CHAPTER: 5 SOFTWARE QUALITY ASSURANCE AND SECURITY

(14 Marks)

5.1 Project Scheduling: Basic principles, work breakdown structure, activity network & critical path method, scheduling technique (CPM, PERT)
5.2 Project tracking: Timeline charts, earned value analysis, Gantt Charts.
5.3 Software Quality Management vs. Software Quality Assurance, Phase of Software Quality Assurance: Planning, Activities, Audit & Review.
5.4 Quality evaluation standards, Six Sigma, ISO for software, CMMI: Levels, Process areas.
5.5 Software Security, introduction to DevOps, Secure software engineering

---

5.1 Project Scheduling:

- Project Scheduling important task to complete in decided timing is quite difficult.
- There are many reasons for delay in developing software product.
- Ex. Technical problem, designing problem or anything else.
- Factors affecting software product scheduling:-
  1. People work relationships
  2. Task definition & parallelism
  3. Effort distribution
  4. Scheduling methods
- Software project scheduling is an action that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks.
- **Project Scheduling Process**
  - It divides the project into various tasks and estimates the duration and resources required to complete each task.

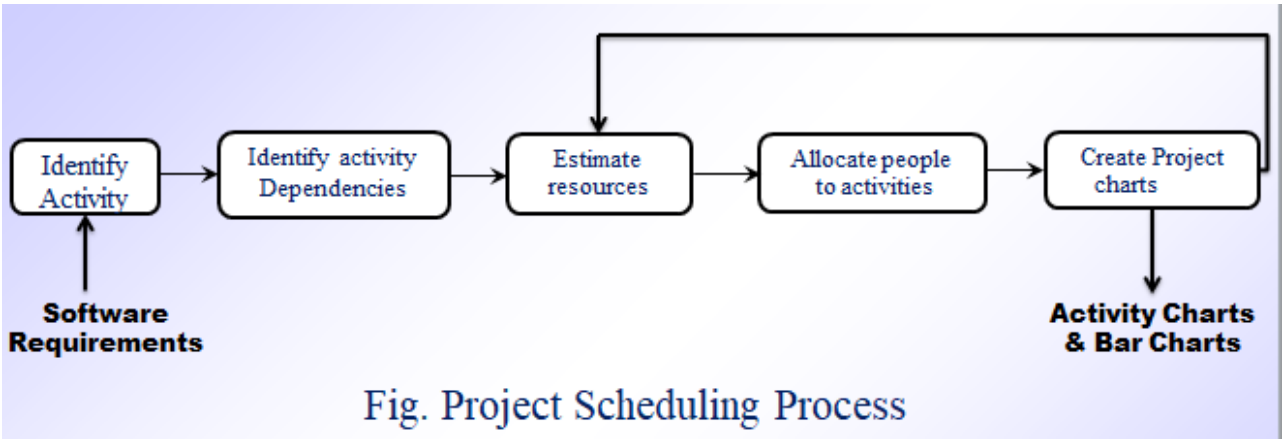


Fig. Project Scheduling Process

- **1. Identify Activity**: - identify specific activities that must be performing to produce various project deliverables.
- **2. Identify activity dependencies:** - identifying & documenting interactivity dependencies.
- **3. Allocates resources:** - resources are allocated & estimating number of work periods which will be needed to complete individual activity
- **4. Create project charts:** - project activity charts are created to analyzing activity sequences, duration& resources requirements to create a project schedule
- **5. Allocate people to activity: -** according to different activity people are allocated to that activity.

Factors that delay project schedule

- An unrealistic deadline established outside the team.
- Changing customer requirements that are not reflected in schedule changes.
- Underestimating the resource required to complete project.
- Risk that was not considered when project began.
- Technical difficulties that could not have been predicted in advance.
- Human difficulties that could not have been predicted in advance.
- Miscommunication among project staff that results in delays.
- A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem.

5.1.1 Basic Principles

Q. List and explain basic principles of project scheduling. (S-19, W-19)

- **Compartmentalization**
  - The project must be compartmentalized into a number of manageable activities, actions, and tasks; both the product and the process are decomposed.
- **Interdependency**
  - The interdependency of each compartmentalized activity, action, or task must be determined.
  - Some tasks must occur in sequence while others can occur in parallel.
  - Some actions or activities cannot commence until the work product produced by another is available.
- **Time allocation**
  - Each task to be scheduled must be allocated some number of work units.
  - In addition, each task must be assigned a start date and a completion date that is a function of the interdependencies.

  - Start and stop dates are also established based on whether work will be conducted on a full-time or part-time basis.
- **Effort validation**
  - Every project has a defined number of people on the team.
  - As time allocation occurs, the project manager must ensure that no more than the allocated numbers of people have been scheduled at any given time.
- **Defined responsibilities**
  - Every task that is scheduled should be assigned to a specific team member.
- **Defined outcomes**
  - Every task that is scheduled should have a defined outcome for software projects such as a work product or part of a work product.
  - Work products are often combined in deliverables.
- **Defined milestones**
  - Every task or group of tasks should be associated with a project milestone.
  - A milestone is accomplished when one or more work products has been reviewed for quality and has been approved.

5.1.2 Work Breakdown Structure (WBS)

- Every one of us has some goals and we plan accordingly to achieve them.
- One of the effective ways to reach our goal quickly is to break down larger goals into realistic achievable steps.
- A work breakdown structure (WBS), as the name says, is the process or technique of dividing complex and difficult projects into smaller units.
- This smaller unit can be a data, product, service or any combination.
- ***A Work Breakdown Structure (WBS) is a hierarchical structure of things that the project will make or outcomes that it will deliver"***
- In other words, WBS is a hierarchical decomposition of work that must be performed to achieve the objective.

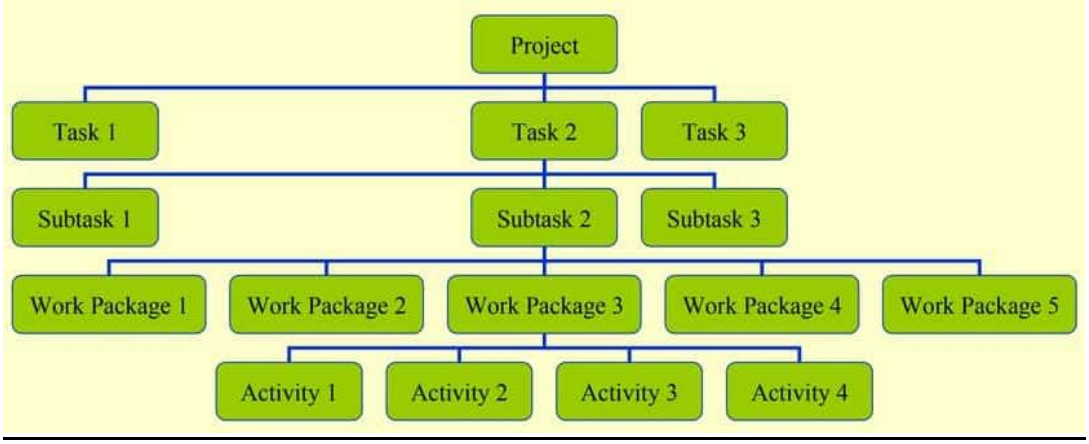
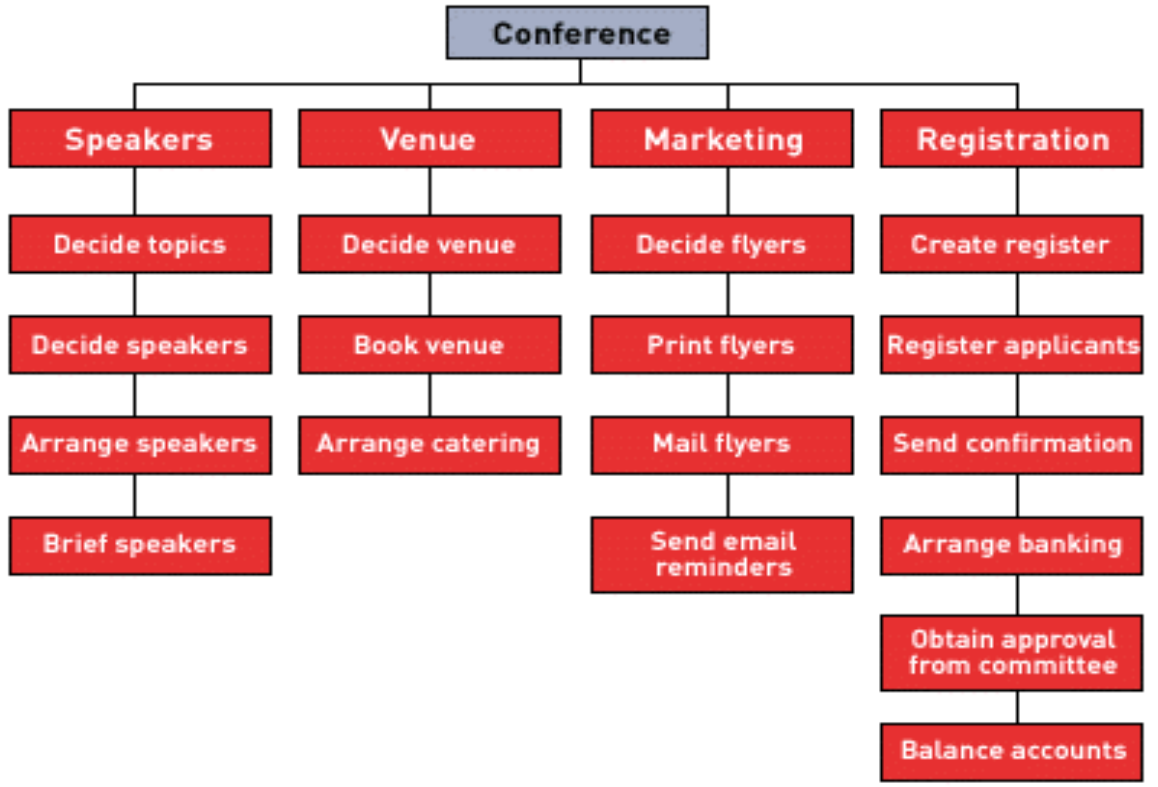


Fig. A Sample Example of WBS

- To constructs a WBS, project managers or subject matter experts first identify the main functional deliverables and break them into tasks, sub-tasks and further into work packages.
- The work package is a list of tasks or activity or "to-dos" to produce a specific or particular unit of work.
- **Example**



Advantages/Importance of WBS:

1. Easy to define, organize and manage the project.
2. Improves the efficiency of the project.
3. Helps to estimate the resources required, such as cost, time, staff, etc.
4. Easy allocation of resources based on the importance of the task/sub-task.
5. Helps to set up milestones in the project.
6. Gives a better understanding of the project to the stakeholders.
7. Easy to identify potential risks in a given project.
8. Helps in identifying communication points and formulating communication plan across the project team.

5.1.3 Activity Network or Task Network

- Also called an activity network
- It is a graphic representation of the task flow for a project.
- It depicts task length, sequence, concurrency, and dependency.
- Points out inter-task dependencies to help the manager ensure continuous progress toward project completion.
- The critical path a task networks there is a certain difficult path called as critical path.
  - A single path leading from start to finish in a task network.
  - It contains the sequence of tasks that must be completed on schedule if the project as a whole is to be completed on schedule.
  - It also determines the minimum duration of the project.

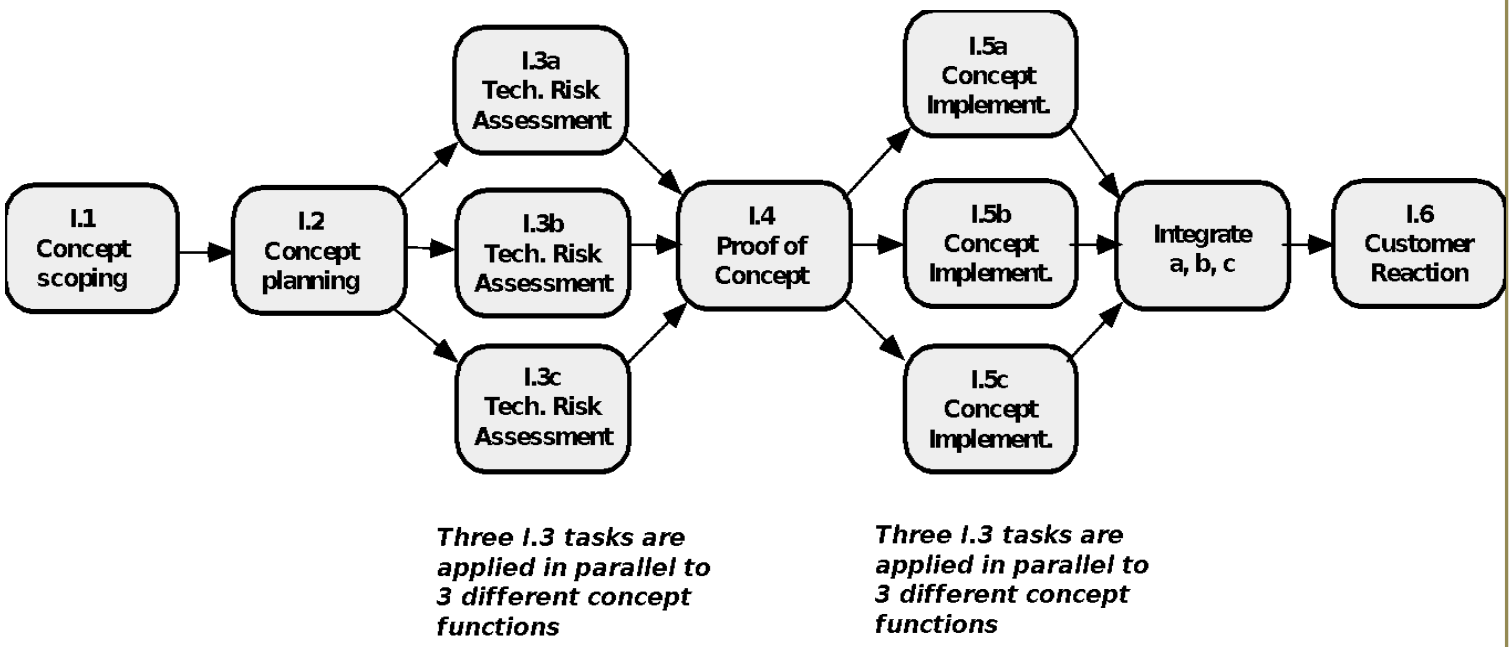
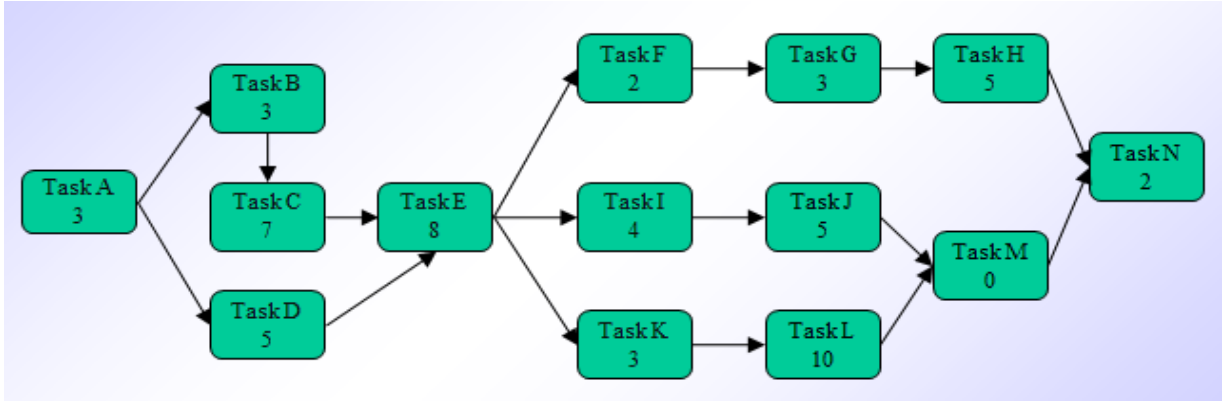


Fig. Activity Network or Task Network

- **Example Task Network**



5.1.4 Project Scheduling Techniques

1. PERT (Program evaluation & review technique)

- PERT is a project management planning tool used to calculate the amount of time it will take to realistically finish a project.
- PERT charts are tools used to plan tasks within a project - making it easier to schedule and coordinate team members accomplishing the work.
- It is useful in basic managerial functions of planning, scheduling & control.
- PERT is used to schedule, organize & coordinate tasks within the project.
- A simple PERT chart comprises circles (nodes) to represent events within the development lifecycle.
- **The basic steps to constructing a PERT chart are:**
  - Identify tasks and estimate duration of times.
  - Identify a single start and end event.
  - Arrange events in sequence (give events a unique number).
  - Establish start and finish times of each task. Keep in mind the estimates made for duration and effort.
  - Determine float
  - Revise

➢ **PERT Example**

- As an example of using a PERT chart, consider the following simple chart showing project with tasks A, B, C, D and E.
- This diagram states that tasks A, B, C and E will take 2 days (assume d is abbreviation for days) and task D has a planned duration of 5 days.
- Task D is dependent on completion of task B, etc.

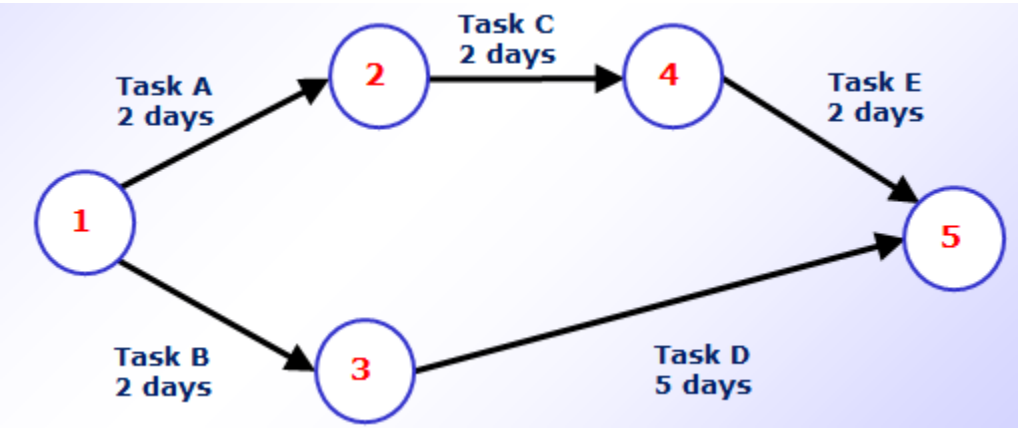


Fig. PERT Example

2. CPM (Critical Path Method)

- Critical path is a sequence of activity between a project's start and finish that takes the longest time to complete.
- Critical path method is based on mathematical calculations and it is used for scheduling project activities.
- The critical path is the path between the start event and end event which takes the longest time.

Note that:

- No task on the critical path can take longer without extending the end date of the project.
- Tasks on the critical path are called critical tasks.
- No critical task can have any slack.
- Tasks on the critical path must be carefully monitored.

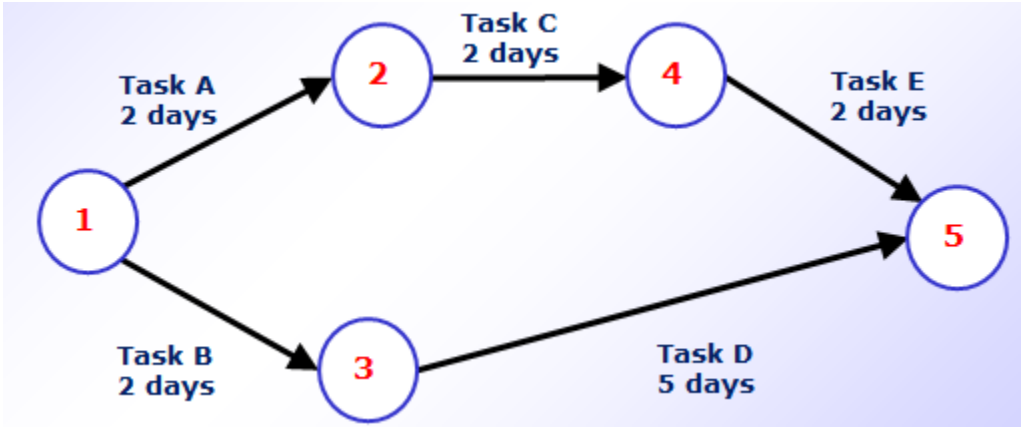
CPM useful for:-

- Identifying tasks which should be completed on time for the entire project to be completed on time.
- It helps in finding the minimum length of time needed to complete a project.

Steps in Critical Path Method

- **Step 1: Make a forward pass through the network as follows: For each activity i beginning at the Start node, compute:**
  Earliest Start Time (ES) = the maximum of the earliest finish times of all activities immediately preceding activity i. (This is 0 for an activity with no predecessors.).
  Earliest Finish Time (EF) = (Earliest Start Time) + (Time to complete activity i. (previous node) this represent the earliest time at which an activity can end.
- **Step 2: Make a backwards pass through the network as follows:**
  Move sequentially backwards from the Finish node to the Start node. At a given node, j, consider all activities ending at node j. For each of these activities, (i,j), compute:
  Latest Finish Time (LF) = the minimum of the latest start times beginning at node j. (For node N, this is the project completion time.). **Latest Start Time (LS)** = (Latest Finish Time) - (Time to complete activity (i,j)).
- **Step 3: Calculate the float time for each activity by:**
  Slack or float = (Latest Start) - (Earliest Start), or
  = (Latest Finish) - (Earlies Finish).

- **Example 1:**



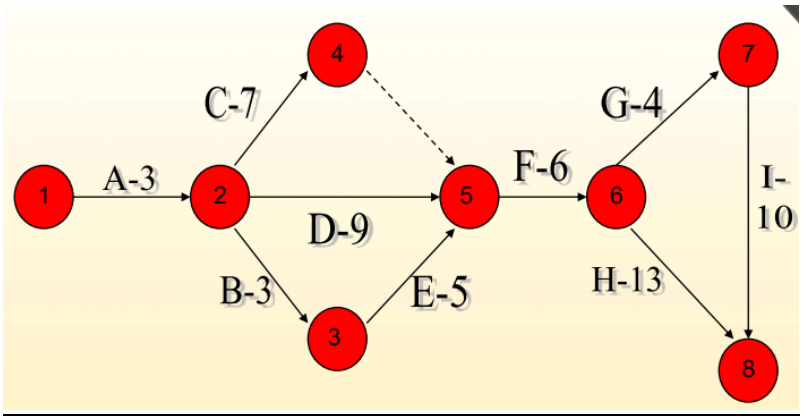
- **Paths in the network diagram :**

  Path 1: B-D = 7 Days

  Path 2: A-C-E= 6 Days

- **In the example above the critical path can be described by events 1,3 and 5 or by tasks B,D.**
- **Path 1:** This is because the time to reach the end event (5) on this path is longer than any other path. This means that task B must take no longer than 2 days and task D no longer than 5 days or the end date for event E will need to be extended.
- The duration of the other path is 6 days. Because the critical path is 7 days, there is slack (or float) of one day on the other path.
- This means that this path can take 1 day longer than planned.
- **Path 2:** That is, any one task on this path (A,C or E) can take 1 day longer than expected. Note this slack must be shared between the tasks on this other path. They cannot all take an extra day.

- **Example 2:**



Paths in the network diagram:

A-D-F-G-I = 32

A-D-F-H = 31

A-C-F-H = 29

A-C-F-G-I = 30

A-B-E-F-H = 30

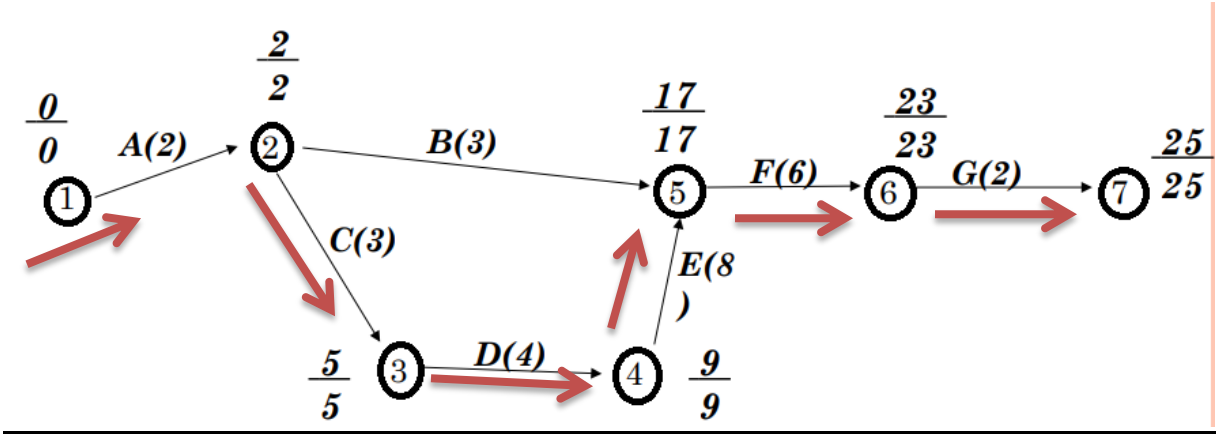
A-B-E-F-G-I = 31

Critical Path : A-D-F-G-I = 32

Example 3: Construct the CPM Network using the details below and determine the critical path

Activity	Immediate predecessors	Completion Time (week)
A	--	2
B	A	3
C	A	3
D	C	4
E	D	8
F	B, E	6
G	F	2

Solution:-



- **Paths in the network diagram :**

  Path 1: A-B-F-G = 13

  Path 2: A-C-D-E-F-G =25

- **Critical Path:-**

  CP = 1 - 2 – 3 - 4 – 5 – 6 – 7

  = A – C -D - E - F - G = 2 + 3 + 4 + 8 + 6 + 2 = 25

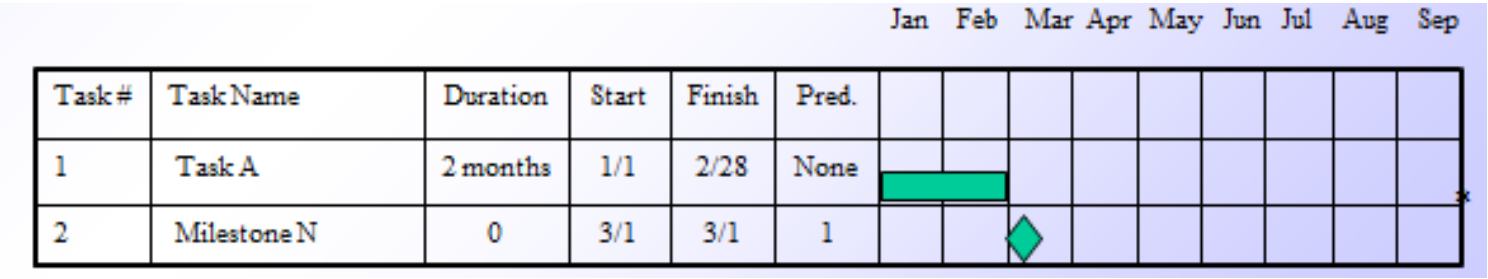
Q. Difference between CPM & PERT

CPM	PERT
CPM works with fixed deterministic time	PERT works with probabilistic time
CPM is useful for repetitive and non-complex projects with a certain degree of time estimates.	PERT is useful for non-repetitive and complex projects with uncertain time estimates.
CPM includes time-cost trade off.	PERT is restricted to time variable.
CPM- for construction projects.	PERT- used for R&D programs.

Q. Explain Gantt chart and its application for project tracking with an example. (W-19)

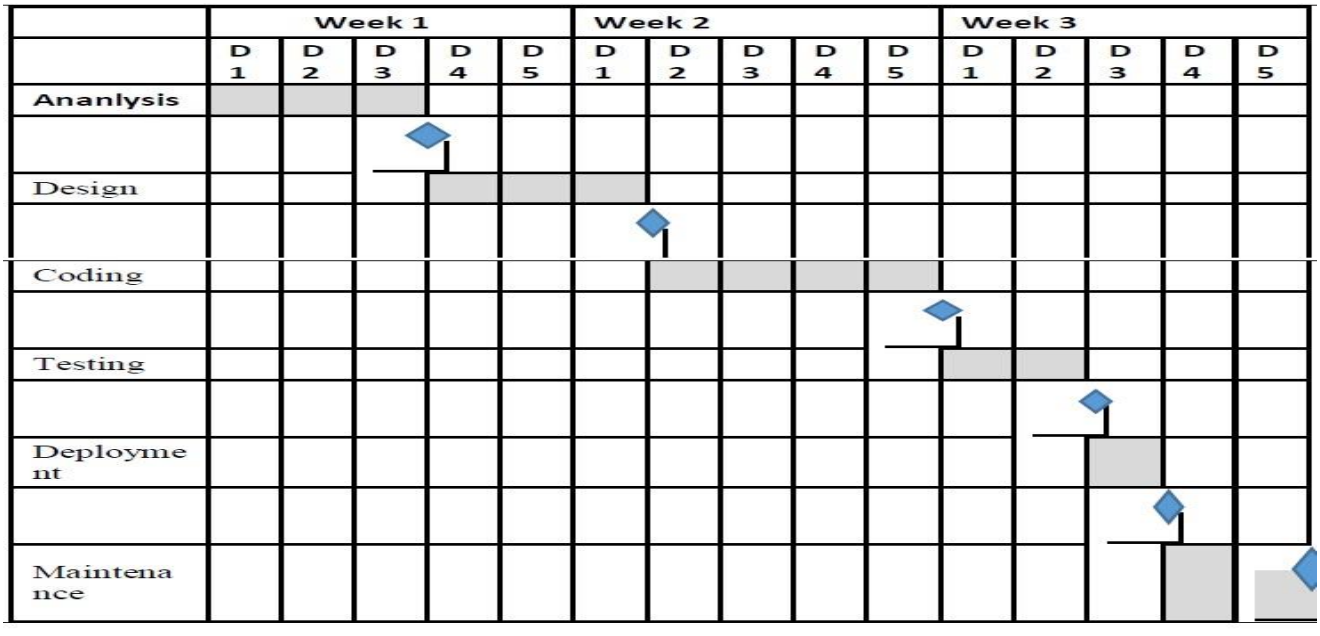
5.2 Project tracking: Timeline Charts (Gantt Charts).

- Also called a Gantt chart; invented by Henry Gantt, industrial engineer, 1917
- GANTT charts are a project planning tool that can be used to represent the timing of tasks required to complete a project.
- All project tasks are listed in the far left column.
- The next few columns may list the following for each task: projected start date, projected stop date, projected duration, actual start date, actual stop date, actual duration, task inter-dependencies (i.e., predecessors)
- To the far right are columns representing dates on a calendar.
- The length of a horizontal bar on the calendar indicates the duration of the task.
- When multiple bars occur at the same time interval on the calendar, this implies task concurrency
- A diamond in the calendar area of a specific task indicates that the task is a milestone; a **milestone** has a time duration of zero

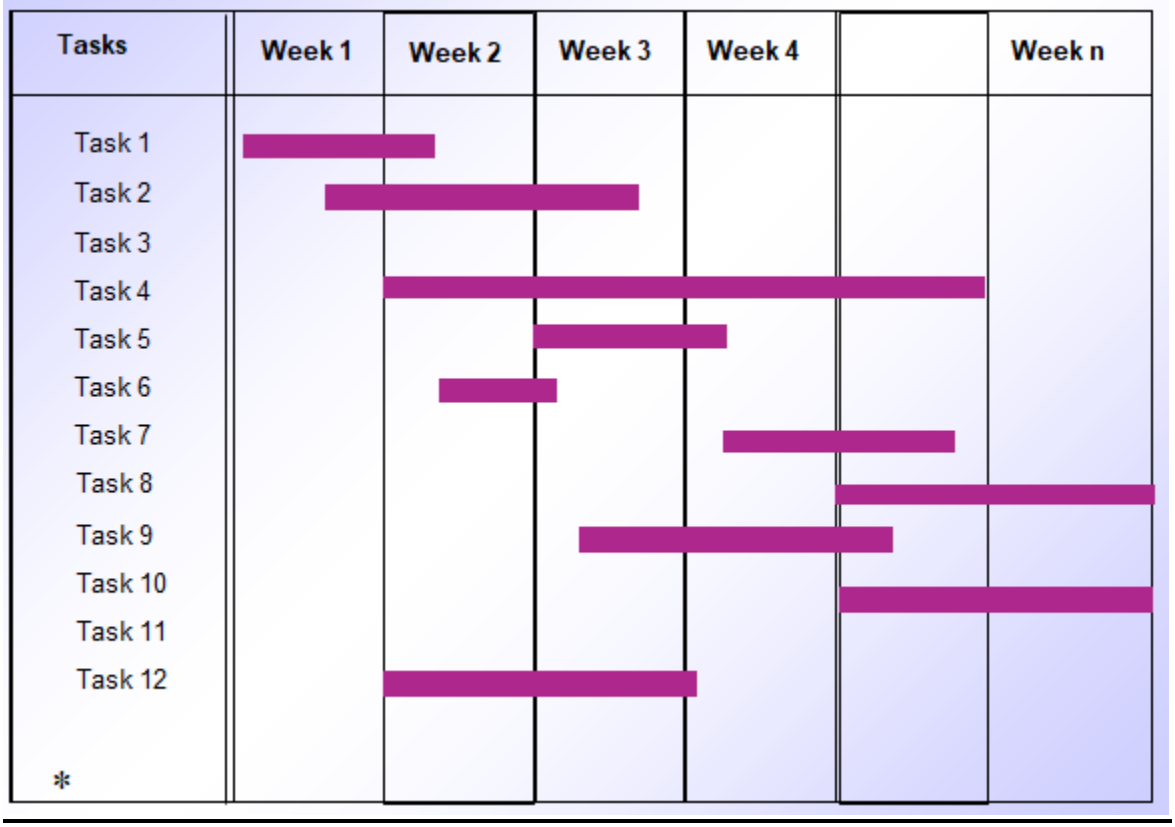


Task #	Task Name	Duration	Start	Finish	Pred.
1	Task A	2 months	1/1	2/28	None
2	Milestone N	0	3/1	3/1	1

Q. Draw time chart for Library Management System (5 days a week). Consider broad phases of SDLC. (S-19)



Timeline Charts Example:



Application of Gantt Chart

- The sheer simplicity and ease-of-access of all relevant information make Gantt charts an ideal choice for teams to use them for organizing their schedules. Due to this, Gantt charts are widely used in project management, IT and development teams.
- Apart from them, marketing, engineering, product launch, manufacturing teams can also use Gantt charts to get an overview of how things are rolling on the work front.

5.3 Software Quality Management vs. Software Quality Assurance

Definition of Quality:-

- Defined as a "characteristic or attribute of something"
- Refers to measurable characteristics that we can compare to known standards.

Software Quality: - Q. Define software quality. (W-19)

- It is defined as the degree to which a system, components, or process meets specified requirements.
  OR
- The degree to which a system, components or process meets customer or user needs or expectation.
- Reduces the amount of rework, which results in lower costs and improved time to market
- Encompasses
  - A software quality assurance process.
  - Specific quality assurance and quality control tasks (including formal technical reviews and a multi-tiered testing strategy).
  - Effective software engineering practices (methods and tools).
  - Control of all software work products and the changes made to them.
  - A procedure to ensure compliance with software development standards.
  - Measurement and reporting mechanisms.

5.3.1 Software Quality Control:-

- It focuses on operational techniques & the activities used to fulfill & verify requirements of quality.
- Involves a series of inspections, reviews, and tests used throughout the software process
- Key concept of quality control is that-"all work products have defined & measurable specifications"
- Ensures that each work product meets the requirements placed on it.
- Includes a feedback loop to the process that created the work product.
  - This is essential in minimizing the errors produced.
- Combines measurement and feedback in order to adjust the process when product specifications are not met.
- Requires all work products to have defined, measurable specifications to which practitioners may compare to the output of each process.
- Feedback loop useful to minimize the defects in products.

5.3.2 Software Quality Assurance: - Q. Define Software Quality Assurance. (S-19)

- SQA consists of a means of monitoring the Software engineering Processes & methods used to ensure quality.
- SQA is a planned & systematic way to evaluate quality of software product standards, processes & procedures.
- Quality assurance consists of the auditing and reporting functions of management.
- The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is meeting its goals.
- **SQA is an umbrella activity that is applied though software process.**
  1. An SQA process.
  2. Specific quality assurance & quality control tasks.
  3. Effective Software engineering Practice.
  4. Control of all software product.
  5. A procedure to ensure compliance with software development standards.
  6. Measurement & reporting mechanism.

5.3.2.1 SQA Activities:- (W-19)

- **Prepares an SQA plan for a project.**
  - Plan is developed & is reviewed by all the people who are involved in project.
- **Participates in the development of the project's software process description.**
  - The software team selects a work process to be performed.
  - SQA group reviews the process description for compliance with organizational policy, internal & external software standards.
- **Reviews software engineering activities (analysis, design, construction, verification, & management) to verify compliance with the defined software process.**
- **Audits designated software work products to verify compliance with those defined as part of the software process:-**
  - SQA group also verifies that corrections have been made, & periodically reports the results of its work to the project manager.
- **Ensures that deviations in software work and work products are documented and handled according to a documented procedure**
- **Records any noncompliance and reports to senior management**
- **Configuration management monitoring :-**
  - It is also called "change control management".
  - controlling the changes to the software items.
  - It helps to store & retrieve the configurable.
- **Helps to collect and analyze software metrics.**

Q. Differentiate between Software Quality (Control) Management and Software Quality Assurance (S-19).

Sr. No.	Software Quality (Control) Management (QC)	Software Quality Assurance (QA)
1	It is a procedure that focuses on fulfilling the quality requested.	It is a procedure that focuses on providing assurance that quality requested will be achieved.
2	QC aims to identify and fix defects.	QA aims to prevent the defect
3	It is a method to verify the quality-Validation	It is a method to manage the quality- Verification
4	It always involves executing a program	It does not involve executing the program
5	It's a Corrective technique	It's a Preventive technique
6	It's a Reactive measure	It's a Proactive measure
7	It is the procedure to verify that deliverables	It is the procedure to create the deliverables
8	QC involves in full software testing life cycle	QA involves in full software development life cycle
9	QC confirms that the standards are followed while working on the product	In order to meet the customer requirements, QA defines standards and methodologies
10	It is performed only after QA activity is done	It is performed before Quality Control
11	It is a High-Level Activity, it can identify an error that QA cannot	It is a Low-Level Activity, it can identify an error and mistakes which QC cannot

5.4 Quality evaluation standards

5.4.1 Six Sigma:-

- Popularized by Motorola in the 1980s
- Is the most widely used strategy for statistical quality assurance.
- Uses data and statistical analysis to measure and improve a company's operational performance.
- Identifies and eliminates defects in manufacturing and service-related processes.
- It used to eliminate error or faults or bugs in software by evaluating the performance of the software process.
- It minimizes the cost of poor quality.
- It monitors day-to-day activities of an organization.
- **The "Six Sigma" refers to six standard deviations.**
- **Three core steps**

- **Define** customer requirements, deliverables, and project goals via well-defined methods of customer communication
- **Measure** the existing process and its output to determine current quality performance (collect defect metrics)
- **Analyze** defect metrics and determine the vital few causes (the 20%)

- **Two additional steps are added for existing processes (and can be done in parallel)**
  - **Improve** the process by eliminating the root causes of defects
  - **Control** the process to ensure that future work does not reintroduce the causes of defects
- All of these steps need to be performed so that you can manage the process to accomplish something.

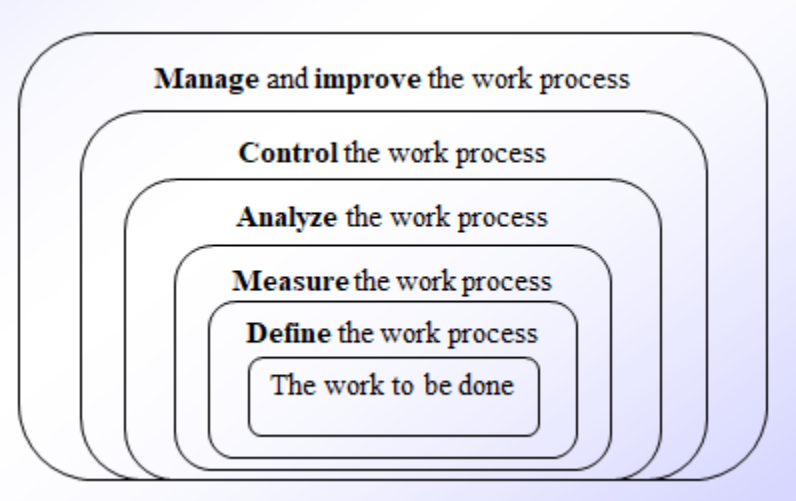


Fig. Layers of Six Sigma

- **Concept of DMAIC:-**

1. **DMAIC(Define, Measure, Analyze, Improve, Control)**
   - DMAIC checks whether a process is performing correctly.
   - DMAIC improves the system by improving the process.
   - It used when a product or process is in existence & is not meeting customer specification.

Sr. No.	Attributes	Functions
1.	Define	Identify the goal of project & user requirement.
2.	Measure	Quantify the process to determine the current performance of the s/w.
3.	Analyze	Examine & determine the causes of the error.
4.	Improve	Improve the process by eliminating errors.
5.	Control	Determine the future process performance for prevention of errors.

- **Concept of DMDAV**

- **DMDAV(Define, Measure, Design, Analyze, verify)**
  - It used when product or process is in existence at an organization but is not meeting customer specification or requirements.

Sr. No.	Attributes	Functions
1.	Define	Specifies the goal of project & user requirements.
2.	Measure	Evaluate and determine user requirements.
3.	Analyze	Identify performance goals and determine how process inputs are likely to affect process outputs.
4.	Design	Work out details, optimize the methods, run simulations if necessary and plan for design verification.
5.	Verify	The process model will avoid the defects and meets the customer requirements.

5.4.2 ISO 9000 quality standards

- International organization for standardizations established the term ISO 9000.
- It refers to a set of quality management standard.
- The ISO 9000 standards are maintained by ISO and administered by accreditation and certifications bodies.
- ISO first published its quality standard in1987, revised them in 1994 and then republished an updated version in 2000, then 2008, then 2015 version and new standard are referred as "ISO 9000:2015 standards".
- ISO standard applied to all kind of organization in all kind of areas include manufacturing, processing, printing, electronics, banking, telecommunications, agriculture, government, educations, software development, biotechnology and so on.
- Experience and knowledge from international experts who contributed in ISO technical committee.
- **Types of ISO 9000 Quality Standards**

1. ISO 9001: This standard applies to the organizations engaged in design, development, production, and servicing of goods. This is the standard that applies to most software development organizations.

2. ISO 9002: This standard applies to those organizations which do not design products but are only involved in the production. Examples of these category industries contain steel and car

manufacturing industries. Therefore, ISO 9002 does not apply to software development organizations.

ISO 9003: This standard applies to organizations that are involved only in the installation and testing of the products. For example, Gas companies.

- **How to get ISO 9000 Certification?**
  - **Application:** Once an organization decided to go for ISO certification, it applies to the registrar for registration.
  - **Pre-Assessment:** During this stage, the registrar makes a rough assessment of the organization.
  - **Document review and Adequacy of Audit:** During this stage, the registrar reviews the document submitted by the organization and suggest an improvement.
  - **Compliance Audit:** During this stage, the registrar checks whether the organization has compiled the suggestion made by it during the review or not.
  - **Registration:** The Registrar awards the ISO certification after the successful completion of all the phases.
  - **Continued Inspection:** The registrar continued to monitor the organization time by time.
- **Benefits ISO standard**
  - **Customer focus.**
  - **Leadership.**
  - **Involvement of people.**
  - **Process approach.**
  - **System approach to management.**
  - **Continuous improvement.**
  - **Decision making.**
  - **Mutually beneficial supplier relationships.**

5.4.3 CMMI: Levels Q. Describe CMMI. Give significance of each level. (S-19, 6 Marks)

- **CMMI History**
- CMMI (Capability Maturity Model Integration) is a proven industry framework to improve product quality and development efficiency for both hardware and software.
- Sponsored by US Department of Defence in cooperation with Carnegie Mellon University and the Software Engineering Institute (SEI)
- Many companies have been involved in CMMI definition such as Motorola and Ericsson
- CMMI has been established as a model to improve business results

- A Capability Maturity Model (CMM) is a reference model of mature practices in a specified discipline, **used to improve and appraise a group's capability to perform that discipline**.
- CMM contains a list of key process areas that includes software engineering, project management, and process improvement divided into different levels
- CMMI, staged, uses 5 levels to describe the maturity of the organization,
- CMMI provides a way to focus and manage hardware and software development from product inception through deployment and maintenance.
- Behavioral changes are needed at both management and staff levels
- Initially a lot of investment required – but, if properly managed, we will be more efficient and productive while turning out products with consistently higher quality.
- **CMMI Staged Representation - 5 Maturity Levels**

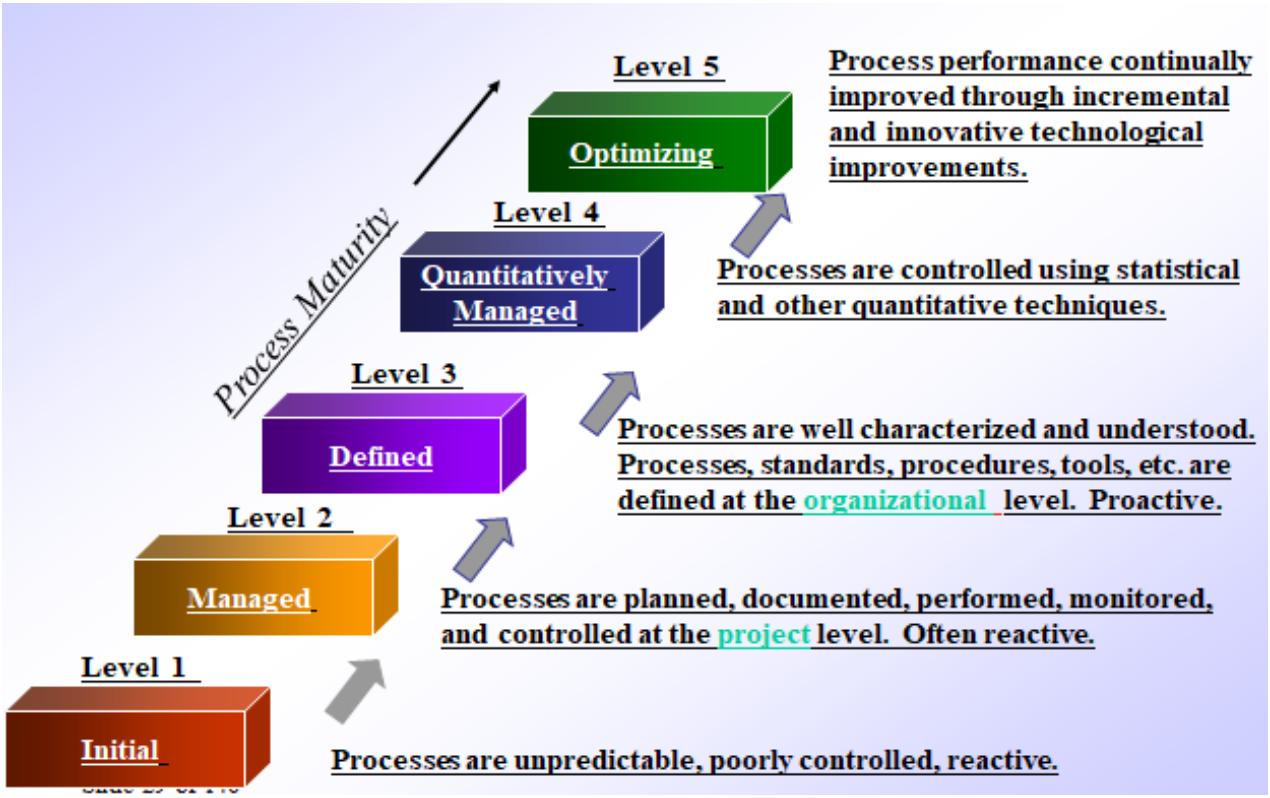


Fig. 5 Maturity Levels of CMMI

- **Maturity Level 0 :- Not performed**
  - One or more specific goals associated with the process area is not satisfied.
- **Maturity Level 1:- Initial or performed**
  - Maturity Level 1 deals with performed processes.
  - Processes are unpredictable, poorly controlled, and reactive.
  - The process performance may not be stable and may not meet specific objectives such as quality, cost, and schedule, but useful work can be done.

- **Maturity Level 2 :- Managed**
  - Maturity Level 2 deals with managed processes.
  - A managed process is a performed process that is also:
    - Planned and executed in accordance with policy
    - Employs skilled people
    - Adequate resources are available
    - Controlled outputs are produced
    - Stakeholders are involved
    - The process is reviewed and evaluated for adherence to requirements
  - Processes are planned, documented, performed, monitored, and controlled at the project level. Often reactive.
  - The managed process comes closer to achieving the specific objectives such as quality, cost, and schedule.
- **Maturity Level 3: Defined at the Organization Level**
  - Maturity Level 3 deals with defined processes.
  - A defined process is a managed process that:
    - Well defined, understood, deployed and executed across the entire organization. Proactive.
    - Processes, standards, procedures, tools, etc. are defined at the organizational (Organization X) level.
- **Maturity Level 4 :- Quantitatively Managed**

  - There is an organizational responsibility to use statistical & other qualitative methods to control sub-processes.

  -i.e. collected process & product measurements must be used in process management.
- **Maturity Level 6:- Optimizing**

  - At this highest level, the organization must use the process & product measurements to drive process improvement.

Q. Compare CMMI and ISO for software w.r.to (W-19)

i)scope
ii)Approach
iii) Implementation.

Sr. No.	CMMI	ISO
1.	Capability maturity model integration	International standard organization
2.	Used for system engg. & s/w development process.	Not for any specific business or service. Used by any organization.
3.	It is a process model	ISO is set of standards.
4.	Very less or no documentation is Required	Registration documentation is Required
5.	Audit not required.	Repeated audits are done.
6.	CMMI is derived from the best practices followed by the leaders in the business.	ISO is certification tools
7.	It is designed & developed in s/w Engg. Institute.	ISO founded in 1947 having it's headquarter at Geneva, Switzerland.
8.	Mainly focuses on inward part of the business process.	focuses on outward part
9.	Focus on continuous quality improvement.	Focuses on minimum requirements with indirect continuous improvements. 34

5.5 Software Security:-

- Software security is an idea implemented to protect software against malicious attack and other hacker risks so that the software continues to function correctly under such potential risks.
- Security is necessary to provide integrity, authentication and availability.
- Any compromise to integrity, authentication and availability makes a software unsecure.
- Software systems can be attacked to steal information, monitor content, introduce vulnerabilities and damage the behavior of software.
- Malware can cause DoS (denial of service) or crash the system itself.

5.5.1 Introduction to DevOps

- An traditional development methodologies, there would be wall of confusion in between the development team and operations team. DevOps overcomes this drawback as it is the collaboration of development and operations.
- DevOps is all about the integration of operations and development process.

- DevOps uses automation and works in small deployments. It is a methodology of continuous delivery.
- DevOps is a software development method that stresses on communication, collaboration and integration between development and IT professionals.
- DevOps is 3-stage conversion:
  1. People- Development and Operation Team.
  2. Process
  3. Products

Benefits of DevOps:

- Deploy code 30 times faster.
- DevOps improves IT performance by improved quality of software deployments.
- With DevOps, you have improved quality of code with 60 times fewer failures.
- Allows more frequent software releases.
- Improved visibility & IT process and requirements.
- More responsiveness to business needs.
- Encourages more agile development.

5.5.2 Secure software engineering

- Software security engineering is about how to develop and maintain software that are resistible to malicious attacks.
- Security engineering is a top priority for business and individuals as more and more intruders try to exploit the network.
- Security in the software development Life Cycle(SDLC) can be implemented in the following way:
  - **Requirement gathering:** Security of requirements. Set the strategies and metrics, compliance and polices.
  - **Design and Analysis:** Risk analysis and review.
  - **Code:** static analysis using tools and risk based security tests.
  - **Deployment:** Penetration testing by breaking security.