Thakur Educational Trust's (Regd.)

THAKUR RAMNARAYAN COLLEGE OF ARTS & COMMERCE

ISO 21001:2018 Certified

PROGRAMME: B.Sc (I.T)

CLASS: S.Y.B.Sc (I.T)

SUBJECT NAME: SOFTWARE ENGINEERING

SEMESTER: IV

FACULTY NAME: Ms. SMRITI DUBEY

UNIT V

Chapter 1 – Process Improvement

Concepts:

Introduction

Software Product Quality Factors

Process Classification

Process Measurement

Process Analysis and Modeling

Process Change

The CMMI Process Improvement Framework

Introduction

Process improvement means understanding existing processes and changing these processes to increase product quality and/or reduce costs and development time.

Two quite different approaches to process improvement and change are used:

1. The process maturity approach, which has focused on improving process and project management and introducing good software engineering practice into an organization. The level of process maturity reflects the extent to which good technical and management practice has

been adopted in organizational software development processes. The primary goals of this approach are improved product quality and process predictability.

2. The agile approach, which has focused on iterative development and the reduction of overheads in the software process. The primary characteristics of agile methods are rapid delivery of functionality and responsiveness to changing customer requirements.

Process characteristic	Key issues
Understandability	To what extent is the process explicitly defined and how easy is it to understand the process definition?
Standardization	To what extent is the process based on a standard generic process? This may be important for some customers who require conformance with a set of defined process standards. To what extent is the same process used in all parts of a company?
Visibility	Do the process activities culminate in clear results, so that the progress of the process is externally visible?
Measurability	Does the process include data collection or other activities that allow process or product characteristics to be measured?
Supportability	To what extent can software tools be used to support the process activities?
Acceptability	Is the defined process acceptable to and usable by the engineers responsible for producing the software product?
Reliability	Is the process designed in such a way that process errors are avoided or trapped before they result in product errors?
Robustness	Can the process continue in spite of unexpected problems?
Maintainability	Can the process evolve to reflect changing organizational requirements or identified process improvements?
Rapidity	How fast can the process of delivering a system from a given specification be completed?

Figure: Process attributes

The process of process improvement is a cyclical process, as shown in Figure. It involves three subprocesses:

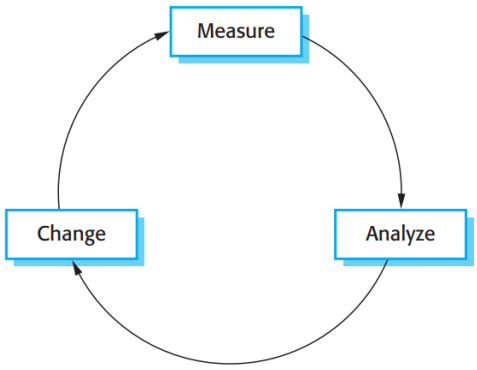


Figure 26.3 The process improvement cycle

1. Process measurement - Attributes of the current project or the product are measured. The aim is to improve the measures according to the goals of the organization involved in process improvement. This forms a baseline that helps you decide if process improvements have been effective.

2. Process analysis - The current process is assessed, and process weaknesses and bottlenecks are identified. Process models (sometimes called process maps) that describe the process may be developed during this stage. The analysis may be focused by considering process characteristics such as rapidity and robustness.

3. Process change - Process changes are proposed to address some of the identified process weaknesses. These are introduced and the cycle resumes to collect data about the effectiveness of the changes.

Software Product Quality Factors

For software products, or any other intellectual products such as books or films where the quality of the product depends on its design, there are four important factors that affect product quality. These are shown in Figure.

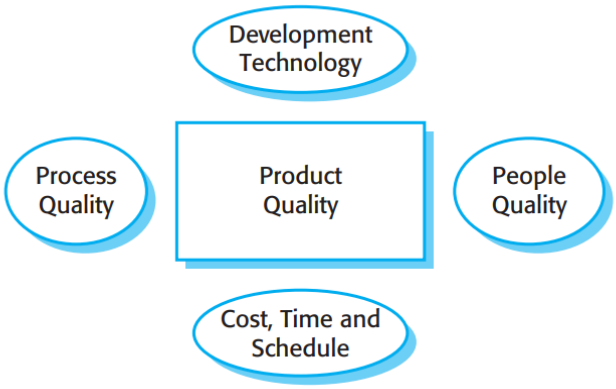


Figure 26.1 Factors affecting software product

The influence of each of these factors depends on the size and type of the project.

For very large systems that include separate subsystems, developed by teams who may be working in different locations, the principal factor that affects product quality is the software process. The major problems with large projects are integration, project management, and communications. There is usually a mix of abilities and experience in the team members and, because the development process usually takes place over a number of years, the development team is volatile. It may change completely over the lifetime of the project.

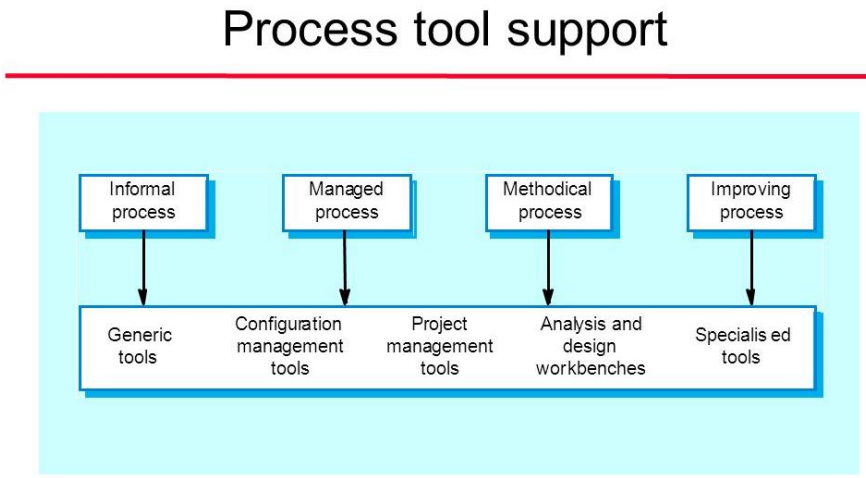
For small projects, however, where there are only a few team members, the quality of the development team is more important than the development process used. Hence, the agile manifesto proclaims the importance of people rather than process. If the team has a high level of ability and experience, the quality of the product is likely to be high, irrespective of the process used. If the team is inexperienced and unskilled, a good process may limit the damage but will not, in itself, lead to high-quality software.

Where teams are small, good development technology is particularly important. The small team cannot devote a lot of time to tedious administrative procedures. The team members spend most of their time designing and programming the system, so good tools significantly affect their productivity. For large projects, a basic level of development technology is essential for information management.

Irrespective of people, process, or tool factors, if a project has an inadequate budget or is planned with an unrealistic delivery schedule, product quality will be affected. A good process requires resources for its effective implementation. If these resources are insufficient, the process cannot be really effective. If resources are inadequate, only excellent people can save a project. Even then, if the deficit is too great, the product quality will be degraded.

Process Classification

Software processes are of different types depending on the degree of formality of the process, the types of products developed, the size of the organization and so on. There are four classes of software processes as shown in figure.



1. Informal Process – When there is no strictly defined process model the development team chooses the process that they will use. Informal processes may use formal procedures such as configuration management but the procedures and the relationships between procedure are defined as required by the development team.

2. Managed Process – A defined process model is used to define the development process. The process model defines the procedures, their scheduling and the relationship between the procedures.

3. Methodical Process – When some defined development method or methods are used these processes benefit from CASE tool support for design and analysis process

4. Improving Process – They have inherent improvement objectives and have a specific budget for improvements and procedures for introducing such improvements . As part of this quantitative process measurement may be introduced.

These classifications obviously overlap and a process may fall into several classes. These classifications are useful because they serve as a basis for multidimensional process improvement.

Process used should depend on type of product which is being developed. For large systems management is usually the principal problem so you need a strictly managed process. For smaller systems more informality is possible. There is no uniformly applicable process which should be standardized within an organization. High costs may be incurred if you force an inappropriate process on a development team.

Process Measurement

Process measurements are quantitative data about the software process, such as the time taken to perform some process activity. For example, you may measure the time required to develop program test cases.

Process measurements can be used to assess whether or not the efficiency of a process has been improved. For example, the effort and time devoted to testing can be monitored. Effective improvements to the testing process should reduce the effort and/or testing time. However, process measurements on their own cannot be used to determine if product quality has improved. Product quality data must also be collected and related to the process activities.

Three types of process metrics can be collected:

1. The time taken for a particular process to be completed - This can be the total time devoted to the process, calendar time, the time spent on the process by particular engineers, and so on.

2. The resources required for a particular process - Resources might include total effort in person-days, travel costs, or computer resources.

3. The number of occurrences of a particular event - Examples of events that might be monitored include the number of defects discovered during code inspection, the number of requirements changes requested, and the average number of lines of code modified in response to requirements change.

A fundamental difficulty in process measurement is knowing what information about the process should be collected to support process improvement. Basili and Rombach (1988) **proposed what they call the GQM (Goal-Question-Metric) paradigm**, which has become widely used in software and process measurement.

1. Goals - A goal is something that the organization is trying to achieve. It should not be directly concerned with process attributes but rather with how the process affects products or the organization itself. Examples of goals might be an improved level of process maturity, shorter product development time, or increased product reliability.

2. Questions - These are refinements of goals where specific areas of uncertainty related to the goals are identified. Normally, a goal will have a number of associated questions that need to be answered. Examples of questions related to the goal of shortening product development times might be "Where are the bottlenecks in our current process?", "How can the time required to finalize product requirements with customers be reduced?", and "How many of our tests are effective in discovering product defects?"

3. Metrics - These are the measurements that need to be collected to help answer the questions and to confirm whether or not process improvements have achieved the desired goal. To help answer the above questions, you might collect data on the time taken to complete each process activity (normalized by system size), the number of formal communications between clients and customers for each requirements change, and the number of defects discovered per test run.

Process analysis

Process analysis is the study of processes to help understand their key characteristics and how such processes are performed in practice by the people involved. Process analysis has a number of closely related objectives:

1. To understand the activities involved in the process and the relationships between these activities.

2. To understand the relationships between the process activities and the measurements that have been made.

3. To relate the specific process or processes that you are analyzing to comparable processes elsewhere in the organization, or to idealized processes of the same type.

During process analysis you are trying to understand what is going on in a process. You are looking for information about that process's problems and inefficiencies. You should also be interested in the extent that the process is used, the software tools used to support the process, and how the process is influenced by organizational constraints.

The most commonly used techniques of process analysis are:

1. Questionnaires and interviews - The engineers and managers working on a project are questioned about what actually goes on. The answers to a formal questionnaire are refined during personal interviews with those involved in the process.

2. Ethnographic studies - Ethnographic studies where process participants are observed as they work, may be used to understand the nature of software development as a human activity. Such analysis reveals subtleties and complexities that may not be revealed by questionnaires and interviews.

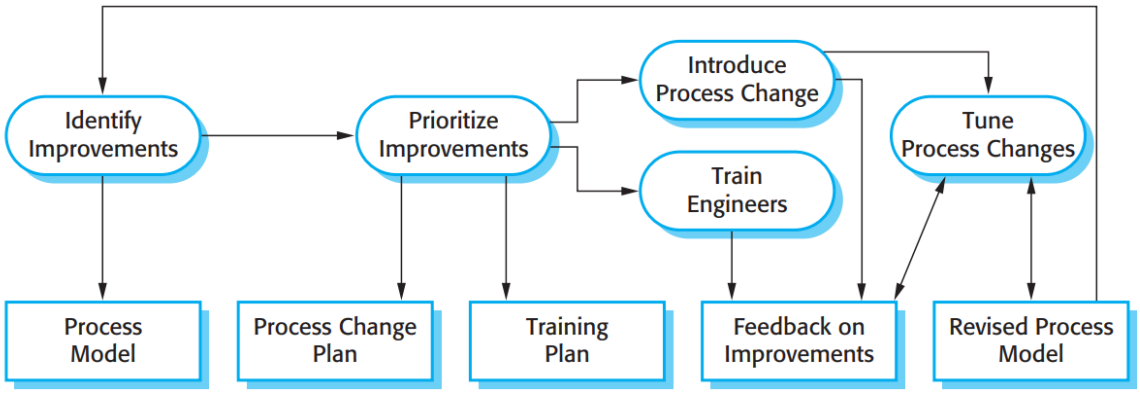
Questionnaire-based analysis can be carried out fairly quickly once the right questions have been identified. However, if the questions are badly worded or inappropriate, you may end up with an incomplete or inaccurate understanding of the process. Furthermore, questionnaire based analysis may appear to them as a form of assessment or appraisal.

Ethnographic analysis is more likely than interviews to discover the true process used. However, this type of analysis can be a prolonged activity that can last several months. It relies on external observation of the process as it is being enacted. To do a complete analysis, you have to be involved from the initial stages of a project through to product delivery and maintenance.

Process change

Process change involves making modifications to the existing process. You may do this by introducing new practices, methods, or tools; changing the ordering of process activities; introducing or removing deliverables from the process; improving communications; or by introducing new roles and responsibilities.

There are five key stages in the process change process (Figure):



1. **Improvement identification** - This stage is concerned with using the results of the process analysis to identify ways to tackle quality problems, schedule bottlenecks, or cost inefficiencies that have been identified during process analysis.

For example, a company may believe that many of its software problems stem from requirements problems. Using a requirement engineering best practice guide, various requirements engineering practices that could be introduced or changed may then be identified.

2. Improvement prioritization - This stage is concerned with assessing possible changes to the process, and prioritizing them for implementation. When many possible changes have been identified; it is usually impossible to introduce them all at once, and you must decide which are the most important. You may make these decisions based on the need to improve specific process areas, the costs of introducing a change, the impact of a change on the organization, or other factors.

For example, a company may consider the introduction of requirements management processes to manage evolving requirements to be the highest priority process change.

3. Process change - Process change introduction means putting new procedures, methods, and tools into place and integrating them with other process activities. You must allow enough time to introduce changes and ensure that these changes are compatible with other process activities and organizational procedures and standards. This may involve acquiring tools for requirements management and designing processes to use these tools.

4. Process training - Without training, it is not possible to gain the full benefits of process changes. The engineers involved need to understand the changes that have been proposed and how to perform the new and changed processes. All too often, process changes are imposed without adequate training and the effect of these changes is to degrade rather than improve product quality. In the case of requirements management, the training might involve a discussion of the value of requirements management, an explanation of the process activities, and an introduction to the tools that have been selected.

5. Change tuning - Proposed process changes will never be completely effective as soon as they are introduced. You need a tuning phase where minor problems can be discovered, and modifications to the process can be proposed and introduced. This tuning phase should last for several months until the development engineers are happy with the new process.

Once a change has been introduced, the improvement process can iterate, with further analysis used to identify process problems, propose improvements, and so on.

CMMI Process Improvement Framework

The CMMI model (Ahern et al., 2001; Chrissis et al., 2007) is intended to be a framework for process improvement that has broad applicability across a range of companies. Its staged version is compatible with the Software CMM and allows an organization's system development and management processes to be assessed and assigned a maturity level from 1 to 5.

CMMI assessment involves examining the processes in an organization and rating these processes or process areas on a six-point scale that relates to the level of maturity in each process area. The idea is that the more mature a process, the better it is. The six-point scale assigns a level of maturity to a process area as follows:

1. Incomplete - At least one of the specific goals associated with the process area is not satisfied. There are no generic goals at this level as institutionalization of an incomplete process does not make sense.

2. Performed - The goals associated with the process area are satisfied, and for all processes the scope of the work to be performed is explicitly set out and communicated to the team members.

3. Managed - At this level, the goals associated with the process area are met and organizational policies are in place that define when each process should be used. There must be documented project plans that define the project goals.

Resource management and process monitoring procedures must be in place across the institution.

4. Defined - This level focuses on organizational standardization and deployment of processes. Each project has a managed process that is adapted to the project requirements from a defined set of organizational processes. Process assets and process measurements must be collected and used for future process improvements.

5. Quantitatively managed - At this level, there is an organizational responsibility to use statistical and other quantitative methods to control subprocesses; that is, collected process and product measurements must be used in process management.

6. Optimizing - At this highest level, the organization must use the process and product measurements to drive process improvement. Trends must be analyzed and the processes adapted to changing business needs.