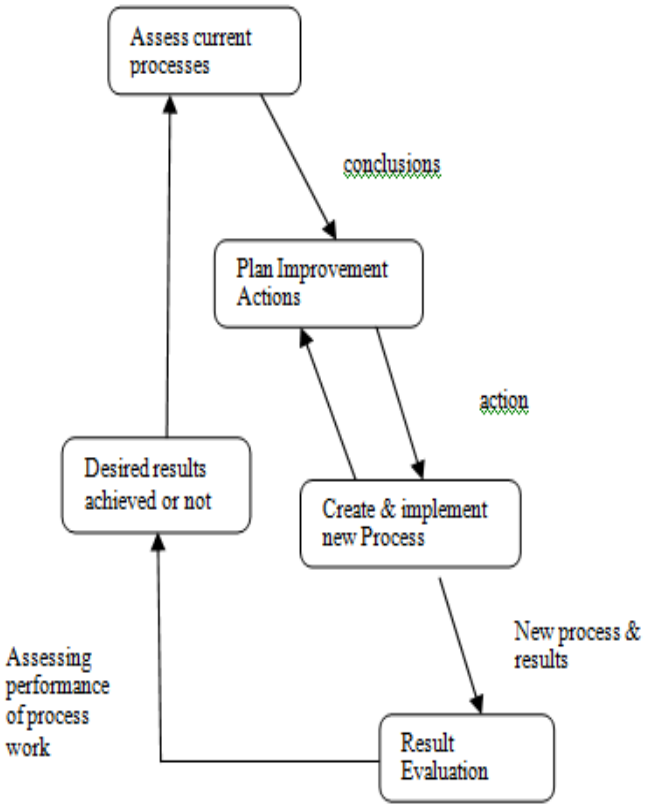

Fig.2 SPI Lifecycle in CMMI

Table2. Existing CMMI Metrics

METRICS	GOAL	METHODOLOGY
Goal Driven Metrics	Objectively measure the program condition as on Project Management Process Improvement & Organizational vision.	Measure the success in achieving the goals and sub-goals & identify metrics that have been overlooked.
Goal Quality Metrics	GQM Analysis, Metric Definition, Data Collection, Data Analysis & Reporting.	FAST
FQIMM	Integrates s/w metrics set, linguistic variables, & interval of confidence.	Lead Appraiser Method: evaluates description of questionnaire, interview & document by appraisal requirement.
Effort Slippage	Improve estimation	Estimating efforts based on requirements & planning based estimation
Defect Density	Minimize defects	Detecting defects at an earlier stage & minimize acceptance test defects.
Rework Effort	Reduce cost	Identify non-value added activities & detecting early defects to reduce rework.
Traceability Metrics	Requirement management	Maps the verification of implemented requirement & test case verification.

IV. OOSPICE Project with Metrics

The OOSPICE Project1 (Software Process Improvement and Capability dEtermination for Object Oriented/Component Based Software Development), which focuses on the industrial practice of CBD and delivers new processes, methods and tools that can be implemented practically. It is a combination of principles of empirical software engineering major concepts: CBD, object-oriented development, software process assessment and software process improvement. The main objectives include a consolidated CBD process model and metamodel foundation, a CBD assessment methodology consisting of an assessment model, an assessment method and an assessment tool.

An accepted principle in software engineering is that the product quality is determined by process quality. This forms the basis for process assessment and improvement approaches like CMM and SPICE. The updating and improving technology for CBD, also focuses on OOSPICE that requires a detailed analysis and allocation of a wide variety of currently offered approaches for yielding an improved process. Based on the identified best-practice approaches, the aim is to formulate suitable process models and an assessment approach that suits the project improvement plans.

SPICE framework is required to perform the analysis and assessment of the software process, on the basis of these assessments information about the strengths, weaknesses and capability of the process to achieve its goals is induced. The

term "process assessment" can be defined as "A disciplined assessment of an organization's software processes with respect to the process model or variant model as described in the International Standard." [20].

It enacts like a bridge between process and process engineering, on one hand, and process and process capability assessment on the other hand, that brings these two sub-disciplines of software engineering together. Technological innovation in the OOSPICE Project is in the following areas: the production of an unified process model and an underpinning metamodel for CBD, addressing the variations observed; the development of a CBD methodology conforming to the unified process model for CBD; the definition and validation of an assessment technique for CBD; the assimilation of user trials of both assessment and development methodologies, etc.

In relation to CBD of systems, the metrics seeks to answer the following questions:

• Does the development of new or modified process reference models require specific and unique aspects for development?

• Does the process assessment involve specific issues that can validate process improvement and capability determination in CBD?

In the Object Oriented context, a modern tailorable framework OPEN provides the basis for a successful process development and utilizes the implemented metrics within the methodology for process improvement. OPEN (Object-Oriented Process, Environment and Notation) includes tasks and techniques that are useful in the early stages of SDLC. It is a process-focused methodological approach for component and software development. Perhaps, it is a framework defined by a process metamodel that is not rigidly specified. Thus, it is well suited for CBD.

A framework for a CBD programme is the "GOAL/QUALITY/METRIC" paradigm. GQM evaluates the goals and questions, thus leading for choosing a legit metrics. The problem with the implementation of this approach to OO metric is that the set of available metrics may be empty. Thus, a eulogistic activity is the identification of a large number of "potentially profitable" metrics. The GQM is useful when an empirical relational system is identified over a characteristic, to select the most befitting metrics.

Currently, there are some internal OO metrics which are used to measure products. Another set of metrics consists of those which involve some external considerations. Internal metrics are those which can devise easier ways of estimations; and external metrics in contrast have major challenges. Thus, internal metrics have been preferred over external metrics as direct surrogates without any intervening model and theory.

V. Future Scope & Work

Realizing the rising demand for the improvement in software quality, it can be concluded that in the impending years, software process improvement will be a prerequisite in software organizations. A number of metrics are proposed and exercised for measuring the quality of the improved processes before its implementation. Future research directions might include improvement in existing process and metrics based on the nature and magnitude of the problem statement. There is a scope for development of various tools to support software process improvement that may help in reducing time, effort and cost of the process improvement in a consistent manner.

Two basic research areas have been identified through this study:

- The role of metrics in managing and improving software development processes
- Methodology applied for improvement in collecting, interpreting, distributing and utilizing data

VII. Conclusion

The summary position will provide an intelligible view of the research conducted, including the overall results of the improvement efforts in process improvement. It will also address the research questions and might contain the practical guidelines for other processes to follow.

VIII. Acknowledgment

This research is supported by the Department of Computer Science and Information Technology at ITM University.

IX. REFERENCES

[1] http://www.slideshare.net/bilalhashmishah/software-process-improvement-12777417

[2] Chayakorn Piyabunditkul, Nithipat Wongchingchai, Apinporn Methawachananon: Step forward CMMI-Project Management by optimized Scrum. Proceedings of the 5th International Conference on Project Management, The Society of Project Management (SPM), Japan, pp 688-698, ISBN 978-4-902378-19-1, 2010

[3] SCAMPI Upgrade Team: Appraisal Requirements for CMMI. Version 1.2 (ARC, V1.2) Technical Report , CMU/SEI-2006-TR-011, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2006.

[4]http://www.pearsonhighered.com/samplechapter/0201729156.pdf.

[5] http://www.wohlin.eu/Articles/ICGSE11.pdf.

[6] Barbara Kitchenham, Shari Lawrence," Quality: The Elusive Target", IEEE Software-Vol. 13, No. 1: JANUARY 1996, pp. 12-21.

[7] Henrike Barkmann, Rudiger Lincke and Welf L owe, "Quantitative Evaluation of Software Quality Metrics in Open-Source Projects".

[8] Avison, D., F. Lau, M. Myers, P. A. Nielsen (1999). "Action research", Communications of the ACM, Vol 42, no. 1, pp. 94-7.

[9] Basili, V., D.M. Weiss (1984). A Methodology for Collection Valid Software Engineering Data, IEEE Transactions on Software Engineering, Vol 10, no. (November 1984), pp. 728-738. Basili, V. (1994). The Goal Question Metric Paradigm, Encyclopedia of Software Engineering - 2 Volume Set, pp 528-532, Copyright by John Wiley & Sons, Inc., with Gianluigi Caldiera and Dieter H. Rombach.

[10] Checkland, P., J. Scholes (1993) Soft systems methodology in action. Wiley, Chichester.

[11] Dekkers, C.S. (1999). The Secrets of Highly Successful Measurement Programs. Cutter IT Journal, vol 12 no. 4, pp. 29-35

[12] Pankaj Jalote. CMM in Practice: Process for Executing Software Projects at Infosys. Addison- Wesley, 2000.

[13] Xu Ru-zhi, Qian Leqiu. Research on Metrics- based Software Process Control Optimization. Journal of Electronics, 2003, 31(12A):1206- 1209.

[14] CMMI-SE/SW Version 1.1, Copyright 2002 by Carnegie Mellon University, January 11, 2002. www.sei.cmu.edu/cmmi.

[15] Xu Ruzhi, Nie Peiyao. Optimizing Software Process based on Risk Assessment and Control. Proceedings of the 5th International Conference on Computer and Information Technology. Shanghai, Sep. 2005. IEEE Computer Society, pp896-900.

[16] Florac WA, Carleton AD. Measuring the Software Process: Statistical Process Control for Software Process Improvement. Addison Welsley, 1999.

[17] http://link.springer.com/chapter/10.1007%2F978-3-642-22206-1_9#page-2.

[18]https://gupea.ub.gu.se/bitstream/2077/27885/1/gupea_2077_27885_1.pdf

[19] CMMI (2010); http://www.sei.cmu.edu/cmmi/

[20]] SPICE Documents: http://www.sqi.gu.edu.au/spice/what.html

[21]https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CEYQFjAD&url=http%3A%2F%2Fhighered.mcgrawhill.com%2Fsites%2Fdl%2Ffree%2F0073375977%2F673802%2Fchapter25.doc&ei=JeZUpeXN42Jrgfms oGIDg&usg=AFQjCNHp7U3SA6PjyHws8BAj-VMHf99tYg&sig2=zVpUepuSAwvB1XyzKXqCLA

[22] http://agile.csc.ncsu.edu/SEMaterials/OOMetrics.htm