Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Software Process Improvement Model

Mamta Shelpar[1], Sona Malhotra[2]
[1]Research Scholar, [2]Assistant Professor,
U.I.E.T, Kurukshetra University Kurukshetra,
Haryana, INDIA

Abstract: - Software process improvement is recognized as an important Part of the software development life cycle. Several models have been developed to assist organizations to evaluate and improve their software Development processes and capabilities. The development of software products is a complex activity with a large number of factors involved in defining success. Measurement is a necessary essential, for software process improvement. However few guidelines exist for logical planning of measurement programs within software projects and for multi-project selection. Software development process presents the task or activities used to produce a software product. A software development life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. There are tons of SDLC models each model contains specific activities to be performed to develop software.

Keywords: Software development, Software process improvement models.

I. INTRODUCTION

Software process improvement and measurement is becoming one of the main methods to solve "software crisis". Software Process Measurement (Software Metric) defines the process of software development, collects and analysis data, that is quantization process of continuous improvement, is important basis of making plan, executing process, implementing control. The standardisation school of SPI is based on the premise that developing software in a more well-defined and predictable way results into a higher quality software within cost and time schedules. This approach is focused on certification and comparisons with a process maturity model. CMM is in fact a framework to evaluate or assess the maturity of software engineering organisations. Organisations developing software in planned and documented ways is placed at a higher maturity level than other organisations.

II. SOFTWARE PROCESS IMPROVEMENT

Software process improvement is a deliberate, planned methodology following standardized documentation practices to capture on paper (and in practice) the activities, methods, practices, and transformations that people use to develop and maintain software and the associated product.

- Identify the current process
- Identify the current status of the organization
- Identify the strength and weakness of the organization
- Analyze where the changes are needed
- What will be the effect of these changes?
  - For long term business
  - Reuse
  - Better quality

SIX SIGMA

Six sigma strategies were developed by Motorola in the early 1990s. Six Sigma is based on statistical approach which does the improvement by historical data and by calculation of mathematical formulas. The goal of the six sigma is to detect the defect and reduce the defect. [1] Six sigma means a company tries to make "error-free product 99.9997% of the time a minuscule 3.4 errors per million opportunities". SIX SIGMA has six stages and reduces the defect step by step. Six Sigma is usually related to the magic number of 3.4 defects per million opportunities.

Limitations of Six Sigma: The limitation of Six Sigma can be given as following.

- Six Sigma is a statistically-based process improvement methodology.
- Often it is very difficult for small companies to take employees away from their regular duties in order to be trained in Six Sigma. If employees are not available to give their services, the company loses money due to a reduction in productivity.
- Six Sigma focuses on prioritizing and solving specific problem which are selected based on the strategies priorities of the company and the problems which are causing the most defects.

III. TOTAL QUALITY MANAGEMENT (TQM)

TQM is a management philosophy, a paradigm, a continuous improvement approach for doing business through a new management model. The TQM philosophy evolved from the continuous improvement philosophy with a focus on quality as the main dimension of business. TQM is made by the combination of three alphabetical letters. These letters are the following [2]

Total- involving the entire organization, supply chain, or product life cycle.

Quality- the literal definition of quality

Management- the system of managing with steps like plan, organization, control etc.

Concept of TQM: TQM is basically field of management. TQM manages the quality from a different perspective. These are following:

- Quality evolution
- Quality dimensions
- Quality management

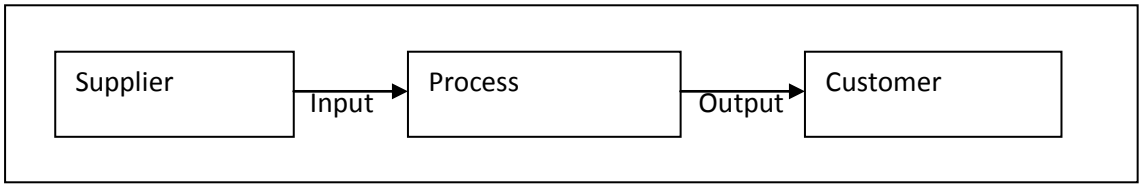

Fig.1 A process view for TQM

CMM- The Capability Maturity Model (CMM) plays an important role in the software improvement efforts (SPI) of organizations worldwide .The process was developed by the Software Engineering Institute at Carnegie Mellon a progressive scale for measuring the maturity of an organization and its ability to use software technologies, the application of an organization's software technologies. A related article by [3] 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:

1. Documentation overload
   - Inapplicable scope of reviews
   - High resource requirements
   - High training costs
   - Lack of need guidance
   - Unrelated practices

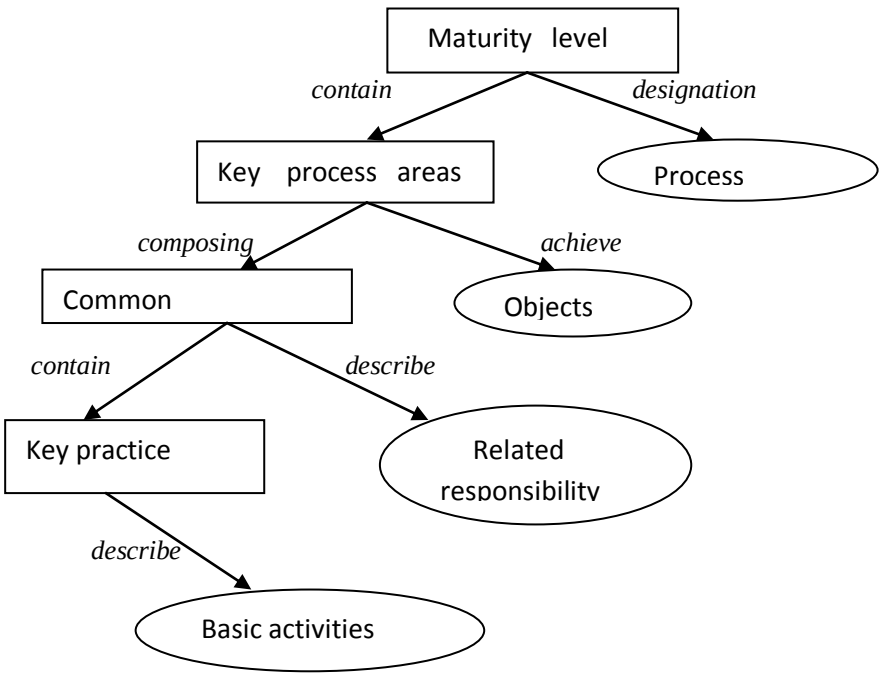

Fig.2 Architecture of CMM

Limitations of CMM

- The main limitation of CMM is key practice describe "what to do" but does not prescribe "how to do"
- CMM is work only a repeating task.

- CMM is goal. Being used just as stamp of approval.

IV. IDEAL MODEL

The IDEAL model is a software process improvement (SPI) model, published in 1996 by the Software Engineering Institute (SEI) of Carnegie Mellon University. The goal of SPI is to improve the development process of software. The IDEAL guidelines are for those organizations that initiate a SPI for the first time and even for those that are continuing an already established SPI. The IDEAL model contains five phases:

A. Initiating Phase:

The Initiating Phase is the origin of the IDEAL model. The initial improvement is established, the roles and responsibilities for the infrastructure are initially defined, and initial resources are assigned [4].Build process infrastructure.

B. Diagnosing Phase:

The strategic business plan, lessons learned from past improvement efforts and other strategic factors, the SPI action plan is initiated. These results are compared with concluded and planned process improvements. The findings are included in the SPI action plan. Thus the diagnosing Phase is for determining the status and the future goals.

C. Establishing Phase:

The Establishing Phase is where the goals of the process improvement effort are prioritized and the approaches are developed. Also, the SPI action plan, which was initiated in the Diagnosing Phase, will be completed. This includes deriving measurable goals from the main goals.

D. Acting Phase:

In the Acting Phase solutions for improvement are created, piloted and then deployed in the entire organization. The Acting Phase is for doing the work required to reach the defined goals. Research and develop solutions to process problems, expand successful process.

E. Leveraging Phase:

The aim of this phase is to make the next pass through the IDEAL model more effective. . The Leveraging Phase is learning from what has been done for the next iteration of the process Improvement cycle. Apply the lessons learned to refine the SPI process.

Limitations of IDEAL Model: Ideal model is also a continuous model. But it is a full method such that there is no recovery.

Educational software development model is software engineering ideas, methods and procedures in educational software development practices.[5] There are many software development models, such as the waterfall model, incremental model rapid prototyping model, spiral model.

STEPS: a) Needs analysis: Using software engineering, software requirements analysis from the perspective of the educational software development need to do necessary analysis of the development process.

b) Design education: educational software service for students, so we need to consider how to give full play to the characteristics of the multimedia computer to suit

c) Software design: Design includes internal structure, software structure, and external structure design.

d) Educational software prototype: Rapid formation of prototype body and actually be able to move. This is a necessary condition for the prototype development. In this step, we full use of the advantages of middleware and develop educational software prototype in the shortest time.

e) Education trial: education trial is the basis of customers evaluate. It provides evaluation standards and reference.

f) Customers evaluate: This is the most important part of the model. It determines the success or software products need to modify.

g) Prototype reprocessing: Developers continue to make changes and additions to the prototypes, until the customers are satisfied.

h) Educational software operation and maintenance: For large-scale educational software, maintenance is indispensable. Maintenance workload may be greater than the software development effort.

PIT-ProcessM – KEY CONCEPT

The PIT-Process meta-model describes formal process elements that can be applied to construct software development processes. PIT-ProcessM includes two complementary views:

1. The static view and the dynamic view. The *static view* shows the structure of process concepts, their characteristics and the relations between them. The *dynamic view* identifies a structure that is specific to temporal development circumstances such as how work is to be organized over time.

2. *Process* is made up of a number of process elements (several disciplines or other existing processes).

3. *Discipline* is a particular organization of process activities according to a common "theme". Each discipline includes several practices which are described through its *Activities*, *Work Products* and *Roles*. [6]

4. *Activity* is performed to produce (e.g., create, evaluate, iterate, and maintain) *Activity* is owned by a *Role*. Beside the owner, there are additional *Roles* that also participate in the activity.

5. *Phase* is the time interval that provides a macro organization of *Activities* to be performed and it has the main goal, to produce a final set of *work products*.

6. *Iteration* is a scheduled part of a *Phase* leading to an increment towards the final phase goal. A *Phase* can have one or more *Iterations.*

7. *Work Product* is a significant element (e.g. document, diagram, model and application) that is produced or consumed by process *Activities*.

8. *Work Product Kind* represents several types of work products. *Work Products* can be classified in several types which identify the kind of input or/and output expected in an activity (such as text documents, UML models, SQL tables, executables, code libraries, and so on).

9. *Role* performs one or more *Activities* in order to produce, either directly or indirectly, versions (or new) of one or more *Work Products.*

V. ASSESSMENT INSTRUMENT

Each SEI level has several associated key process areas. The progress assessment instrument lets you determine the scores associated with the SEI level our organization is trying to achieve. [7]

1. Commitment to perform
2. Ability to perform
3. Activities performed
4. Monitoring implementation
5. Verifying

Implementation Expanded and grouped these themes under three primary evaluation dimensions and developed criteria for them:

A. Approach: criteria here the organization's commitment to and management's support for the practice, as well as the organization's ability to implement the practice.

B. Deployment: The breadth and consistency of practice implementation across project area are the key criteria here.

C. Result: Criteria here are the breadth and consistency of positive results over time and across project areas.

TICK IT

Tick IT model is supported by the UK and Swedish software industries, has been to stimulate software system developers to think about:

- What quality really is in the context of the processes of software development?
- How quality may be achieved, and How quality management systems may be continuously improved.[8] Although certification of compliance to ISO 9001 is a contractual requirement for software suppliers in certain market areas,
- It should be a product of the more fundamental aims of quality achievement and improvement, and the delivery of customer satisfaction. With regard to certification itself,
- Increase the communication between operators and management.
- Detection of Problems and to decrease the recurrence of problems.
- Cause and Effect Diagram
- Run Chart
- Pareto Diagram
- Scatter Diagram
- Histogram
- Flow chart
- Control Chart

VI. CONCLUSION:

The work in this paper describes the common way to improve the software development process and existing improvement models also. We have discussed models like CMM etc, existing models have some limitations. These models provide the steps of software process improvement methods and models so that to get understand the concepts of improvement and its procedures.

REFERENCE:

[1] D.F. Rico**," *ROI of Software Process Improvement"*.** J. Ross Publishing, 2004.

[2] Dr. JyotsnaSengupta, RupinderKaur, **"Software Process Models and Analysis on Failure of Software Development Projects",** International Journal of Scientific & Engineering Research, Feb 2011.

[3] Bharti Sharma, Neeraj Sharma and Neeshu Sharma **"Software Process Improvement:A Comparative Analysis of SPI models"** second International Conference on Emerging Trends in Engineering and Technology, ICETET, 2009.

[4] Ruth Klendauer, Axel Hoffmann, Jan Marco Leimeister and Marina Berkovich, Helmut Krcmar **"Using the IDEAL Software Process Improvement Model for Implementation of Automotive SPICE"** 2012 IEEE CHASE 2012, Zuric.

[5] Huiqing Zhang , Huini liu **"Educational software process improvement model and strategy"**, International Conference on Computer Science and Information Processing (CSIP) 2012.

[6] Paula Ventura Martins**" PIT-Process: A Software Process Improvement Meta model"** Seventh International Conference on the Quality of Information and Communications Technology 2010.

[7] MICHAEL K . DASKALANTONAKIS**, "Achieving Higher SEI Levels"** Motorola, IEEE 1994.

[8] Dutta, Soumitra, Michael Lee, and Luk Van Wassenhove, **"Software Engineering in Europe: A study of Best Practices"**, IEEE Software, vol. 16, no. 3 (May/June 1999).

.
.

