UNIT-1

A Software Maturity Framework:

Fundamentally, software development must be predictable.

The software process is the set of tools, methods, and practices we use to produce a software product. The objectives of software process management are to produce products according to plan while simultaneously improving the organization's capability to produce better products.

The basic principles are those of statistical process control. A process is said to be stable or under statistical control if its future performance is predictable within established statistical limits. When a process is under statistical control, repeating the work in roughly the same way will produce roughly the same result. To obtain consistently better results, it is necessary to improve the process. If the process is not under statistical control, sustained progress is not possible until it is.

Lord Kelvin - "When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced the stage of science." (But, your numbers must be reasonably meaningful.)

The mere act of measuring human processes changes them because of people's fears, and so forth.

Measurements are both expensive and disruptive; overzeal-ous measurements can disrupt the process under study.

SOFTWARE PROCESS IMPROVEMENT

To improve their software capabilities, organizations must take six steps;

•Understand the current status of the development pro-cesses

•Develop a vision of the required process

•Establish a list of required process improvement actions in order of priority

•Produce a plan to accomplish the required action •Commit the resources to execute the plan

•Start over at the first step.

PROCESS MATURITY LEVELS

1. Initial. Until the process is under statistical control, orderly progress in process improvement is not possible. Must at least achieve rudimentary predictability of schedules and costs.

2. **Repeatable**. The organization has achieved a stable pro-cess with a repeatable level of statistical control by initiating rigorous project management of commitments, costs, sched-ules, and changes

3. **Defined.** The organization has defined the process as a basis for consistent implementation and better understand-ing. At this point, advanced technology can be introduced.

4. **Managed**. The organization has initiated comprehensive process measurements and analysis. This is when the most significant quality improvements begin.

5. **Optimizing**. The organization now has a foundation for continuing improvement and optimization of the process.

These levels are selected because they

•Represent the historical phases of evolutionary improve-ment of real software organizations

•Represent a measure of improvement that is reasonable to achieve from a prior level

•Suggest improvement goals and progress measures •Make obvious a set of intermediate improvement priorities once an organization's status in this framework is known.

The Initial Process (Level 1)

Usually ad hoc and chaotic - Organization operates without formalized procedures, cost estimates, and project plans. Tools are neither well intergrated with the process nor uni-formly applied. Change control is lax, and there is little senior management exposure or understanding of the problems and issues. Since many problems are deferred or even forgotten, software installation and maintenance often present serious problems.

While organizations at this level may have formal procedures for planning and tracking work, there is no management mechanism to insure they are used. Procedures are often abandoned in a crisis in favor of coding and testing. Level 1 organizations don't use design and code inspections and other techniques not directly related to shipping a product.

Organizations at Level 1 can improve their performance by instituting basic project controls. The most important ones are

•Project management •Management oversight •Quality assurance •Change control

The Repeatable Process (Level 2)

This level provides control over the way the organization establishes plans and commitments. This control provides such an improvement over Level 1 that the people in the organization tend to

believe they have mastered the software problem. This strength, however, stems from their prior expe-rience in doing similar work. Level 2 organizations face major risks when presented with new challenges.

Some major risks:

- New tools and methods will affect processes, thus destroying the historical base on which the organization lies. Even with a defined process framework, a new tech-nology can do more harm than good.
- When the organization must develop a new kind of prod-uct, it is entering new territory.
- Major organizational change can be highly disruptive. At Level 2, a new manager has no orderly basis for under-standing an organization's operation, and new members must learn the ropes by word of mouth.

Key actions required to advance from Repeatable to the next stage, the Defined Process, are:

- Establish a process group: A process group is a technical resource that focuses heavily on improving software pro-cesses. In most software organizations, all the people are generally devoted to product work. Until some people are assigned full-time to work on the process, little orderly progress can be made in improving it.
- Establish a software development process architecture (or development cycle) that describes the technical and management activities required for proper execution of the development process. The architecture is a structural decomposition of the development cycle into tasks, each of which has a defined set of prerequisites, functional decompositions, verification procedures, and task comple-tion specifications.

The Defined Process (Level 3)

The organization has the foundation for major and continuing change. When faced with a crisis, the software teams will continue to use the same process that has been defined.

However, the process is still only qualitative; there is little data to indicate how much is accomplished or how effective the process is. There is considerable debate about the value of software process measurements and the best one to use.

The key steps required to advance from the Defined Process to the next level are:

- Establish a minimum set of basic process measurements to identify the quality and cost parameters of each process step. The objective is to quantify the relative costs and bene-fits of each major process activity, such as the cost and yield of error detection and correction methods.

•Establish a process database and the resources to man-age and maintain it. Cost and yield data should be main-tained centrally to guard against loss, to make it available for all projects, and to facilitate process quality and productivity analysis.

•Provide sufficient process resources to gather and main-tain the process data and to advise project members on its use. Assign skilled professionals to monitor the quality of the data before entry into the database and to provide guidance on the analysis methods and interpretation.

•Assess the relative quality of each product and inform management where quality targets are not being met. Should be done by an independent quality assurance group

The Managed Process (Level 4)

Largest problem at Level 4 is the cost of gathering data. There are many sources of potentially valuable measure of the software process, but such data are expensive to collect and maintain.

Productivity data are meaningless unless explicitly defined. For example, the simple measure of lines of source code per expended development month can vary by 100 times or more, depending on the interpretation of the parameters.

When different groups gather data but do not use identical definitions, the results are not comparable, even if it makes sense to compare them. It is rare when two processes are comparable by simple measures. The variations in task com-plexity caused by different product types can exceed five to one. Similarly, the cost per line of code for small modifications is often two to three times that for new programs.

Process data must not be used to compare projects or indi-viduals. Its purpose is too illuminate the product being devel-oped and to provide an informed basis for improving the process. When such data are used by management to evalu-ate individuals or terms, the reliability of the data itself will deteriorate.

The two fundamental requirements for advancing from the Managed Process to the next level are:

- •Support automatic gathering of process data. All data is subject to error and omission, some data cannot be gath-ered by hand, and the accuracy of manually gathered data is often poor.
- •Use process data to analyze and to modify the process to prevent problems and improve efficiency.

The Optimizing Process (Level 5)

To this point software development managers have largely focused on their products and will typically gather and ana-lyze only data that directly relates to product improvement. In the Optimizing Process, the data are available to tune the process itself.

For example, many types of errors can be identified far more economically by design or code inspections than by testing. However, some kinds of errors are either uneconomical to detect or

almost impossible to find except by machine. Exam-ples are errors involving interfaces, performance, human fac-tors, and error recovery.

So, there are two aspects of testing: removal of defects and assessment of program quality. To reduce the cost of remov-ing defects, inspections should be emphasized. The role of functional and system testing should then be changed to one of gathering quality data on the program. This involves study-ing each bug to see if it is an isolated problem or if it indicates design problems that require more comprehensive analysis.

With Level 5, the organization should identify the weakest elements of the process and fix them. Data are available to justify the application of technology to various critical tasks, and numerical evidence is available on the effectiveness with which the process has been applied to any given product.

The Principles of Software Process Change

People:

•The best people are always in short supply

•You probably have about the best team you can get right now.

•With proper leadership and support, most people can do much better than they are currently doing

Design:

•Superior products have superior design. Successful products are designed by people who understand the application (domain engineer).

•A program should be viewed as executable knowledge. Program designers should have application knowledge.

The Six Basic Principles of Software Process Change: •Major changes to the process must start at the top. •Ultimately, everyone must be involved.

•Effective change requires great knowledge of the current process

•Change is continuous

•Software process changes will not be retained without conscious effort and periodic reinforcement

•Software process improvement requires investment

Continuous Change:

•Reactive changes generally make things worse •Every defect is an improvement opportunity

•Crisis prevention is more important than crisis recovery

Some Common Misconceptions about the Software Pro-cess

•We must start with firm requirements •If it passes test it must be OK •Software quality can't be measured •The problems are technical

•We need better people

•Software management is different

FIRM REQUIEMENTS - A software perversity law seems to be the more firm the specifications, the more likely they are to be wrong. With rare exceptions, requirements change as the software job progresses. Just by writing a program, we change our perceptions of the task. Requirements cannot be firm because we cannot anticipate the ways the tasks will change when they are automated.

For large-scale programs, the task of stating a complete requirement is not just difficult; it is impossible. Generally, we must develop software incrementally.

However, we must have stability long enough to build and test a system. However, if we freeze requirements too early, later retrofits are expensive.

SOFTWARE MANAGEMENT IS DIFFERENT - Management must not view it as black art. Must insist on tracking plans and reviews.

Champions, Sponsors, and Agents

- Champions - Ones who initiate change. They bring man-agement's attention to the subject, obtain the blessing of a sponsor, and establish the credibility to get the change program launched. The champion maintains focus on the goal, strives to overcome obstacles, and refuses to give up when the going gets tough.
- Sponsors - Senior manager who provides resources and official backing. Once a sponsor is found, the champion's job is done; it is time to launch the change process.
- Agents - Change agents lead the planning and implemen-tation. Agents must be enthusiastic, technically and politi-cally savvy, respected by others, and have management's confidence and support.

- Agents - Change agents lead the planning and implemen-tation. Agents must be enthusiastic, technically and politi-cally savvy, respected by others, and have management's confidence and support.

Elements of Change

•Planning

•Implementation

•Communication

Software Process Assessment

Process assessments help software organizations improve themselves by identifying their crucial problems and estab-lishing improvement priorities. The basic assessment objec-tives are:

•Learn how the organization works •Identify its major problems

•Enroll its opinion leaders in the change process

The essential approach is to conduct a series of structured interviews with key people in the organization to learn their problems, concerns, and creative ideas.

ASSESSMENT OVERVIEW

A software assessment is not an audit. Audit are conducted for senior managers who suspect problems and send in experts to uncover them. A software process assessment is a review of a software organization to advise its management and professionals on how they can improve their operation.

The phases of assessment are:

•Preparation - Senior management agrees to participate in the process and to take actions on the resulting recom-mendations or explain why not. Concludes with a training program for the assessment team

•Assessment - The on-site assessment period. It takes several days to two or more weeks. It concludes with a preliminary report to local management.

•Recommendations - Final recommendations are pre-sented to local managers. A local action team is then formed to plan and implement the recommendations.

Five Assessment Principles:

•The need for a process model as a basis for assessment

•The requirement for confidentiality

•Senior management involvement

•An attitude of respect for the views of the people in the organization be assessed

•An action orientatio

Process models:

THE CAPABILITY MATURITY MODEL INTEGRATION (CMMI):

The CMMI represents a process meta-model in two different ways:

- As a continuous model
- As a staged model. Each process area is formally assessed against specific goals and practices and is rated according to the following capability levels.

Level 0: Incomplete. The process area is either not performed or does not achieve all goals and objectives defined by CMMI for level 1 capability.

Level 1: Performed. All of the specific goals of the process area have been satisfied. Work tasks required to produce defined work products are being conducted.

Level 2: Managed. All level 1 criteria have been satisfied. In addition, all work associated with the process area conforms to an organizationally defined policy; all people doing the work have access to adequate resources to get the job done; stakeholders are actively involved in the process area as required; all work tasks and work products are "monitored, controlled, and reviewed;

Level 3: Defined. All level 2 criteria have been achieved. In addition, the process is "tailored from the organizations set of standard processes according to the organizations tailoring guidelines, and contributes and work products, measures and other process-improvement information to the organizational process assets".

Level 4: Quantitatively managed. All level 3 criteria have been achieved. In addition, the process area is controlled and improved using measurement and quantitative assessment."Quantitative objectives for quality and process performance are established and used as criteria in managing the process"

Level 5: Optimized. All level 4 criteria have been achieved. In addition, the process area is adapted and optimized using quantitative means to meet changing customer needs and to continually improve the efficacy of the process area under consideration"

The CMMI defines each process area in terms of "specific goals" and the "specific practices" required to achieve these goals. Specific practices refine a goal into a set of process-related activities. The specific goals (SG) and the associated specific practices(SP) defined for project planning are

SG 1 Establish estimates

SP 1.1 Estimate the scope of the project

SP 1.2 Establish estimates of work product and task attributes

SP 1.3 Define project life cycle

SP 1.4 Determine estimates of effort and cost

SG 2 Develop a Project Plan

SP 2.1 Establish the budget and schedule

SP 2.2 Identify project risks

SP 2.3 Plan for data management

SP 2.4 Plan for needed knowledge and skills

SP 2.5 Plan stakeholder involvement

SP 2.6 Establish the project plan

SG 3 Obtain commitment to the plan

SP 3.1 Review plans that affect the project

SP 3.2 Reconcile work and resource levels

SP 3.3 Obtain plan commitment

In addition to specific goals and practices, the CMMI also defines a set of five generic goals and related practices for each process area. Each of the five generic goals corresponds to one of the five capability levels. Hence to achieve a particular capability level, the generic goal for that level

and the generic practices that correspond to that goal must be achieved. To illustrate, the generic goals (GG) and practices (GP) for the project planning process area are

GG 1 Achieve specific goals

GP 1.1 Perform base practices

GG 2 Institutionalize a managed process

GP 2.1 Establish and organizational policy

GP 2.2 Plan the process GP 2.3 Provide resources

GP 2.4 Assign responsibility

GP 2.5 Train people

GP 2.6 Manage configurations

GP 2.7 Identify and involve relevant stakeholders

GP 2.8 Monitor and control the process

GP 2.9 Objectively evaluate adherence

GP 2.10 Review status with higher level management

GG 3 Institutionalize a defined process

GP 3.1 Establish a defined process

GP 3.2 Collect improvement information

GG 4 Institutionalize a quantitatively managed process

GP 4.1 Establish quantitative objectives for the process

GP 4.2 Stabilize sub process performance

PERSONAL AND TEAM PROCESS MODELS:

The best software process is one that is close to the people who will be doing the work.Each software engineer would create a process that best fits his or her needs, and at the same time meets the broader needs of the team and the organization. Alternatively, the team itself would create its own process, and at the same time meet the narrower needs of individuals and the broader needs of the organization.

Personal software process (PSP) The personal software process (PSP) emphasizes personal measurement of both the work product that is produced and the resultant quality of the work product.

The PSP process model defines five framework activities:

planning,

high-level design,

high level design review,

development, and postmortem.

Planning: This activity isolates requirements and, base on these develops both size and resource estimates. In addition, a defect estimate is made. All metrics are recorded on worksheets or templates. Finally, development tasks are identified and a project schedule is created

. High level design: External specifications for each component to be constructed are developed and a component design is created. Prototypes are built when uncertainty exists. All issues are recorded and tracked.

High level design review: Formal verification methods are applied to uncover errors in the design. Metrics are maintained for all important tasks and work results.

Development: The component level design is refined and reviewed. Code is generated, reviewed, compiled, and tested. Metrics are maintained for all important task and work results.

Postmortem: Using the measures and metrics collected the effectiveness of the process is determined. Measures and metrics should provide guidance for modifying the process to improve its effectiveness.

PSP stresses the need for each software engineer to identify errors early and, as important, to understand the types of errors that he is likely to make. PSP represents a disciplined, metrics-based approach to software engineering.

Team software process (TSP): The goal of TSP is to build a "self-directed project team that organizes itself to produce high-quality software.

The following are the objectives for TSP:

- Build self-directed teams that plan and track their work, establish goals, and own their processes and plans. These can be pure software teams or integrated product teams(IPT) of 3 to about 20 engineers.
- Show managers how to coach and motivate their teams and how to help them sustain peak performance.
- Accelerate software process improvement by making CMM level 5 behavior normal and expected.
- Provide improvement guidance to high-maturity organizations.
- Facilitate university teaching of industrial-grade team skills.

A self-directed team defines

- roles and responsibilities for each team member

- tracks quantitative project data

- identifies a team process that is appropriate for the project

- a strategy for implementing the process

- defines local standards that are applicable to the teams software engineering work;

- continually assesses risk and reacts to it

- Tracks, manages, and reports project status.

- TSP defines the following framework activities:

launch, high-level design,

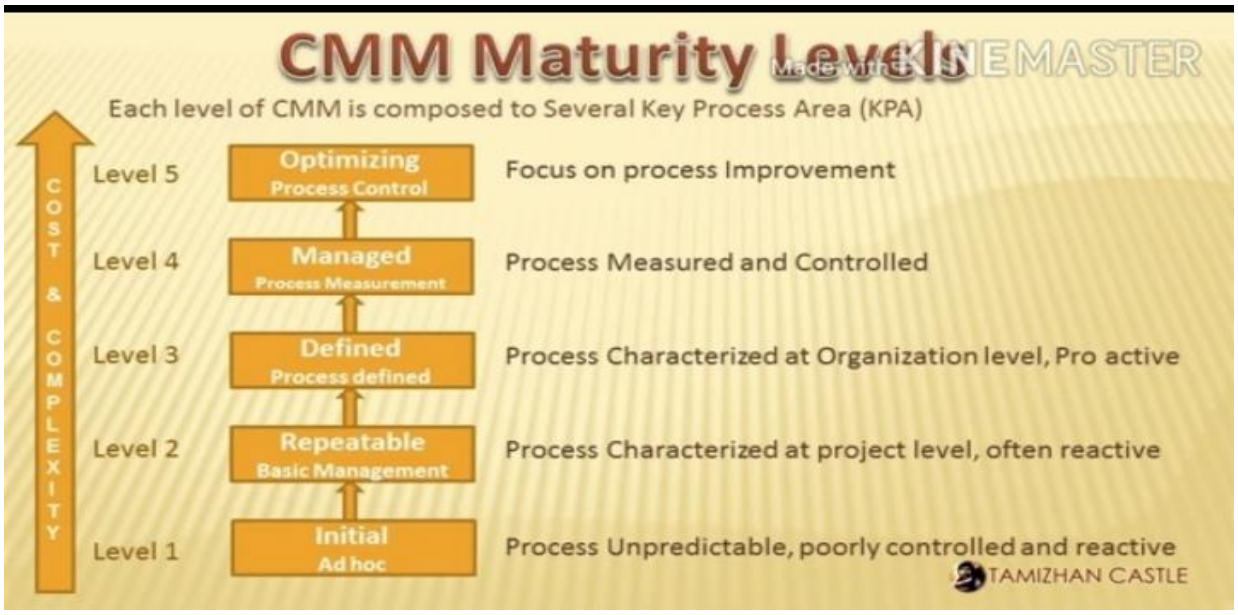
mplementation, integration and test, and postmortem.

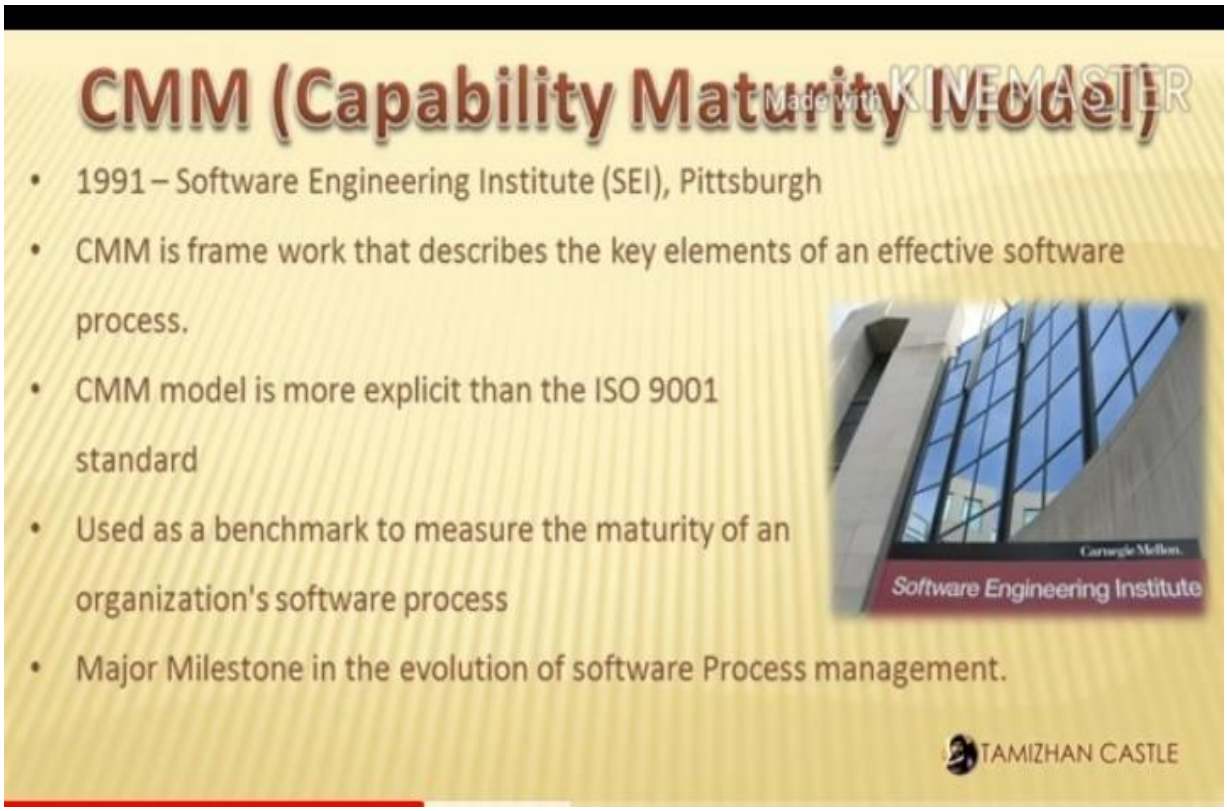
TSP makes use of a wide variety of scripts, forms, and standards that serve to guide team members in their work.

Scripts define specific process activities and other more detailed work functions that are part of the team process. Each project is "launched" using a sequence of tasks. The following launch script is recommended

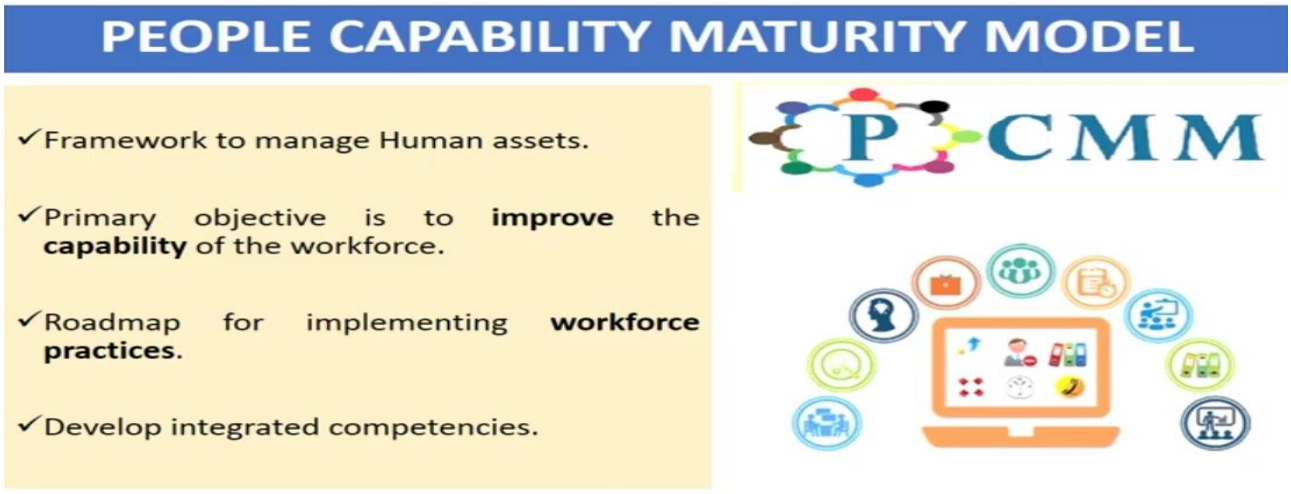
- Review project objectives with management and agree on and document team goals
- Establish team roles
- Define the teams development process
- Make a quality plan and set quality targets
- Plan for the needed support facilitie

CMM




CMM (Capability Maturity Model)
Made with KINEMASTER
• 1991 – Software Engineering Institute (SEI), Pittsburgh
• CMM is frame work that describes the key elements of an effective software process.
• CMM model is more explicit than the ISO 9001 standard
• Used as a benchmark to measure the maturity of an organization's software process
• Major Milestone in the evolution of software Process management.
Carnegie Mellon
Software Engineering Institute
TAMIZHAN CASTLE

PCMM

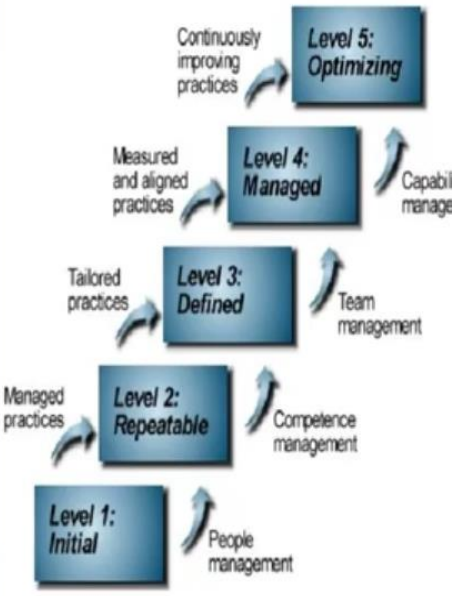

PEOPLE CAPABILITY MATURITY MODEL
✓Framework to manage Human assets.
✓Primary objective is to improve the capability of the workforce.
✓Roadmap for implementing workforce practices.
✓Develop integrated competencies.
P CMM

PCMM OBJECTIVES

- To improving capability of organisations by increasing capability of workforce;
- An attribute of the organisation rather than of A few individuals;
- To align the motivation of individuals with that of the organisation;
- To retain valuable human assets.

PCMM LEVELS

- ❑**INITIAL** - No consistent way of performing workforce practices.
- ❑**REPEATABLE** - Establish a foundation to deploy common workforce practices across the organization.
- ❑**DEFINED** - Identifies and develops workforce competencies and align them with business strategies and objectives.
- ❑**MANAGED** - Empowers and integrates workforce competencies and manages performance quantitatively.
- ❑**OPTIMIZING** - Continuously improves and aligns personal, work-group, and organizational capability.



PRINCIPLES

- Workforce capability is directly related to business performance.
- Competitive issue and a source of strategic advantage.
- Defined in relation to the organization's strategic business objectives.
- Workforce competencies.
- Measured and improved at multiple levels.
- Process composed from proven practices and procedures.
- Provide improvement opportunities.
- Continually evolve workforce practices and develop new workforce competencies.