UNIT - I

A software maturity frame work:

* The software process is the set of tools, methods and pratices we use to produce a software product.

* The objectives of s/w process mgmt are to produce products according to plan while simultaneously improving the organizations capability to produce better products.

* A process is said to be stable (or) under statistical control if its future performance is predictable within established statistical limits.

* The basic principle behind statistical control is measurement. As Lord Kelvin said a century ago. "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".

Software Process Improvement

* An important first step in addressing s/w problems is to treat the entire s/w task as a process that can be controlled, measured & improved.

* A fully effective s/w process must consider the relationships of all the required tasks, the tools and methods used, and the skill, training and motivation of the people involved.

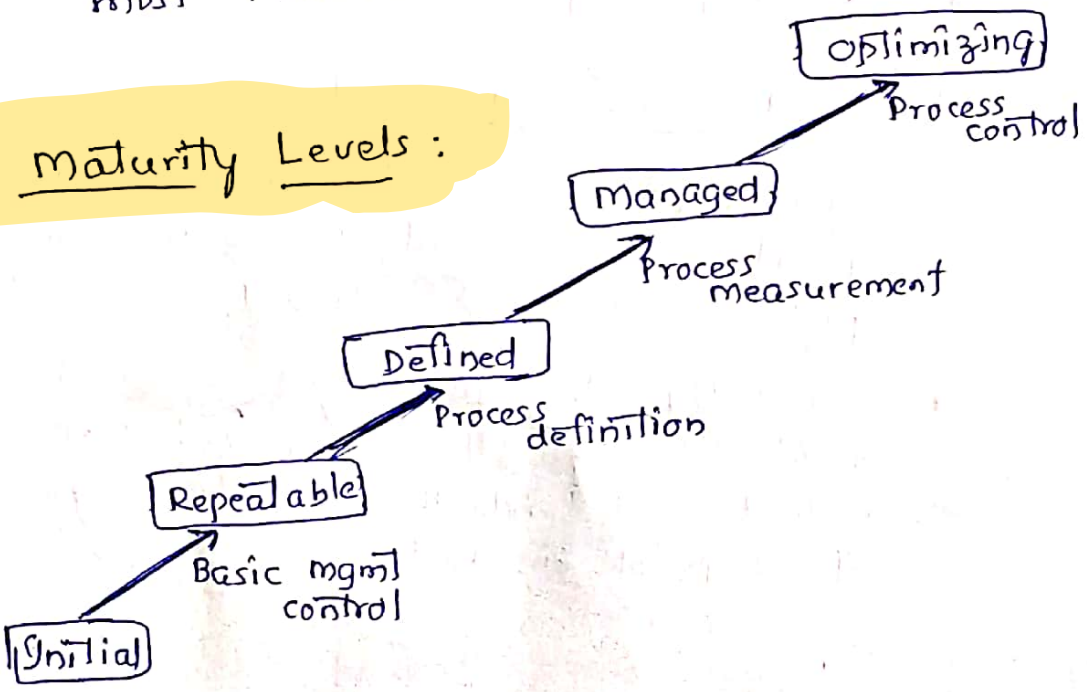
To improve their s/w capabilities, organizations must take six steps:

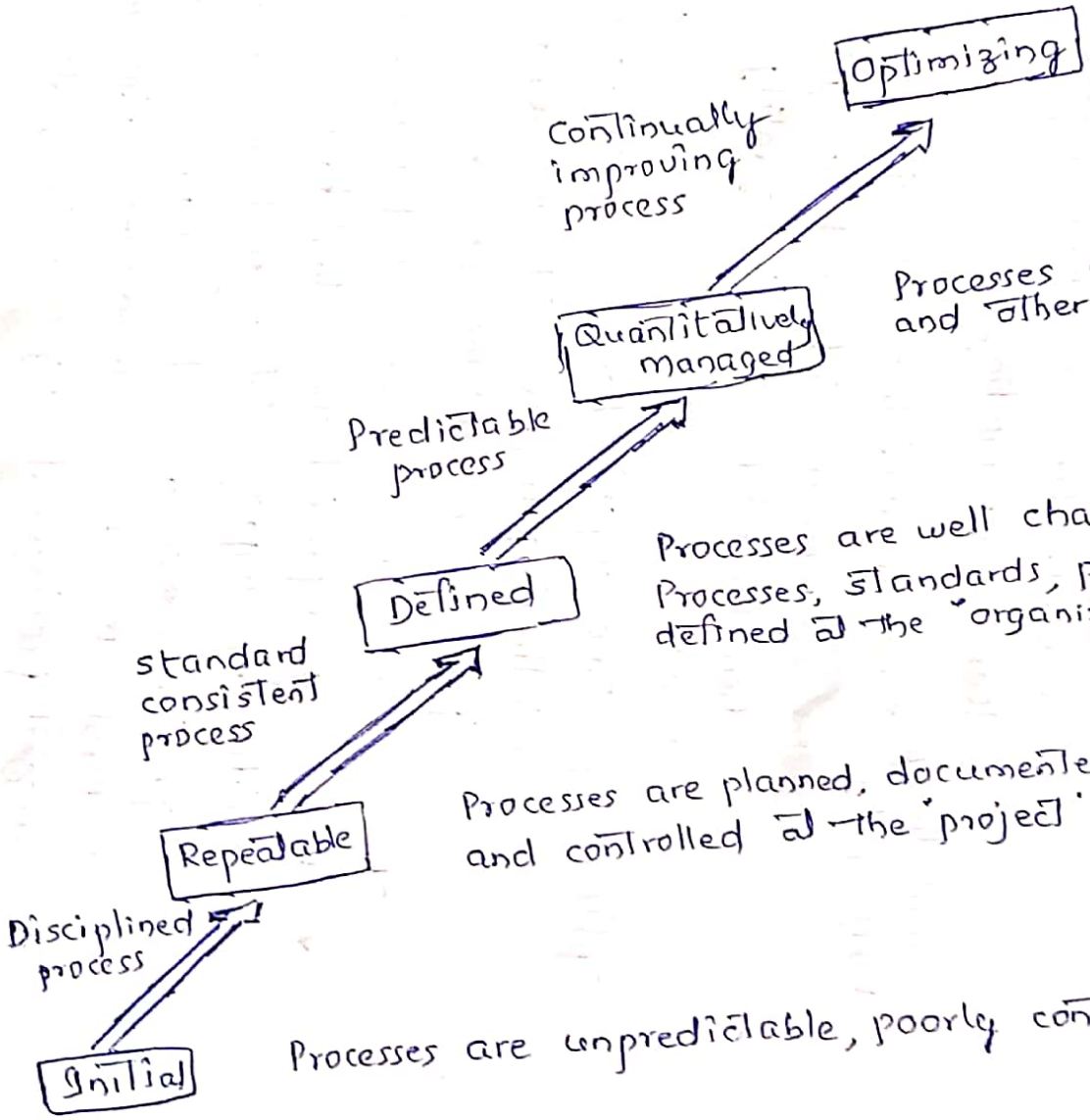
1. Understand the current status of their development process (or processes.
2. Develop a vision of the desired process.
3. Establish a list of required process improvement actions in order of priority
4. Produce a plan to accomplish the required actions.
5. Commit the resources to execute the plan.
6. Start over at step 1.

* To improve an organization, it is helpful to have a clear picture of the ultimate goal and some way to guage progress along the way

* By establishing their organization's position in maturity structure, s/w professionals and their managers can more readily identify areas where improvement actions will be most fruitful.

Process Maturity Levels:





Process performance continually improved through incremental and innovative technological improvements

Processes are controlled using statistical and other quantitative techniques

Processes are well characterized and understood. Processes, standards, procedures, tools etc are defined at the "organizational" level. Proactive

Processes are planned, documented, performed, monitored and controlled at the "project" level. often reactive

Processes are unpredictable, poorly controlled, reactive

Initial: Until the process is under statistical control, orderly progress in process improvement is not possible.

Repeatable: The organization has achived a stable process with a repeatable level of statistical control by initiating rigorous project mgmt of commitments, costs, schedules and changes.

Defined: The organization has defined the process as a basis for consistent implementation and better understanding.

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

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

These levels have been selected because they:

* Resonably represent the actual historical phases of evolutionary improvement of real sfw organization.
* Represent a measurement of improvement that is resonable to achieve from the prior level.
* Suggest provisional improvement goals and progress measures.

The Principles of s/w process change:

More generally, the software process change rests on many of the principles of organizational change. However, to have general familiarity with such issues as resistance to change, champions, sponsors, agents, unfreezing and refreezing before launching a s/w process change program.

Processe in Perspective:

* When some approach seems to fit a need, it is often tempting to assume it will solve all the problems.
* While process management provides a powerful basis for assessing s/w problems and a consistent framework for organizational improvement.
* It is not a cure-all.
* Two other key areas that need to be considered are
  - People and
  - Design methods.

People:

* The best people are always in short supply.
* You probably have about the best team you can go right now.
* With proper leadership and support, most people can do much better than they are currently doing.

Design methods:

* 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 s/w process change:

The basic principles of s/w change are:

- major changes to software process must start at the top.
- Ultimately, everyone must be involved
- Effective change requires a goal and knowledge of the current process
- change is continuous
- software process change will not be retained without conscious effort and periodic reinforcement.
- software process improvement requires investment.

Major changes to the s/w process must start at the top:

Senior management leadership is required to launch the change effort and to provide continuing resources and priority.

Ultimately, everyone must be involved:

Software engineering is a team effort and anyone who does not participate in improvement will miss the benefits and may even inhibit progress.

Effective change requires a goal and knowledge of the current process:

To use a map, you must know where you are.

Change is continuous:

* Reactive change generally make things worse.
* Every defect is an improvement opportunity
* Crisis prevention is more important than crisis recovery

Software process change will not be retained without conscious effort and periodic reinforcement:

The tendency for improvements to deteriorate is characterized by the term "entropy" (webster's: a measure of degree of disorder in a system)

New methods must be carefully introduced and periodically monitored or they to will rapidly decay.

Human adoption of new process involves four stages:

* Installation: Initial training
* Practice: People learn to perform as instructed
* Proficiency: Traditional learning curve
* Naturalness: method ingrained and performed without intellectual effort.

Software process improvement requires investment:
(It takes, time, skill and money)

* To improve the s/w process, some one must work on it.
* Unplanned process improvement is wishful thinking
* Automation of a poorly defined process will produce poorly defined results.
* Improvements should be made in small tested steps
* Train, train, train!

Some common misconceptions about the s/w process:

Effective change depends on realism. Without an appreciation of the current state of the s/w process and a realistic view of the future, it is easy to succumb to wishful thinking.

These then block the rational thought needed to address problems in an orderly way

Some of the more common software process misconceptions are:

- 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 requirement:

* A s/w perversity law seems to be the more firm the specifications are, the more likely they are to be wrong
* With rare exceptions, the requirements must change as the software job progresses.
* Just by writing a program, we change our perceptions of the task.
* Requirements by their very nature 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 practically impossible
* Generally, we develop s/w incrementally

If it passes test, it must be OK:

* Establish as high quality a s/w process as you can.
* Determine the most complex, essentially error-free programs you can build with this process.
* When highly reliable programs are needed, limit their complexity to what you know your can produce with your essentially defect-free process.
* Test your programs as thoroughly as you economically can.
* Recognize that your capability to build progressively larger defect-free programs will improve as you manage.

The problems are technical:

In spite of the many improved languages, tools and environments, the problems of s/w cost, schedule and quality remain.

We need better people:

Since the software professionals make the errors, some people erroneously feel that they should be blamed for them.

Software management is different:

many factors make s/w different from other engg. fields:

* s/w is generally more complex than other engg. products.
* Since s/w engg. is relatively new, there are not yet many managers (or) senior professionals with sufficient eaperience to appreciate the benefits of an effective process.
* The insignificant cost of reproducing s/w forces s/w solutions to many late-discovered system problems.

* The s/w discipline is not grounded in natural science.
* S/w is often the system element that presents the function to the end user.
* S/w is also often the crucial ingredient that couples all the other system elements together.
* S/w is often viewed by non-software practitioners as a "black art".

A Strategy for implementing software Process Change:

- Champions
- Sponsors
- Agents
- The elements of change

Champions: are ones who initiate the change.

They bring management's attention to the subject, obtain the blessings of a sponsor, and establish the credibility to get the change program launched.

The champions maintain focus on the goal, strives to overcome obstacles and refuses to give up when the going gets rough.

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: Once it is clear that changes are needed and management sponsorship has been obtained, the essential next step is to identify the change 'agents' who will lead change planning and implementation.

The key points to consider while selecting agents:

* Agents should be enthusiastic about leading the change process.
* Agents must be both technically and politically capable of understanding the problems and ensuring that effective solutions are implemented.
* Agents need the respect of the people they are to deal with.
* Agents must have management's confidence and support

The elements of change:

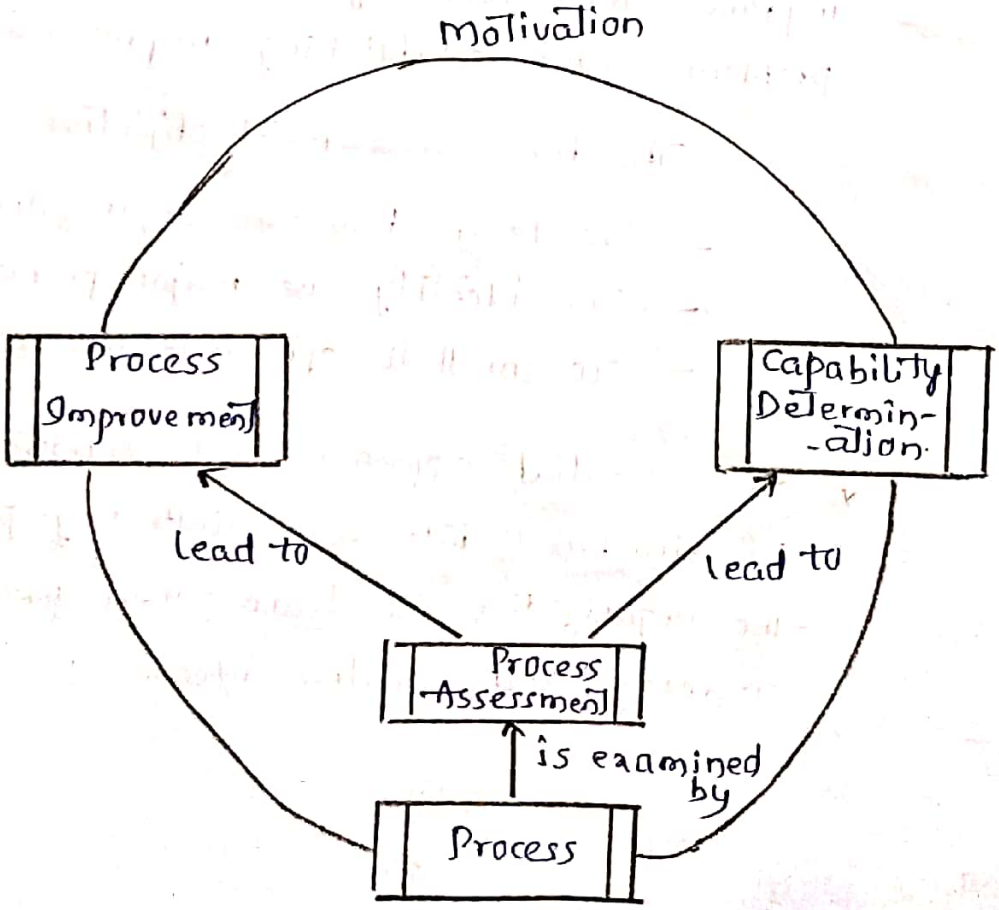
* Planning
* Implementation
* Communication

Refreezing: It can take many forms, but the basic objective is to ensure that an achieved capability is retained in general practice. Several common techniques are:

* Retain the mgmt team that instituted the change
* modify the organization's procedures.
* Establish measurements and incentives
* set up a dedicated staff to monitor and support performance
* Establish a training program

Software Process Assessment:

* Software Process Assessment examines whether the software processes are effective and efficient in accomplishing the goals.
* This is determined by the capability of selected software processes.
* The capability of a process determines whether a process with some variations is capable of meeting user's requirements.
* In addition, it measures the extent to which the software process meets the user's requirements.
* It is useful to the organization as it helps in improving the existing processes.
* In addition, it determines the strengths, weaknesses and the risks involved in the processes



* The process assessment leads to process capability determination and process improvement.
* Process capability determination is an organized assessment, which analyzes the software processes in an organization.
* In addition, it identifies the capabilities of a process and the risks involved in it.
* The process improvent identifies the changes to be made in the software processes.
* The software capability determination motivates the organization to perform s/w process improvement.

* Process assessments help software organizations improve themselves by identifying their crucial problems and establishing improvement priorities.

The basic assessment objectives are:

- To learn how the organization works
- To identify the major problems
- To 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 problem, concerns and creative ideas.

Assessment Overview:

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

Assessment Phases:

The phases of assessment are:

- Preparation
- Assessment
- Recommendations

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

Assessment: It is the on-site assessment period. This activity typically takes several days, although it can take two or more weeks, depending on the size of the organization. It concludes with a preliminary report of the finding to local management.

Recommendations: Final recommendations are presented to local managers. A local team is then assembled to plan and implement the recommendations.

Five Assessment Principles

As in many activities, the basic requirements for a good assessment are a competent team, sound leadership and a co-operative organization. Because the software process is human-intensive. However, some special considerations should be kept in mind.

They are:

- The need for a process model as a basis for the assessment
- The requirement for confidentiality
- Senior manager involvement
- An attitude of respect for the views of the people in the organization being assessed.
- An action orientation.

Start with a proces model: Without a model, there is no standard; therefore no measure of change.

Observe strict confidentiality: Otherwise, people will learn, they cannot speak in confidence. This means managers can't be in interviews with their subordinates.

Involve senior management: The senior managers (called site manager here) sets the organization priorities. The site manager must be personally involved in the assessment and its follow-up actions. Without this support, the assessment is a waste of time because lasting improvement must survive periodic crisis.

Respect the people in the assessed organization:

They probably work hard and are trying to improve. Do not appear arrogant; otherwise, they will not co-operate and may try to prove the team is ineffective. The only source of real information is from the workers.

Focus on action:

Finally, to have lasting effect, the assessment must be directed towards improvement. An action orientation keeps the questions focussed on current problems and the need to solve them.

Implementation considerations

The greatest assessment risk is that no significant improvement actions will be taken: without proper management focus and few superficial changes won't help.

A catalyst is needed to maintain the improvement priority, such as goals and management reviews.

- Risks
- Staffing

Risks:

- Schedule conflicts
- Inadequate support
- Lack of follow-through

The Initial Process (Level 1)

Usually adhoc and chaotic:

Organization operates without formalized procedures, cost estimates and project plans.

Tools are neither well integrated with the process nor uniformly applied.

Change control is lack, 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.

Organizations at this level can improve their performance by instituting basic project controls. The most important ones are:

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

In the Initial Process, professionals are driven from crisis to crisis by unplanned priorities and unmanaged change.

Such groups seldom meet commitments. They may deliver on schedule occasionally.

Why Organizations are Chaotic

Simplest explanation: People don't like to admit they don't know. When political pressure amounts for a commitment, it is difficult to say you don't know. So, people tend to promise products they can't deliver.

While lack of commitment discipline is the common reason for chaotic behaviour, there are other forces.

* Under extreme pressure, software managers make a guess instead of a plan. The guess is usually low, so chaos develops
* When things get rough, there may be a strong temptation to believe in magic. A saviour (or) new technology may be the answer.
* Programs take more code than expected.
* As the program become larger, new technical and management issues araise.
* Since these are unlike previous experience, they are a surprise.

Unplanned commitments:

When a new feature seems simple, management may just commit to do it without planning. Often, the change is far more complicated than thought. This results in confusion, changed priorities and stress. The users and senior management become frustrated and unwilling to listen to excuses when this becomes a pattern.

Gurus:

Gurus can make this worse because they run projects out of their heads. With nothing written down, everyone comes to them for guidance.

Magic:

- No silver bullets" — Brooks

Problems of scale:

Software size is insidious because of its impact on the development process.

As software products become larger, the progressive level of scale are roughly as follows:

* One person knows the details of the program.
* One person understands it but can't remember it so the design must be documented.
* One person understands the overall design of the program, but the details of its component modules are understood by separate experts.
* A large software project is defined and understood at the product management level, but a separate team understands the characteristics and design of each of its component programs.
* When the system is very large and has evolved through many versions, there may be no one who understands it.

Software Process Entropy :

As we build new systems, we learn what we should have built. This is a good reason for prototypes. This can help avoid chaos.

The Way Out

- Apply systematic project management :
  The work must be estimated, planned and managed.
- Adhere to careful change control :
  Changes must be controlled, including requirements, design, implementation and test.
- Utilize independent software assurance :
  An independent technical means is required to assure that all essential project activities are properly formed.

Basic Principles for controlling Chaos in software Organization are :

- Plan the work
- Track and maintain the plan.
- Divide the work into individual parts.
- Precisely define the requirements for each part.
- Rigorously control the relationships among the parts.
- Treat software development as a learning process.
- Recognize what you don't know.
- When the gap between your knowledge and the task is severe, fix it before proceeding
- Manage, audit and review the work to ensure it is done as planned.
- Commit to your work and work to your commitments.
- Refine the plan as your knowledge improves.

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 s/w problem.

Some major risks:

- New tools and methods will affect processes, thus destroying the historical base on which the organization lies.
- When the organization must develop a new kind of product, it is entering new territory.
- Major organizational change can be highly disruptive. At Level 2, a new manager has no orderly basis for understanding an organization's operation, and new members must learn the ropes by words of mouth.

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

- Establish a Process Group
- Establish s/w development process architecture
- Introduce a family of s/w engineering methods and Technologies.

Establish a Process Group: The process group is a technical resource that focuses heavily on improving s/w process. In most s/w 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 s/w development process architecture : (or) development cycle)

It 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 decomposition, verification procedures and task completion specifications.

Introduce a family of s/w engineering methods and technologies

These include design and code inspections, formal design methods, library control systems and comprehensive testing methods. Prototyping and modern languages should be considered.

Managing software organizations

The basic principles of project organizations are,

- Each project has a plan that is based on a hierarchy of commitments.
- A management system resolves the natural conflicts between the projects and between the line and staff organizations.
- An oversight and review systems audits and tracks progress against the plans.

The foundation for software project management is the commitment discipline. Commitments are supported by plans, reviews and so forth, but the commitments are met by people.

Product and Period Plans

The period (operating) plan deals with technical and business issues in annual and organizational means. Thus expenses, capital requirements and product delivery commitments are established for each period by each organizational entity.

Product plans focus on the activities and objectives of each product. The issues are function, cost, schedule and quality together with related resources and checkpoints.

The distinction between period and product can be confusing. Project personnel view their work as the fundamental business of the organization and have little appreciation for period information.

Organizations, however, are generally measured on a period basis, and the project must be translated into period terms for inclusion in annual budgets, plans or stockholders reports

Contention Process:

An effective management system requires a parallel contention system to encourage the open expression of differences and their rational resolution.

The principle is that the best decisions are based on full understanding of the relavant issues.

The principles of the contention system are,

- All major decisions are reviwed with the involved parties in advance and the parties are requested to agree.. Where possible, any issues are resolved before proceeding.
- When the time comes for a decision, all dissenting parties are present and asked to state their views.
- When there is no agreement, senior management determines if there is knowledgeable agreement, if any disagreeing parties are absent (or) if more preparation is needed.

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.

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 benefits of each major process activity
* Establish a process database and the resources to manage and maintain it.
* Provide sufficient process resources to gather and maintain 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.
* Assess the relative quality of each project and inform management where quality targets are not being met. Should be done by an independent quality assurance group.

The Managed Process (Level 4)

Larger 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 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.

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 gathered 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 Optimized Process (Level 5)

To this point software development managers have largely focused on their products and will typically gather and analyze 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. Examples are errors involving interfaces, performance, human factors and error recovery

So, there are two aspects of testing removal of defects and assessment of program quality. To reduce the cost of removing defects, inspections should be emphasized.

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 evidance is available on the effectiveness with which the process has been applied to any given product.

Process Reference models

CMM, CMMI, P-CMM, PSP, TSP

Capability maturity model (CMM)

The CMM defines five levels of software process maturity based on an organization's support for certain "key" process areas (KPAs).

Level 1 KPAs: A level 1 (initial) process describes an organization with an immature (or) undefined process.

Level 2 KPAs: Requirements management, software project planning, software project tracking and over sight, software subcontract management, software quality assurance, software configuration management.

Level 3 KPAs: Organizational process focus, organizational process definition, training program, integrated software management, intergroup coordination, peer reviews.

Level 4 KPAs: Process measurement and analysis, quality management, defect.

Level 5 KPAs: Technology innovation, process change management.

The goal of most organizations is to achieve a level 3 process. A software capability evaluation (SCE) is commonly used to assess an organization's maturity.

A typical SCE uses the SEI maturity Questionnaire as part of a thorough audit. The evaluation includes detailed analysis, interviews and other forms of assessment. The questionnaire is generally used as entry point to provide context for initiating the evaluation.

One of the key drawbacks of the SEI cmm is that the KPAs focus primarily on the document artifacts of the conventional process such as design, requirements and traceability documents as well as subcontracts, contracts, plans and reports.

In practice, the real indicator of process maturity is the level of predictability in project performance. correlating project performance to the five levels of cmm maturity should demonstrate the following trends.

Level-1: Has random (unpredictable) performance.

Level-2: Achieves repeatable performance from project to project

Level-4: Demonstrate project performance that improves on subsequent projects either substantially in one dimension of performance (or significantly across multiple dimensions (for example, cost and quality)

Level-3: Shows better performance on successive projects in terms of cost, schedule or quality

Level-5: Corresponds to off-scale performance on subsequent projects or substantial improvement across all dimensions. Level 5 organizations almost always occupy a very narrow niche

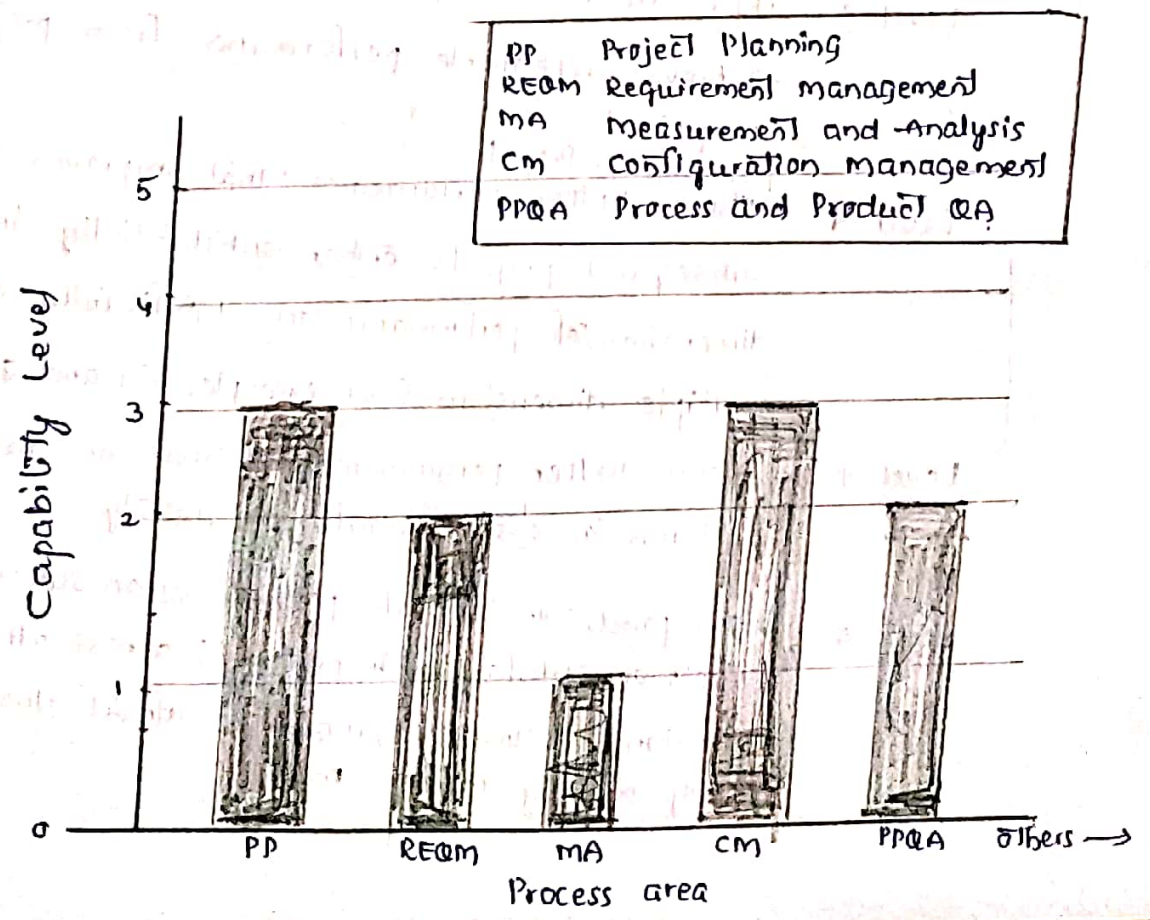
Capability maturity model Integration (CMMI)

The Software Engineering Institute (SEI) has developed a comprehensive process meta model. The SEI contends that an organization should develop a process model that conforms to the Capability maturity model Integration (CMMI) guidelines.

The CMMI represents a process meta model in two different ways

- as a continuous model and
- as a staged model

The continuous CMMI meta model describes a process in two dimensions as illustrated in fig. Each process area eg: project planning (or) requirements management is formally assessed against specific goals and practices and is rated according to the following capability levels.



Level 0 : Incomplete.

The process area (e.g. requirements management) either not performed or doesnot achieve all goals and objectives defined by the cmm3 for level 1 capability

Level 1 : Performed.

All of the specific goals of the process area as defined by cmm3 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 confirms to an organizationally defined policy; all people doing the work have access to adequate resources to get the job done; 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 organization's set of standard processes according to the organization's tailoring guidelines and contributes 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.

Level 5: Optimized

All capability 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 "specific practices" required to achieve these goals.

'Specific goals' establish the characteristics that must exist if the activities implied by a process area to be effective.

'Specific practices' refine a goal into a set of process related activities.

For example, project planning is one of eight process areas defined by the CMMI for the 'Project management' category. The specific goals (SG) and specific practices (SP) defined for project planning are

SG 1 Establish estimates

SP 1.1-1 Estimate the scope of the project

SP 1.2-1 Establish estimates of work product and task attributes

SP 1.3-1 Define project life cycle

SP 1.4-1 Determine estimate of effort and cost

SG 2 Develop a Project Plan

SP 2.1-1 Establish the budget and schedule

SP 2.2-1 Identify project risks

SP 2.3-1 Plan for data management

SP 2.4-1 Plan for project resources

SP 2.5-1 Plan for needed knowledge and skills

SP 2.6-1 Plan for stakeholders involvement

SP 2.7-1 Establish the project plan

SG 3 Obtain commitment to plan

SP 3.1-1 Review plans that affect the project

SP 3.2-1 Reconcile work and resource levels

SP 3.3-1 Obtain plan commitment

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

To illustrate, the generic goal (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 an 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 subprocess performance

GG 5 Institutionalize an optimizing process

GP 5.1 Ensure continuous process improvement

GP 5.2 Correct root causes of problems

The staged CMMI model defines the same process areas, goals and practices as the continuous model.

People Capability Maturity Model

(Short names: People CMM, PCMM, P-CMM)

It is a maturity framework that focuses on continuously improving the management and development of the human assets of an organization.

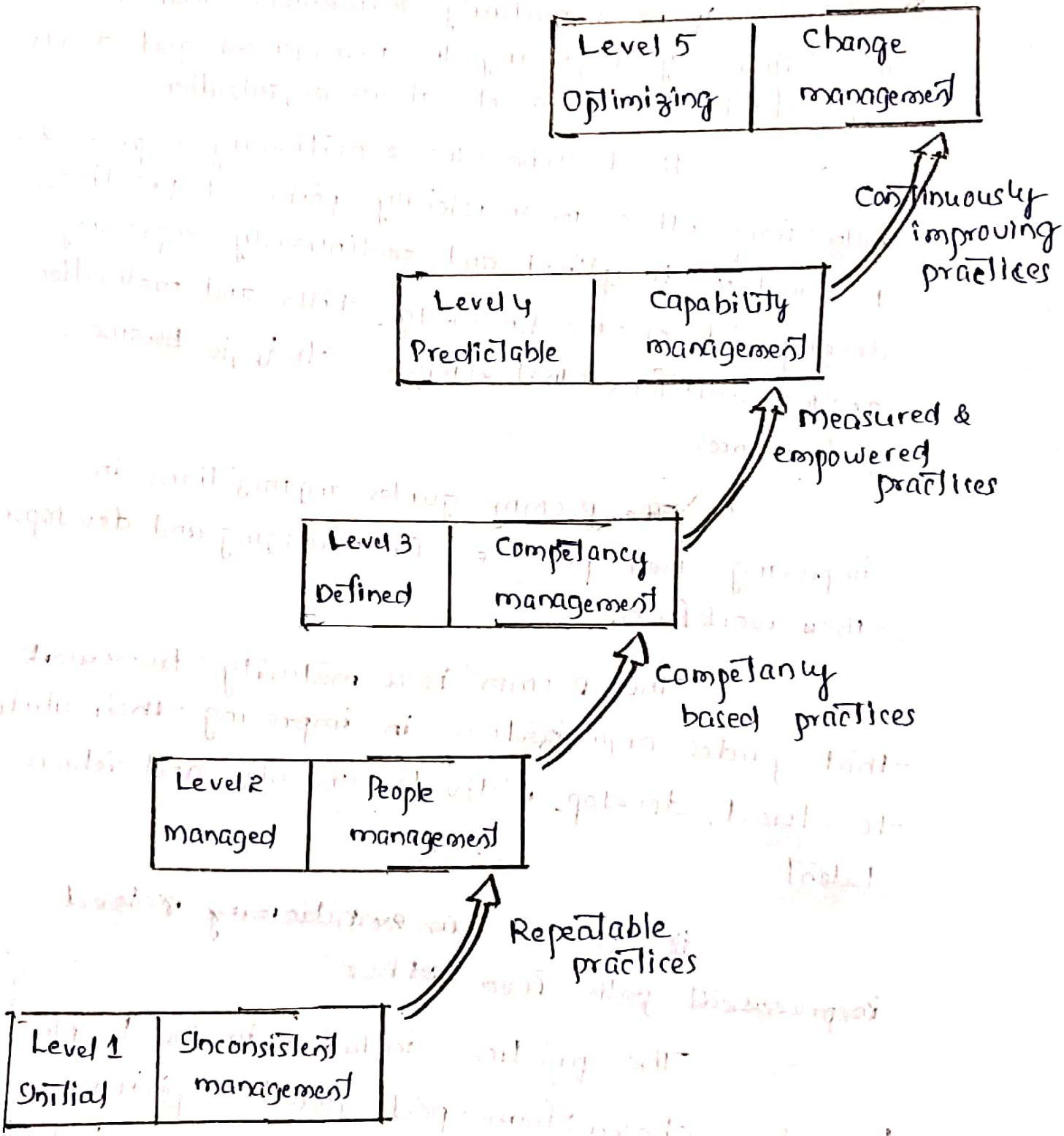
It describes the evolutionary improvement path from adhoc, inconsistently performed practices, to a mature, disciplined and continuously improving development of the knowledge, skills and motivation of the workforce that enhances strategic business performance.

The P-CMM guides organizations in improving their processes for managing and developing their workforces.

The P-CMM is a maturity framework that guides organizations in improving their ability to attract, develop, motivate, organize and retain talent

~~It provides an evolutionary ordered improvement path from adhoc~~

The practices included in the People CMM have been chosen from past global experience because they have significant impact on individual, team, unit and organizational performance.



P-CMM Levels

Personal and Team software Process

The best software process is one that is close to the people who will be doing the work.

If a software process has been developed at a corporate or organizational level, it can be effective only if it is manageable to significant adaptation to meet the needs of the project team that is actually doing software engineering work.

'Walts Humphrey' argues that it is possible to create a "Personal software process" and / or a 'team software process'. Both require hard work, training and coordination, but both are achievable.

Personal software Process (PSP)

- Every developer uses some process to build computer software.
- The process may be haphazard (or) adhoc, may change on a daily basis, may not be efficient, effective (or) even successful, but a process does exist.
- Walts Humphrey suggests that in order to change an ineffective personal process, an individual must move through four phases, each requiring training and careful instrumentation.

- The PSP emphasizes personal measurement of both the work product that is produced and the resultant quality of the work product.
- The PSP define five framework activities:
  - Planning
  - High-level design
  - High-level design review
  - Development
  - Postmortem

Planning: This activity isolates requirements and based on these develops both size and resource estimates. In addition, a defect estimate (the number of defects projected for the work). 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 uncertainity exists. All issues are recorded and tracked.

High-level design review: Formal verification methods are applied to uncover error 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 tasks 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. This is accomplished through a rigorous assessment activity performed on all work products produced by the software engineer.

Team Software Process (TSP)

- Because many industry-grade software projects are addressed by a team of practitioners.
- Watts Humphrey extended the lessons learned from the introduction of PSP and proposed a "team software process (TSP).
- The goal of TSP is to build a "self-directed" project team that organizes itself to produce high quality software.

Humphrey defines the following objectives for TSP:

* Build self-directed teams that plan and track their work. These can be pure s/w teams of 3 to about 20 engineers.
* Show managers how to coach and motivate their teams and how to help them sustain peak performance.
* Accelerate s/w process improvement by making CMM level 5 behaviour normal and expected.
* Provide improvement guidance to high-maturity organizations.
* Facilitate university teaching of industrial-grade team skills.

Each project is 'launched' using a sequence of tasks that enables the team to establish a solid basis for starting the project. The following launch script is advised:

- Review project objectives with management and agree on and document team goals.
- Establish team roles.
- Define the team's development process.
- Make a quality plan and set quality targets.
- Plan for the needed support facilities.
- Produce an overall development strategy.
- Make a development plan for the entire project.
- Make detailed plans for each engineer for the next phase
- Merge the individual plans into a team
- Rebalance team workload to achieve a minimum overall schedule
- Assess project risks and assign tracking responsibility for each key risks.

UNIT-I

Short Answer Questions

1) To improve the software capabilities, organizations must take six steps, List them.
2) What are process maturity Levels?
3) List the six basic principles of s/w process change.
4) Who are champions ?
5) Who are agents ?
6) What is refreezing ?
7) List the five assessment principles.
8) What are specific goals of cmmi.
9) Distinguish PSP and TSP.

Long Answer Questions

1) Explain about software maturity Framework?
2) Explain about Principles of software Process Change?
3) Write about s/w Process assessment.
4) Explain about the Initial process
5) Explain about repeatable process
6) Explain about defined and managed process.
7) Explain about optimizing process.
8) Explain about cmm and cmmi process reference model.