Overview of Project Management

UNIT - I

Contents

Project - Definition

- Definition: A project is “a temporary endeavor undertaken to accomplish a unique product or service”.
- Definition: Unique process consisting of a set of coordinated and controlled activities with start and finish dates, undertaken to achieve an objective conforming to specific requirements, including constraints of time, cost, quality and resources.
- Definition: Projects are unique transient endeavors undertaken to achieve a desired outcome.
- Definition: A project is a sequence of unique, complex, and connected activities that have one goal or purpose and that must be completed by a specific c time, within budget, and according to specification.
- Definition: a project is defined as a specific, finite activity that produces an observable and measurable result under certain preset requirements.

Project – Attributes/Characteristics

- A Project has clear objective(s);
- A Project is purposeful as it has a rational and measurable purchase.
- A Project is logical as it has a certain life-cycle.
- A Project has a scope and unique purpose.
- A Project has time, cost, quality and resource constraints.
- A Project involves change and uncertainty.
- A Project should have a primary sponsor and/or customer.
- A Project involves team of people;
- A Project is structured as it has interdependencies between its tasks and activities.

Project Management

A simple definition of project management includes a handful of key premises:

- Project management is no small task.
- Project management has a definite beginning and end. It's not a continuous process.
- Project management uses various tools to measure accomplishments and track project tasks. These include Work Breakdown Structures, Gantt charts and PERT charts.
- Projects frequently need ad-hoc resources rather than dedicated, full-time positions common in organisations.
- Project management reduces risk and increases the chance of success.

- The Project Management Institute (PMI) formally defines project management as follows: “The application of knowledge, skills, tools and techniques to project activities to meet the project requirements.”

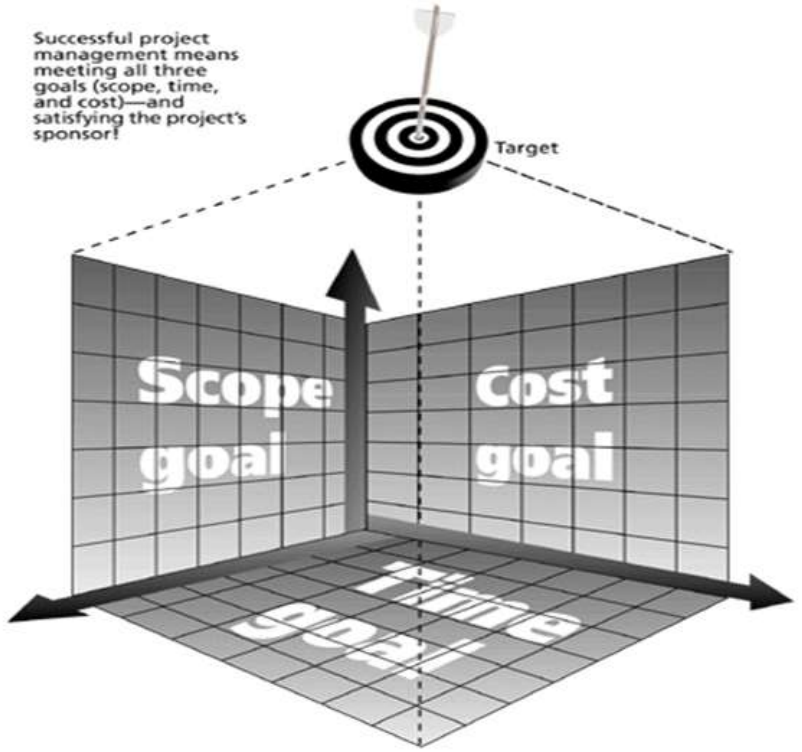
- Defination: The art of organising, leading, reporting and completing a project through people.

Project Management

- Project management is a set of tools, templates, and processes designed to answer the following six questions:
  - What business situation is being addressed by this project?
  - What does the business need to do?
  - What will you do?
  - How will you do it?
  - How will you know you did it?
  - How well did you do?
- Often, a triangle, commonly called the "triple constraint", is used to summarise project management. The three most important factors are time, cost and scope.

The Triple Constraint

- Every project is constrained in different ways by its
  - Scope goals: What is the project trying to accomplish?
  - Time goals: How long should it take to complete?
  - Cost goals: What should it cost?
- These three variables form the sides of a triangle and are an interdependent set. If any one of them changes, at least one other variable must also change to restore balance to the project.



The Triple Constraint of PM

The Triple Constraint

- The greatest challenge of project management is the integration and control of the three principal interrelated components of each single project.
  - Projects must be within cost.
  - Projects must be delivered on time.
  - Projects must be in scope.
  - Projects must meet customer quality requirements.
- The cost, time, quality triangle implies a tension between the three components such that if any one of the components was to be changed, then it would have an impact on one, or both, of the others.
- More recently this has given way to a project management diamond with cost, time, scope and quality as the four vertices and customer expectations as a central theme. No two customer expectations are the same so you must ask what their expectations are.

The Triple Constraint

- Consider the following constraints that operate on every project:
  - Scope
  - Quality
  - Cost
  - Time
  - Resources
  - Risk
- Except for Risk these constraints form an interdependent set—a change in one constraint can require a change in one or more of the other constraints in order to restore the equilibrium of the project.
- In this context, the set of fi ve parameters form a system that must remain in balance for the project to be in balance. Because they are so important to the success or failure of the project, each parameter is discussed individually in this section.

Scope

- Scope is a statement that defines the boundaries of the project.
- It tells not only what will be done, but also what will not be done.
- In the information systems industry, scope is often referred to as a functional specification. In the engineering profession, it is generally called a statement of work.
- Scope may also be referred to as a document of understanding, a scoping statement, a project initiation document, or a project request form. Whatever its name, this document is the foundation for all project work to follow. It is critical that the scope be correct

Quality

- The following two types of quality are part of every project:
  - Product quality—The quality of the deliverable from the project. As used here “product” includes tangible artifacts like hardware and software as well as business processes.
  - Process quality—The quality of the project management process itself. The focus is on how well the project management process works and how it can be improved. Continuous quality improvement and process quality management are the tools used to measure process quality
- A sound quality management program with processes in place that monitor the work in a project is a good investment. Not only does it contribute to client satisfaction, but it helps organizations use their resources more effectively and efficiently by reducing waste and revisions. Quality management is one area that should not be compromised.
- The payoff is a higher probability of successfully completing the project and satisfying the client.

Cost

- The dollar cost of doing the project is another variable that defi nes the project. It is best thought of as the budget that has been established for the project.
- This is particularly important for projects that create deliverables that are sold either commercially or to an external customer.

Time

- The client specifies a time frame or deadline date within which the project must be completed.
- To a certain extent, cost and time are inversely related to one another. The time a project takes to be completed can be reduced, but costs increase as a result

Resources

- Resources are assets such as people, equipment, physical facilities, or inventory that have limited availabilities, can be scheduled, or can be leased from an outside party.
- Some are fi xed; others are variable only in the long term.
- In any case, they are central to the scheduling of project activities and the orderly completion of the project

Risk

- Risk is not an integral part of the scope triangle, but it is always present and spans all parts of the project both external as well as internal, and therefore it does affect the management of the other fi ve constraints.

Barriers, Risks and Issues That Affect Project Success

- Many things can go wrong in project management. Any barriers, risks and issues can affect every phase and process of project management. Here are just some of the things that can possibly go wrong:
  - Poor communication
  - Disagreement
  - Misunderstandings
  - Uncertain weather
  - Union strikes
  - Personality conflicts
  - Poor management
  - Poorly defined goals and objectives

Barriers, Risks and Issues That Affect Project Success

- A good project management discipline will not eliminate all risks, issues and surprises - but it *will* provide standard processes and procedures to deal with them and help prevent the following:
  - Projects finishing late, exceeding budget and not meeting customer expectations
  - Inconsistency between the processes and procedures used by project managers, leading to the favoring of some project managers more than others
  - Successful projects, despite a lack of planning, achieved through high stress levels, goodwill and significant amounts of overtime
  - Project management being seen as not adding value and as a waste of time and money
  - Unforeseen internal and external events impacting the project

Project Management Activities

- Software Project Management consists of many activities, that includes planning of the project, deciding the scope of product, estimation of cost in different terms, scheduling of tasks, etc.
- The list of activities are as follows:
  - Project planning and Tracking
  - Project Resource Management
  - Scope Management
  - Estimation Management
  - Project Risk Management
  - Scheduling Management
  - Project Communication Management
  - Configuration Management

Project Management Life Cycle (PMLC)

A project management life cycle (PMLC) is a sequence of processes that includes:

- Project Initiation
- Project Planning
- Project Execution
- Project Monitoring and controlling
- Project Closing

the projects to which it applies.

- A valid PMLC always starts with a scoping process and ends with a closing process. All five of the processes must each be done at least once and may be repeated any number of times in some logical order.

PMLC


Initiating Process
Planning Process
Executing Process
Monitoring & Controlling Process
Closing Process
PROJECT LIFECYCLE

Project Initiation

- Initiation is first phase of project management life cycle where the feasibility and the business value of the project are determined.
- The key outcome of the Initiating phase is our "**Project Charter**"! It is the bible of your Project!
- Before we undertake any endeavor, we perform certain background checks, initial research, execution feasibility, and commercial viability and then decide if it should be undertaken at all. And exactly that is what is covered in our Project Charter :-
  - Project constraints and Problem statement
  - Goals / Projected Benefits
  - Identify Stakeholders
  - Project Scope (In-scope and Out-of-scope items)
  - Identifying Deliverables
  - Identifying Risks
  - Defining project resources, cost &budget.
  - Identifying stakeholders

Project Initiation

It explains :-

- Why a project was undertaken? What problems need to be addressed?
- What specific strategic gaps and initiatives need to be served?
- What needs, objectives and profits were served by this project?
- Who are the key stakeholders, sponsors and project team?
- What are the roles and responsibilities of each person associated with the project?
- What does the project entail? What will be exactly delivered and left out of the deliverables?
- What is to be delivered, at what time and within how much cost?
- And lastly who is in charge and authorized to run the project?

Project Initiation

- The other activities involved in initiating process group are:
  - Assigning the project manager
  - Determining the stakeholder needs, expectations and high-level requirements
  - Define the project success criteria
  - Identify particular budget for particular stage
  - Make sure that the project is aligned with the organizations strategic goal

Project Planning

- FAILING to PLAN is PLANNING to FAIL.
- Planning is the second yet most important phase in the project management life cycle. Project planning is at the heart of the project life cycle and tells everyone involved where you're going and how you're going to get there.
- The planning phase is when the project plans are documented, the project deliverables and requirements are defined, and the project schedule is created.
- The plans created during this phase will help you manage time, cost, quality, changes, risk, and related issues. They will also help you control staff and external suppliers to ensure that you deliver the project on time, within budget, and within schedule.

Project Planning

- Steps for the project planning phase may include the following:
  - Creating a project plan: Identify the project timeline, including the phases of the project, the tasks to be performed, and possible constraints
  - Creating workflow diagrams: Visualize your processes to make sure team members clearly understand their role in a project
  - Estimating budget and creating a financial plan: Use cost estimates to determine how much to spend on the project to get the maximum return on investment
  - Gathering resources: Build your functional team from internal and external talent pools while making sure everyone has the necessary tools (software, hardware, etc.) to complete their tasks
  - Anticipating risks and potential quality roadblocks: Identify issues that may cause your project to stall while planning to mitigate those risks and maintain the project's quality and timeline
  - Holding a project kickoff meeting: Bring your team on board and outline the project so they can quickly get to work

Project Execution

- This is the phase when the project starts taking it shape.
- The programmers are working on coding, web designers with the graphic material, status and performance reports are made by project managers. This phase is also called implementation phase.
- Here are some of the important things are being taken care of in the implementation phase.
  - **Report progress:** Regular updates and status reports are required when the project is in the execution stage. It's important to provide the required information in the right format and identify the issues as well. These resources will prove beneficial in the times of a crisis.
  - **Hold weekly meetings:** Weekly meetings can save your team deviating from the important activities. Clear agendas should be set for the meeting so that no time is wasted as team members are already well-aware what the meeting is for and the overall productivity doesn't get affected.
  - **Manage problems:** As the project is in motion, problems are bound to happen. You can face issues like quality, time management, the decline in a team's morale that can threaten the success of a project.

Monitoring and Controlling

- In the project management process, the third and fourth phases are not sequential in nature. This phase runs simultaneously with project execution, thereby ensures that objectives and project deliverables are met.
- As a project manager, you can make sure that no one deviates from the original plan by establishing Critical Success Factors (CSF) and Key Performance Indicators (KPI).
- During the monitoring phase of project management, the manager is also responsible for quantitatively tracking the effort and cost during the process. This tracking not only ensures that the project remains within the budget but also is important for future projects.
- Some of the common KPIs to measure project performance are:
  - **Project objectives:** If a project stays on schedule and desired budget, it's an indication that it will meet the expectations of the decision-makers and clients.
  - **Quality deliverables:** This helps to determine if deliverables are being met or not.
  - **Cost tracking:** Project managers need to be accountable for the effort and cost of resources.
  - **Project performance:** Any changes made in the project due to scope-creep or other unforeseen circumstances are taken into account while measuring the overall progress of the project.

Project Closure

- The closing phase is a critical step in the project management life cycle. It signals the official end of the project and provides a period for reflection, wrap-up, and organization of materials.
- Generally, once the project is completed and delivered, the effective project managers set aside some time to identify the strengths, valuable team members are recognized, what went wrong, how it can be rectified, and what are the takeaways from the project.
- How a project manager closes the project?
  - Project performance is evaluated
    If there are elements that went really well or something didn't go as planned, it is the time to bring them up. The project manager brings out the performance reports and evaluates how well the project has performed.
  - Closing the project with a team meeting
    The final team meeting is a great way to reflect how well the project went and share the takeaways with the team members so that the future projects can be handled in a better way.

Role of Project Manager

- The role of the project manager is one of great responsibility. The project manager's job is to direct, supervise and control the project from beginning to end. Project managers should not carry out project work - managing the project is enough. Here are some of the activities a project manager undertakes:
  - The project manager must define the project, reduce it to a set of manageable tasks, obtain appropriate resources and build a team to perform the work.
  - The project manager must set the final goal of the project and motivate the project team to complete the project on time.
  - The project manager must inform all stakeholders of progress on a regular basis.
  - The project manager must assess and monitor risks to the project and mitigate them.
- No project ever goes quite as planned. Project managers must learn to adapt to and manage change.

Skills of Project Manager

- A project manager must have a range of competencies: Leadership, People management (customers, suppliers, functional managers and project team), Effective communication (verbal and written), Influencing, Negotiation, Conflict management, Planning, Contract management, Estimating, Problem solving, Creative thinking and Time management
- Project managers bear ultimate responsibility for making things happen. Traditionally, they have carried out this role as mere implementers. To do their jobs they needed to have the necessary administrative and technical competencies.
- Today they play a far broader role. In addition to the traditional skills, they need to have business skills, customer relations skills, and political skills.
- Psychologically, they must be results-oriented self-starters with a high tolerance for ambiguity because little is clear-cut in today's business environment. Shortcomings in any of these areas can lead to project failure.

Skills of Project Manager

- A Good Project Manager
  - –Takes ownership of the whole project
  - –Is proactive not reactive
  - –Adequately plans the project
  - –Is Authoritative (**NOT** Authoritarian)
  - –Is Decisive
  - –Is a Good Communicator
  - –Manages by data and facts not uniformed optimism
  - –Leads by example
  - –Has sound Judgement
  - –Is a Motivator
  - –Is Diplomatic
  - –Can Delegate

Customer Role

- A customer, often also referred to as client, can be a person or an organization that orders and buys products or services that a business offers.
- In project management, the customer is the one defining the requirements of the project and often setting the parameters such as budget and deadlines. The customer, therefore, influences the constraints of a project heavily and plays an active part in the project's process.
- Customers' role includes tasks such as:
  - approving the project plan
  - requesting changes to the project
  - approving or declining the product or service at the end of a project.
- You can distinguish between two types of customers:
  - Internal customer: from the same organization
  - External customer: belongs to another organization

Software Project Planning & Estimation

UNIT – II

Contents

- Tasks in Project Planning
- Work Breakdown Structures (WBS)
- Planning Methods
- Development Life Cycle Models
- A Generic Project Model
- Software Estimation
- COCOMO Model, Budgeting
- Scheduling techniques – PERT, CPM, Gnatt Chart

Project Planning

- Project planning is an organized and integrated management process, which focuses on activities required for successful completion of the project.
- It prevents obstacles that arise in the project such as changes in projects or organization's objectives, non-availability of resources, and so on.
- Project planning also helps in better utilization of resources and optimal usage of the allotted time for a project.
- The other objectives of project planning are listed below.
  - It defines the roles and responsibilities of the project management team members.
  - It ensures that the project management team works according to the business objectives.
  - It checks feasibility of the schedule and user requirements.
  - It determines project constraints.

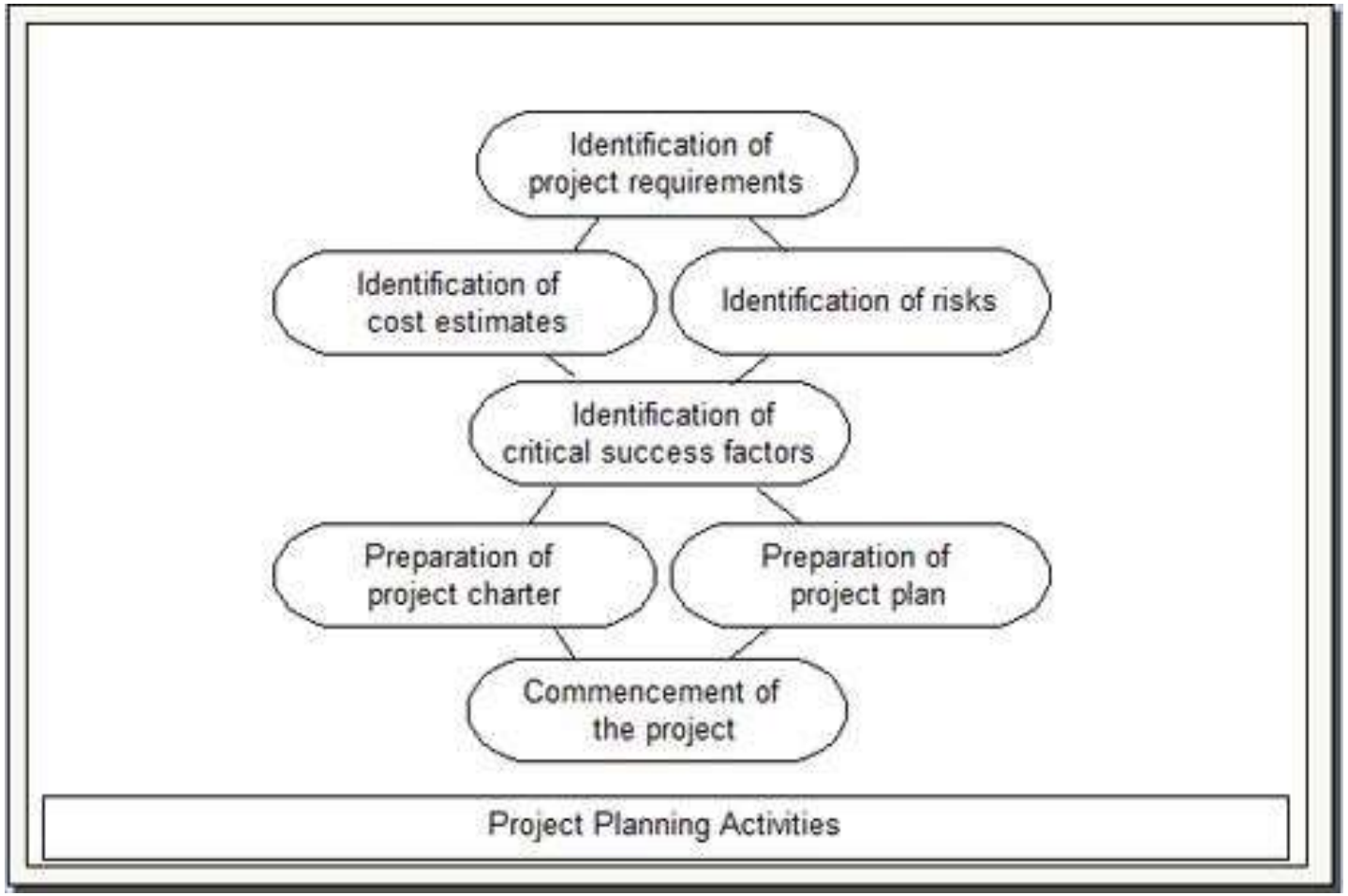
Project Planning

- For effective project planning, some principles are followed. These principles are listed below.
  - **Planning is necessary:** Planning should be done before a project begins. For effective planning, objectives and schedules should be clear and understandable.
  - **Risk analysis:** Before starting the project, senior management and the project management team should consider the risks that may affect the project. For example, the user may desire changes in requirements while the project is in progress. In such a case, the estimation of time and cost should be done according to those requirements (new requirements).
  - **Tracking of project plan:** Once the project plan is prepared, it should be tracked and modified accordingly.
  - **Meet quality standards and produce quality deliverables:** The project plan should identify processes by which the project management team can ensure quality in software. Based on the process selected for ensuring quality, the time and cost for the project is estimated.
  - **Description of flexibility to accommodate changes:** The result of project planning is recorded in the form of a project plan, which should allow new changes to be accommodated when the project is in progress.
- Project planning comprises project purpose, project scope, project planning process, and project plan. This information is essential for effective project planning and to assist project management team in accomplishing user requirements.

Project Planning Process

- The project planning process involves a set of interrelated activities followed in an orderly manner to implement user requirements in software and includes the description of a series of project planning activities and individual(s) responsible for performing these activities. In addition, the project planning process comprises the following.
  - Objectives and scope of the project
  - Techniques used to perform project planning
  - Effort (in time) of individuals involved in project
  - Project schedule and milestones
  - Resources required for the project
  - Risks associated with the project.
- Project planning process comprises several activities, which are essential for carrying out a project systematically. These activities refer to the series of tasks performed over a period of time for developing the software. These activities include estimation of time, effort, and resources required and risks associated with the project.

Project Planning Activities

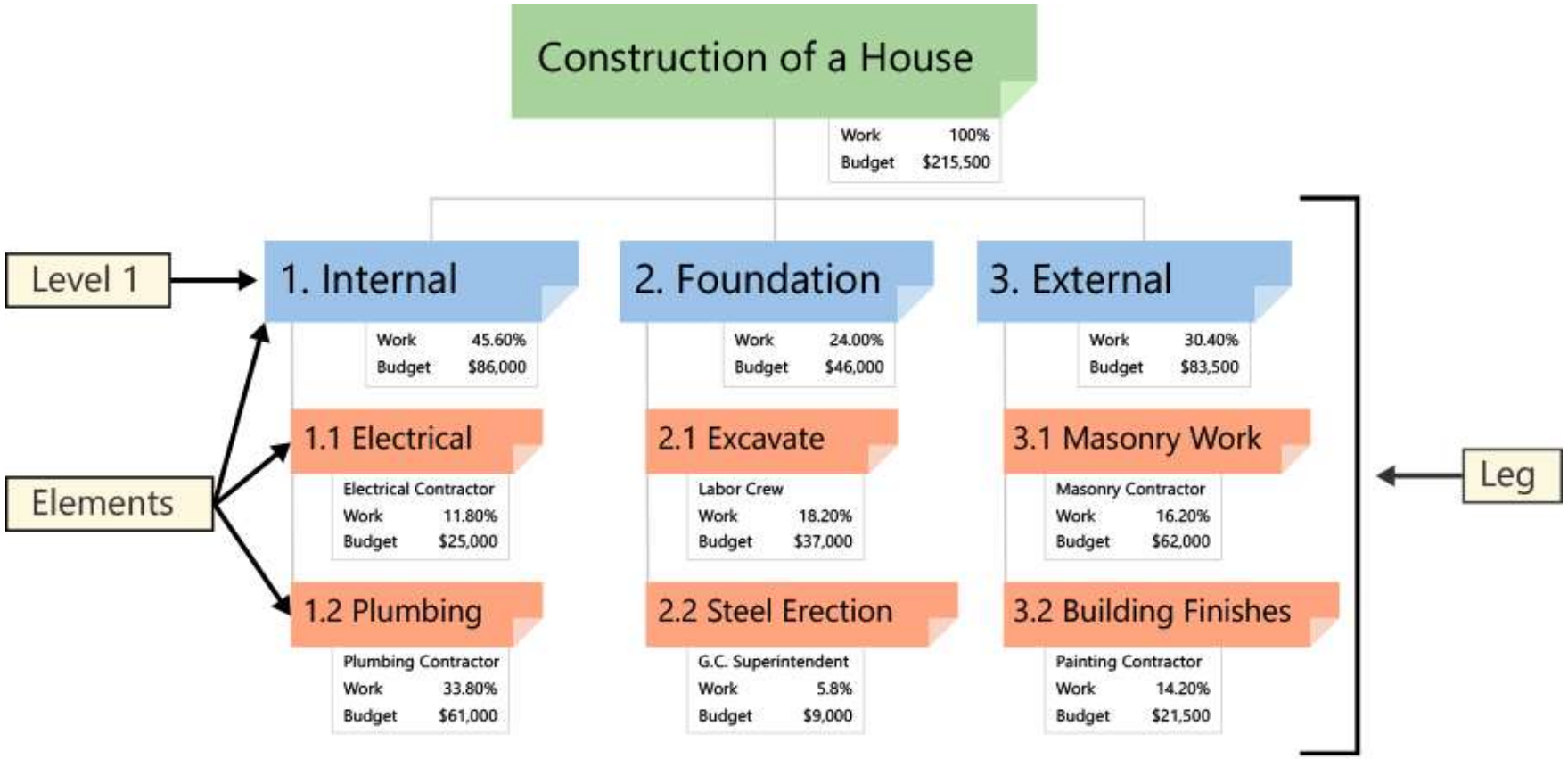


Work Breakdown Structure

- Breaking work into smaller tasks is a common productivity technique used to make the work more manageable and approachable.
- For projects, the **Work Breakdown Structure (WBS)** is the tool that utilizes this technique and is one of the most important project management documents. It singlehandedly integrates scope, cost and schedule baselines ensuring that project plans are in alignment.
- The Project Management Institute (PMI) Project Management Book of Knowledge (PMBOK) defines the Work Breakdown Structure as a "deliverable oriented hierarchical decomposition of the work to be executed by the project team."
- There are two types of WBS:
  - Deliverable-Based
  - Phase-Based.
- The most common and preferred approach is the Deliverable-Based approach. The main difference between the two approaches are the **Elements** identified in the first Level of the WBS.

Deliverable WBS

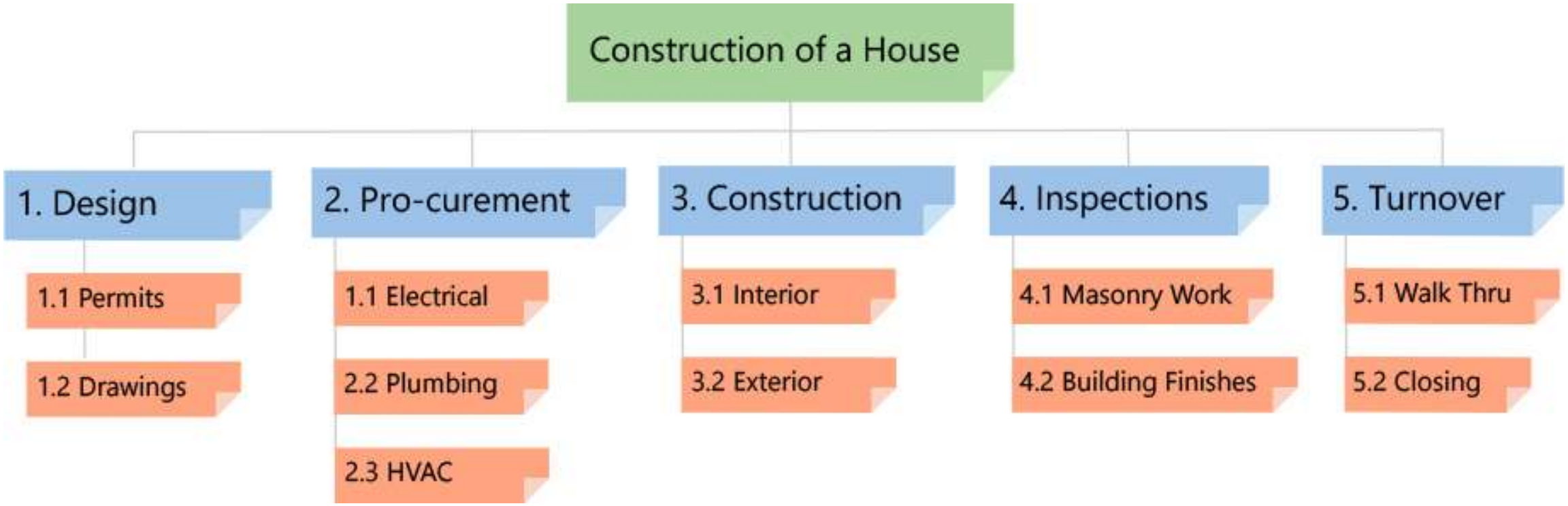
- A Deliverable-Based Work Breakdown Structure clearly demonstrates the relationship between the project deliverables (i.e., products, services or results) and the scope (i.e., work to be executed).
- Figure 1 is an example of a Deliverable-Based WBS for building a house.
- In Figure 1, the Level 1 Elements are summary deliverable descriptions. The Level 2 Elements in each Leg of the WBS are all the unique deliverables required to create the respective Level 1 deliverable.



Deliverable WBS

Phase-Based WBS

- In Figure 2, a Phase-Based WBS, the Level 1 has five Elements.
- Each of these Elements are typical phases of a project.
- The Level 2 Elements are the unique deliverables in each phase.
- Regardless of the type of WBS, the lower Level Elements are all deliverables. Notice that Elements in different Legs have the same name.
- A Phase-Based WBS requires work associated with multiple elements be divided into the work unique to each Level 1 Element.
- A WBS Dictionary is created to describe the work in each Element.



Phase-Based WBS

Quality of a WBS

- A well-designed WBS makes it easy to assign elements of the WBS to any project activity. A good WBS should exhibit the following characteristics:
  - Definable—can be described and easily understood by project participants.
  - Manageable—a meaningful unit of work where specific responsibility and authority can be assigned to a responsible individual.
  - Estimateable—duration can be estimated in time required to complete, and cost can be estimated in resources required to complete.
  - Independent—minimum interface with or dependence on other ongoing elements (i.e., assignable to a single control account, and clearly distinguishable from other work packages).
  - Integratable—integrates with other project work elements and with higher level cost estimates and schedules to include the entire project.
  - Measurable—can be used to measure progress; has start and completion dates and measurable interim milestones.
  - Adaptable—sufficiently flexible so the addition/elimination of work scope can be readily accommodated in the WBS framework.

How to make WBS?

1. **GATHER CRITICAL DOCUMENTS**
   - Gather critical project documents.
   - Identify content containing project deliverables, such as the Project Charter, Scope Statement and Project Management Plan (PMP) subsidiary plans.
2. **IDENTIFY KEY TEAM MEMBERS**
   - Identify the appropriate project team members.
   - Analyze the documents and identify the deliverables.
3. **DEFINE LEVEL 1 ELEMENTS**
   - Define the Level 1 Elements. Level 1 Elements are summary deliverable descriptions that must capture 100% of the project scope.
   - Verify 100% of scope is captured. This requirement is commonly referred to as the 100% Rule.

How to make WBS?

4. DECOMPOSE (BREAKDOWN) ELEMENTS

- Begin the process of breaking the Level 1 deliverables into unique lower Level deliverables. This "breaking down" technique is called Decomposition.
- Continue breaking down the work until the work covered in each Element is managed by a single individual or organization. Ensure that all Elements are mutually exclusive.
- Ask the question, would any additional decomposition make the project more manageable? If the answer is "no", the WBS is done.

5. CREATE WBS DICTIONARY

- Define the content of the WBS Dictionary. The WBS Dictionary is a narrative description of the work covered in each Element in the WBS. The lowest Level Elements in the WBS are called Work Packages.
- Create the WBS Dictionary descriptions at the Work Package Level with detail enough to ensure that 100% of the project scope is covered. The descriptions should include information such as, boundaries, milestones, risks, owner, costs, etc.

How to make WBS?

6. CREATE GANTT CHART SCHEDULE

- Decompose the Work Packages to activities as appropriate.
- Export or enter the Work Breakdown Structure into a Gantt chart for further scheduling and project tracking.

Software Development Life Cycle (SDLC)

- **SDLC** is a systematic process for building software that ensures the quality and correctness of the software built.
- SDLC process aims to produce high-quality software that meets customer expectations.
- SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software.
- Every phase of the SDLC life Cycle has its own process and deliverables that feed into the next phase. S
- DLC stands for **Software Development Life Cycle** and is also referred to as the Application Development life-cycle

Why SDLC?

- It offers a basis for project planning, scheduling, and estimating
- Provides a framework for a standard set of activities and deliverables
- It is a mechanism for project tracking and control
- Increases visibility of project planning to all involved stakeholders of the development process
- Increased and enhance development speed
- Improved client relations
- Helps you to decrease project risk and project management plan overhead

SDLC Phases

The entire SDLC process divided into the following stages:

Phase 1: Requirement collection and analysis

Phase 2: Feasibility study:

Phase 3: Design:

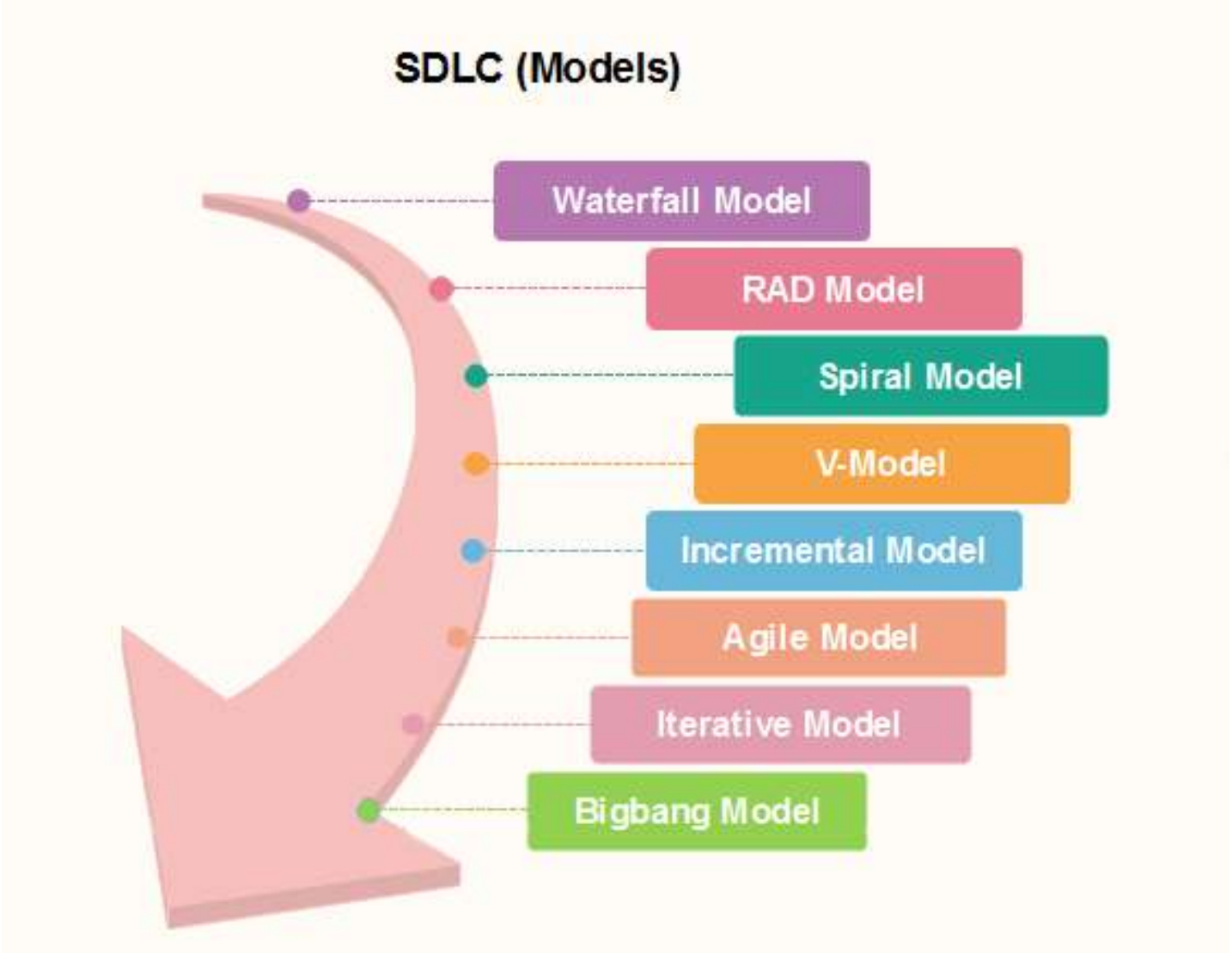
Phase 4: Coding:

Phase 5: Testing:

Phase 6: Installation/Deployment:

Phase 7: Maintenance:

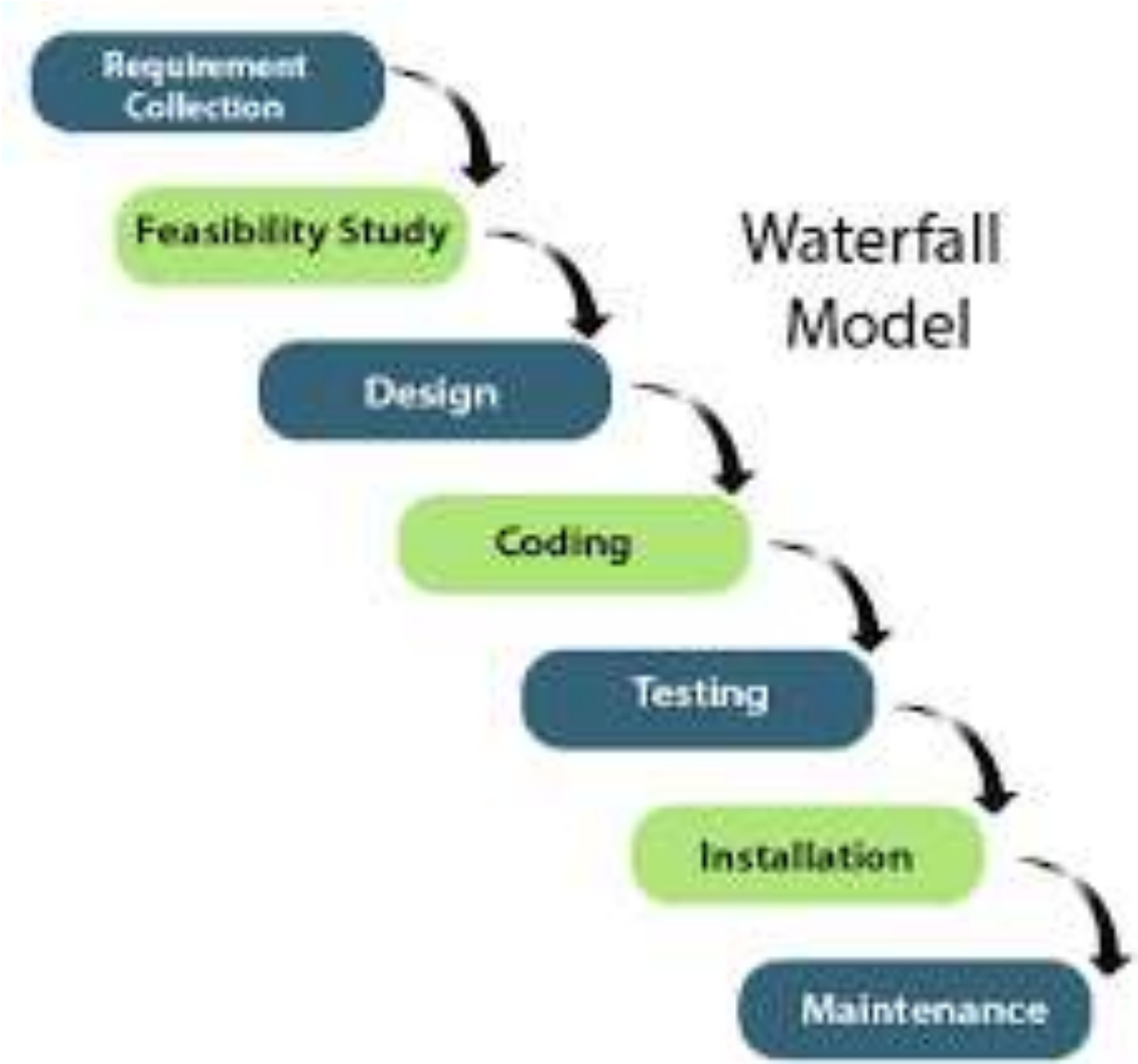
SDLC Models


SDLC (Models)
Waterfall Model
RAD Model
Spiral Model
V-Model
Incremental Model
Agile Model
Iterative Model
Bigbang Model

Waterfall

- Waterfall is the oldest and most straightforward of the structured SDLC methodologies — finish one phase, then move on to the next. No going back. Each stage relies on information from the previous stage and has its own project plan. Waterfall is easy to understand and simple to manage.
- But early delays can throw off the entire project timeline. And since there is little room for revisions once a stage is completed, problems can’t be fixed until you get to the maintenance stage.
- This model doesn’t work well if flexibility is needed or if the project is long term and ongoing.

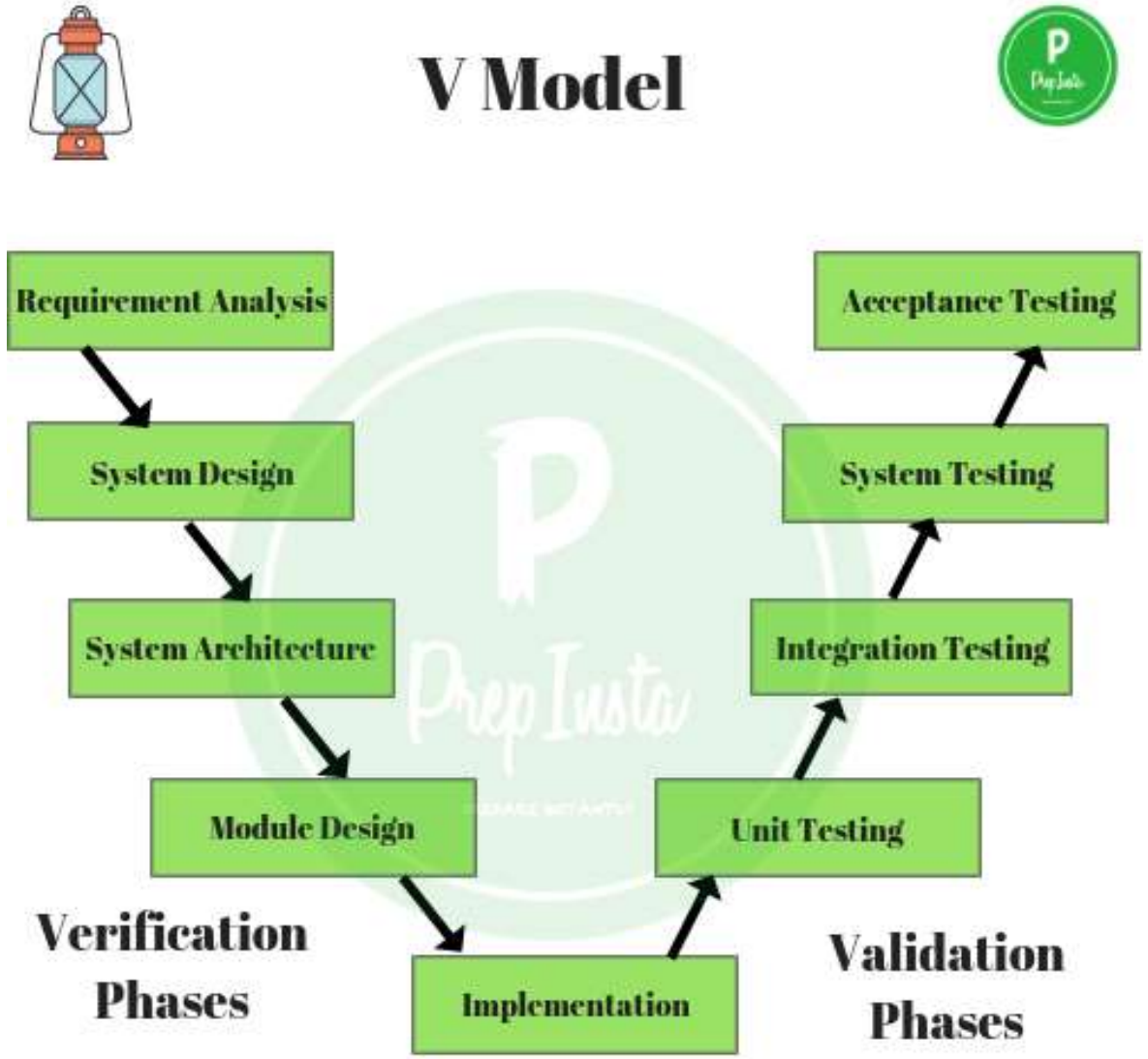
Waterfall Model



V-Shaped Model

- Also known as the Verification and Validation model, the V-shaped model grew out of Waterfall and is characterized by a corresponding testing phase for each development stage. Like Waterfall, each stage begins only after the previous one has ended.
- This model is useful when there are no unknown requirements, as it’s still difficult to go back and make changes.

V Model


V Model
Requirement Analysis
System Design
System Architecture
Module Design
Implementation
Unit Testing
Integration Testing
System Testing
Acceptance Testing
Verification Phases
Validation Phases

RAD Model

- RAD or Rapid Application Development process is an adoption of the waterfall model; it targets developing software in a short period.
- The RAD model is based on the concept that a better system can be developed in lesser time by using focus groups to gather system requirements.
  - Business Modeling
  - Data Modeling
  - Process Modeling
  - Application Generation
  - Testing and Turnover

Iterative Model

- The Iterative model is repetition incarnate. Instead of starting with fully known requirements, you implement a set of software requirements, then test, evaluate and pinpoint further requirements. A new version of the software is produced with each phase, or iteration. Rinse and repeat until the complete system is ready.
- One advantage over other SDLC methodologies: This model gives you a working version early in the process and makes it less expensive to implement changes.
- One disadvantage: Resources can quickly be eaten up by repeating the process again and again.

Incremental Model

- The incremental model is not a separate model. It is necessarily a series of waterfall cycles. The requirements are divided into groups at the start of the project.
- For each group, the SDLC model is followed to develop software. The SDLC process is repeated, with each release adding more functionality until all requirements are met.
- In this method, each cycle act as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

Big bang model

- Big bang model is focusing on all types of resources in software development and coding, with no or very little planning. The requirements are understood and implemented when they come.
- This model works best for small projects with smaller size development team which are working together.
- It is also useful for academic software development projects. It is an ideal model where requirements are either unknown or final release date is not given.

Spiral Model

- The spiral model is a **risk-driven process model. The** spiral technique is a combination of rapid prototyping and concurrency in design and development activities.
- One of the most flexible SDLC methodologies, the Spiral model takes a cue from the Iterative model and its repetition; the project passes through four phases over and over in a “spiral” until completed, allowing for multiple rounds of refinement.
- Each cycle in the spiral begins with the identification of objectives for that cycle, the different alternatives that are possible for achieving the goals, and the constraints that exist. This is the first quadrant of the cycle (upper-left quadrant).
- The next step in the cycle is to evaluate these different alternatives based on the objectives and constraints. The focus of evaluation in this step is based on the risk perception for the project.
- The next step is to develop strategies that solve uncertainties and risks. This step may involve activities such as benchmarking, simulation, and prototyping.
- This model allows for the building of a highly customized product, and user feedback can be incorporated from early on in the project. But the risk you run is creating a never-ending spiral for a project that goes on and on.

Spiral Model

Agile Model

- Agile methodology is a practice which promotes continues interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.
- Any agile software phase is characterized in a manner that addresses several key assumptions about the bulk of software projects:
  - It is difficult to think in advance which software requirements will persist and which will change. It is equally difficult to predict how user priorities will change as the project proceeds.
  - For many types of software, design and development are interleaved. That is, both activities should be performed in tandem so that design models are proven as they are created. It is difficult to think about how much design is necessary before construction is used to test the configuration.
  - Analysis, design, development, and testing are not as predictable (from a planning point of view) as we might like.

Software Project Estimation

- **Estimation** is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable.
- Estimation determines how much money, effort, resources, and time it will take to build a specific system or product. Estimation is based on −
  - Past Data/Past Experience
  - Available Documents/Knowledge
  - Assumptions
  - Identified Risks
- The four basic steps in Software Project Estimation are −
  - Estimate the size of the development product.
  - Estimate the effort in person-months or person-hours.
  - Estimate the schedule in calendar months.
  - Estimate the project cost in agreed currency.

Software Project Estimation

- Estimation need not be a one-time task in a project. It can take place during –
  - Acquiring a Project.
  - Planning the Project.
  - Execution of the Project as the need arises.
- Project scope must be understood before the estimation process begins. It will be helpful to have historical Project Data.

Project Size Estimation

- Estimation of the size of software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are:
  - Lines of Code
  - Number of entities in ER diagram
  - Total number of processes in detailed data flow diagram
  - Function points

Lines of Code (LOC)

- As the name suggest, LOC count the total number of lines of source code in a project. The units of LOC are:
  - KLOC- Thousand lines of code
  - NLOC- Non comment lines of code
  - KDSI- Thousands of delivered source instruction
- The size is estimated by comparing it with the existing systems of same kind. The experts use it to predict the required size of various components of software and then add them to get the total size.
- **Advantages:**
  - Universally accepted and is used in many models like COCOMO.
  - Estimation is closer to developer's perspective.
  - Simple to use.
- **Disadvantages:**
  - Different programming languages contains different number of lines.
  - No proper industry standard exist for this technique.
  - It is difficult to estimate the size using this technique in early stages of project.

Number of entities in ER diagram

- ER model provides a static view of the project. It describes the entities and its relationships.
- The number of entities in ER model can be used to measure the estimation of size of project. Number of entities depends on the size of the project. This is because more entities needed more classes/structures thus leading to more coding.
- **Advantages:**
  - Size estimation can be done during initial stages of planning.
  - Number of entities is independent of programming technologies used.
- **Disadvantages:**
  - No fixed standards exist. Some entities contribute more project size than others.
  - Just like FPA, it is less used in cost estimation model. Hence, it must be converted to LOC.

Total number of processes in detailed data flow diagram

- Data Flow Diagram(DFD) represents the functional view of a software. The model depicts the main processes/functions involved in software and flow of data between them.
- Utilization of number of functions in DFD to predict software size. Already existing processes of similar type are studied and used to estimate the size of the process. Sum of the estimated size of each process gives the final estimated size.
- **Advantages:**
  - It is independent of programming language.
  - Each major processes can be decomposed into smaller processes. This will increase the accuracy of estimation
- **Disadvantages:**
  - Studying similar kind of processes to estimate size takes additional time and effort.
  - All software projects are not required to construction of DFD.

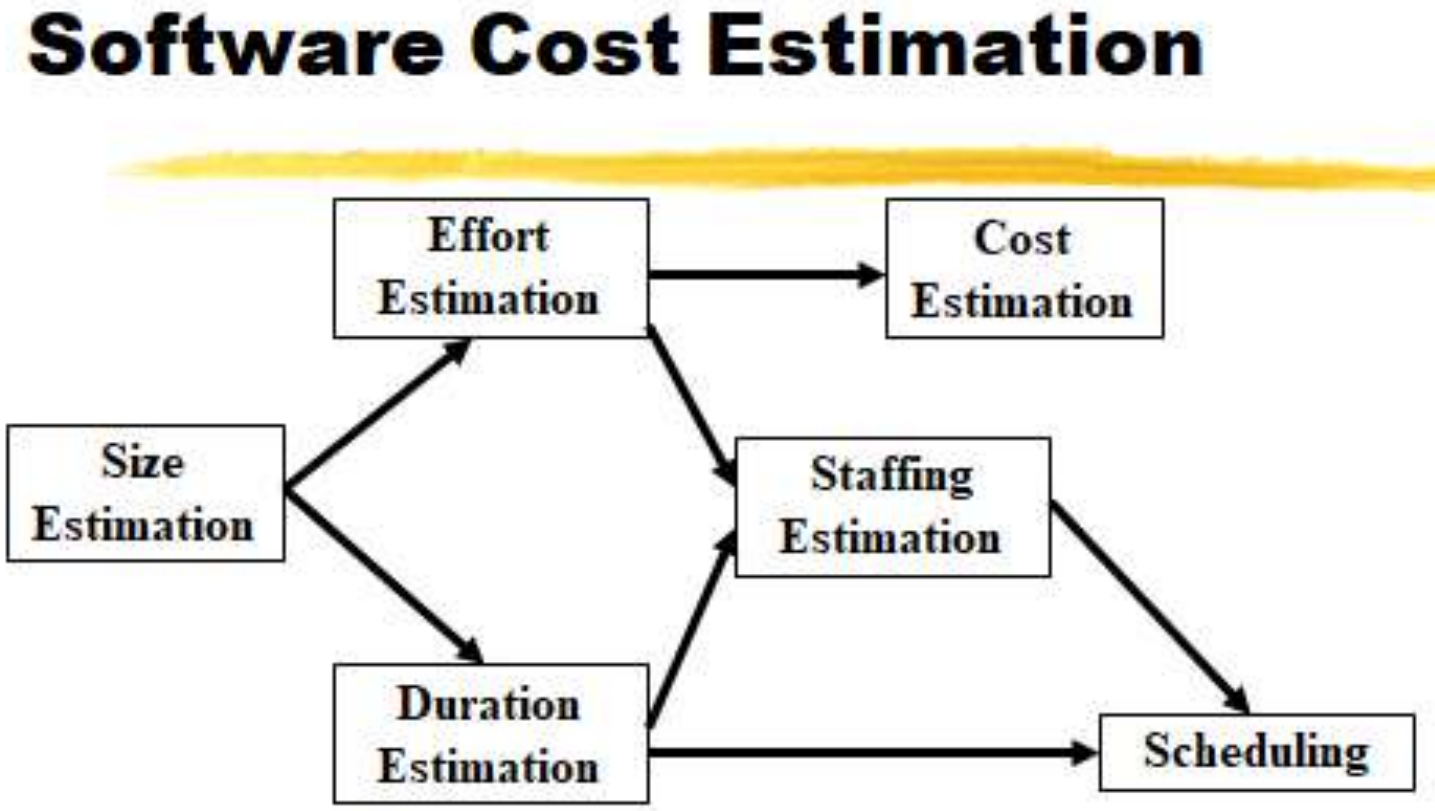
Function Point Analysis (FPA)

- Proposed by Albrecht in early 80's, in this method, the number and type of functions supported by the software are utilized to find FPC(function point count).
- **Function Point Analysis (FPA) technique** quantifies the functions contained within software in terms that are meaningful to the software users. FPs consider the number of functions being developed based on the requirements specification.
- **Function Points (FP) Counting** is governed by a standard set of rules, processes and guidelines as defined by the International Function Point Users Group (IFPUG). These are published in Counting Practices Manual (CPM).
- Overcomes the shortcomings of LoC

Function Point Analysis (FPA)

- Input:
  - A set of related inputs is counted as one input.
- Output:
  - A set of related outputs is counted as one output.
- Inquiries:
  - Each user query type is counted.
- Files:
  - Files are logically related data and thus can be data structures or physical files.
- Interface:
  - Data transfer to other systems.

Software Cost Estimation



Project Cost Estimation

- Project Cost Estimation is defined as the process of approximating the total expenditure of the project.
- The accuracy of the cost estimation depends on the accuracy and details of the project scope, which is the scope baseline. T
- he scope will also define any constraints like date, resources or budget. The risk register will help to calculate estimate types of costs, the expenses made behind the contingent action and the expenses made to cope with risks.
- The cost of the project can be estimated from various process sources
  - Creating Work Breakdown Structure (WBS)
  - Develop Schedule
  - Plan human resources
  - Identifying risks

Project Cost Estimation

- There are two key types of costs addressed by the cost estimation process:
  - Direct costs: These are the costs associated with a single area, such as a department or this particular project itself. Examples of direct costs include fixed labor, materials and equipment.
  - Indirect costs: These are costs incurred by the organization at large, such as utilities and quality control.
- Within these two categories, some typical elements that a cost estimation will take into account include:
  - Labor: the cost of project team members working on the project, both in terms of wages and time.
  - Materials and equipment: The cost of resources required for the project, from physical tools to software to legal permits.
  - Facilities: the cost of using any working spaces not owned by the organization.
  - Vendors: the cost of hiring third-party vendors or contractors.
  - Risk: the cost of any contingency plans implemented to reduce risk.

COCOMO Model

- COCOMO (COnstructive COst MOdel) proposed by Barry Boehm in 1981.
- COCOMO predicts the effort and schedule for a software project development based on inputs relating to the size of the software and a number of cost drivers that affect the productivity.
- COCOMO has three different models that reflect the complexity
  - The Basic Model
  - The Intermediate Model
  - The Detailed Model

COCOMO Model

- For each of the three product categories:
  - From size estimation (in KLOC), Boehm provides equations to predict:
    - project duration in months
    - effort in programmer-months
- Boehm obtained these equations:
  - examined historical data collected from a large number of actual projects.

The Basic COCOMO Model

- Applicable to small to medium size software projects
- Used for quick and rough estimates
- The 3 modes of software development are considered:
- Organic
  - It belongs to small & simple software projects which are handled by a small team with good domain knowledge and few rigid requirements.
  - **Example:** Small data processing or Inventory management system.

The Basic COCOMO Model

- Semidetached
  - It is an intermediate (in terms of size and complexity) project, where the team having mixed experience (both experience & inexperience resources) to deals with rigid/nonrigid requirements.
  - **Example:** Database design or OS development.
- Embedded
  - This project having a high level of complexity with a large team size by considering all sets of parameters (software, hardware and operational).
  - **Example:** Banking software or Traffic light control software.

The Basic COCOMO Model

- Gives only an approximate estimation:
  - Effort = a1 * (KLOC) * a2
  - Tdev = b1 * (Effort) * b2
  - KLOC is the estimated kilo lines of source code,
  - a1,a2,b1,b2 are constants for different categories of software products,
  - Tdev is the estimated time to develop the software in months,
  - Effort estimation is obtained in terms of person months (PMs).

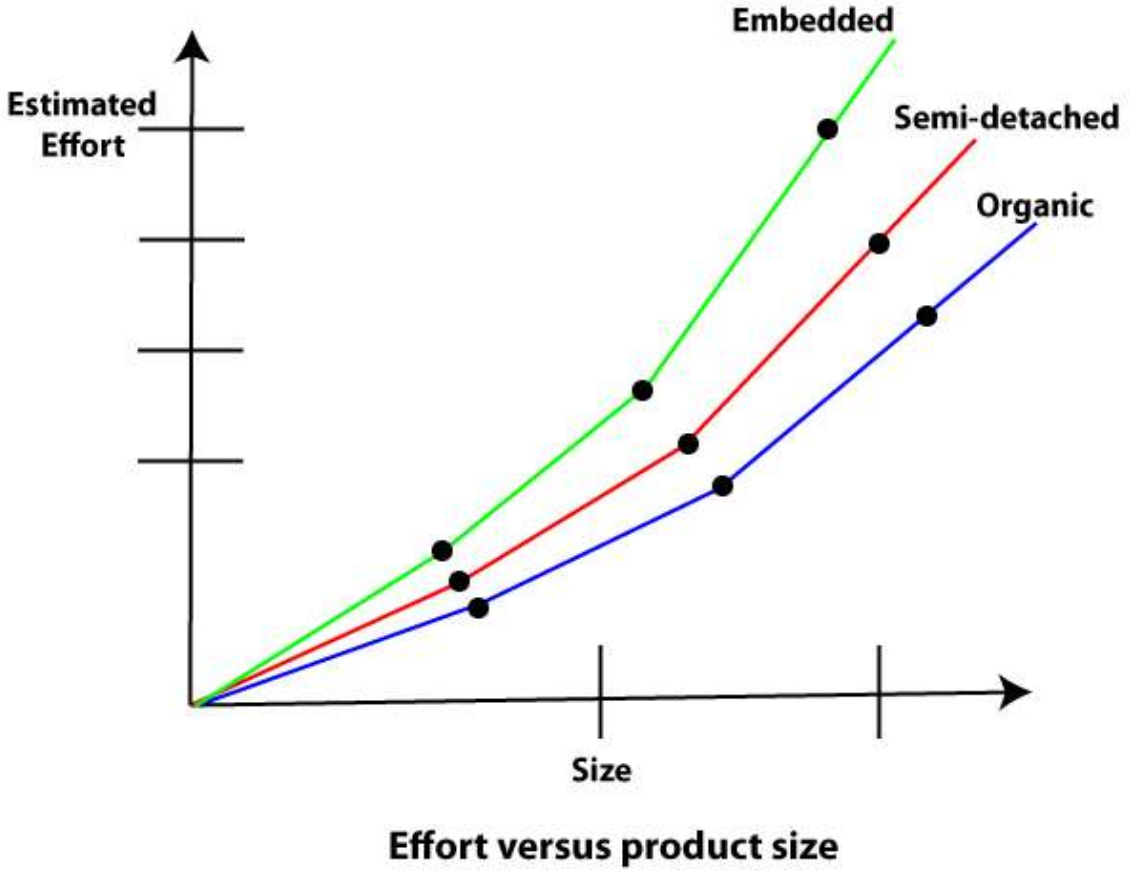
Development Time Estimation

- Organic:
  - Tdev = 2.5 * (Effort) * 0.38 Months
- Semi-detached:
  - Tdev = 2.5 * (Effort) * 0.35 Months
- Embedded:
  - Tdev = 2.5 * (Effort) * 0.32 Months

Development Effort Estimation

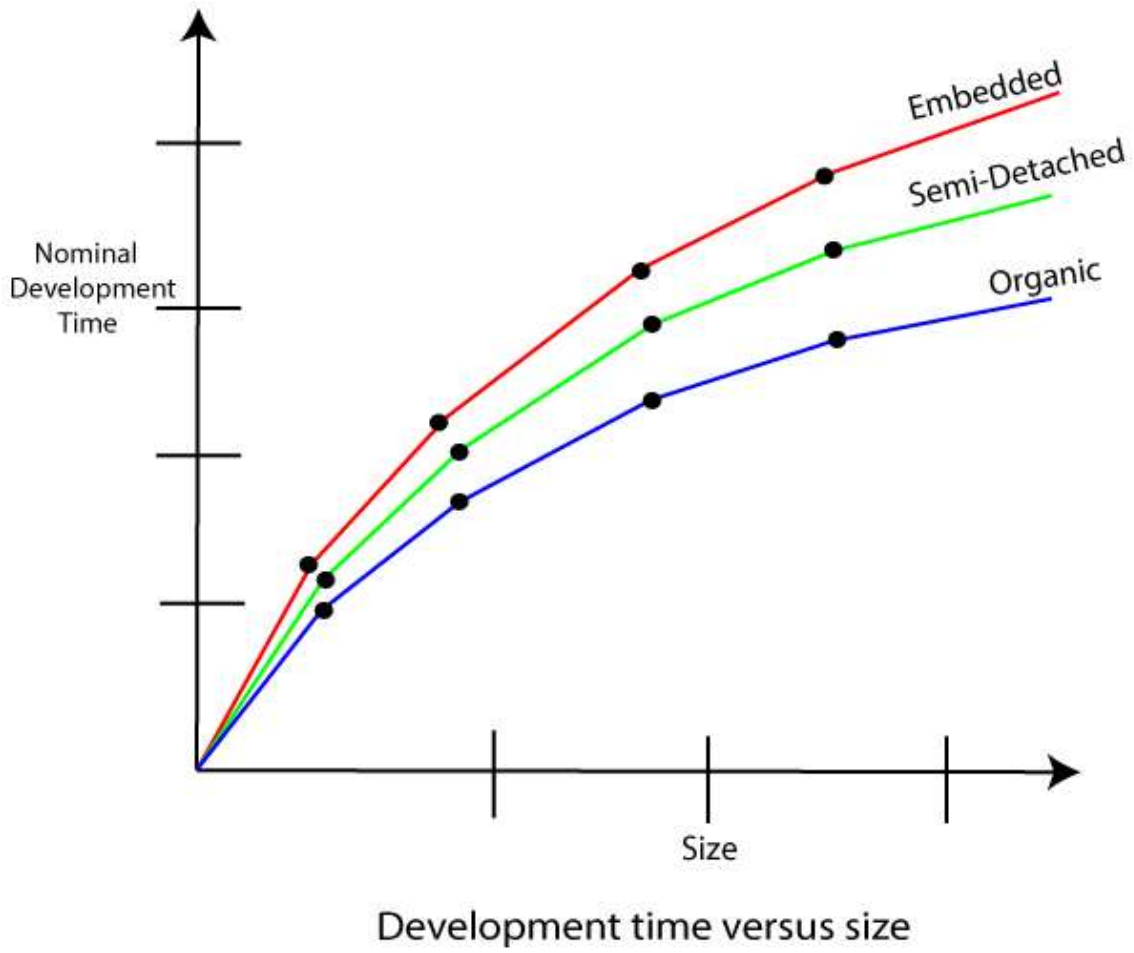
- Organic :
  - Effort = 2.4 * (KLOC) * 1.05 PM
- Semi-detached:
  - Effort = 3.0 * (KLOC) * 1.12 PM
- Embedded:
  - Effort = 3.6 * (KLOC) * 1.20PM

The Basic COCOMO Model



Effort versus product size

- Some insight into the basic COCOMO model can be obtained by plotting the estimated characteristics for different software sizes.
- Fig shows a plot of estimated effort versus product size.
- From fig, we can observe that the effort is somewhat superliner in the size of the software product.
- Thus, the effort required to develop a product increases very rapidly with project size.



Development time versus size

The Basic COCOMO Model

- Development time
  - sublinear function of product size.
- When product size increases two times,
  - development time does not double.
- Time taken:
  - almost same for all the three product categories.
- This can be explained by the fact that for larger products, a larger number of activities which can be carried out concurrently can be identified. This

The Basic COCOMO Model

- Development time does not increase linearly with product size:
  - For larger products more parallel activities can be identified:
    - can be carried out simultaneously by a number of engineers.
- Development time is roughly the same for all the three categories of products:
  - For example, a 60 KLOC program can be developed in approximately 18 months
    - regardless of whether it is of organic, semi-detached, or embedded type.
  - There is more scope for parallel activities for system and application programs,
    - than utility programs.

Example 1

- The size of an organic software product has been estimated to be 32,000 lines of source code.

- Effort = $2.4*(32)^{1.05}$ = 91 PM
- Nominal development time = $2.5*(91)^{0.38}$ = 14 months

Example 2

PROBLEM : Suppose a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the three model i.E., Organic, semi-detached & embedded.

SOLUTION

The basic COCOMO equation takes the form:

$Effort=a_1*(KLOC)\ a_2\ PM$
$Tdev=b_1*(efforts)b_2\ Months$
Estimated Size of project= 400 KLOC

(i)Organic Mode

E = 2.4 * (400) * 1.05 = 1295.31 PM
D = 2.5 * (1295.31) * 0.38=38.07 M

(ii)Semidetached Mode

E = 3.0 * (400) * 1.12=2462.79 PM
D = 2.5 * (2462.79) * 0.35=38.45 M

(iii) Embedded Mode

E = 3.6 * (400) * 1.20 = 4772.81 PM
D = 2.5 * (4772.8) * 0.32 = 38 M

Scheduling Techniques

- Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) are the two most commonly used techniques by project managers. These methods are used to calculate the time span of the project through the scope of the project.
- For tracking progress and reporting purposes, the Gantt Chart is a visualization technique used in project management.

a. Critical Path Method

Every project's tree diagram has a critical path. The Critical Path Method estimates the maximum and minimum time required to complete a project. CPM also helps to identify critical tasks that should be incorporated into a project. Delivery time changes do not affect the schedule. The scope of the project and the list of activities necessary for the completion of the project are needed for using CPM. Next, the time taken by each activity is calculated. Then, all the dependent variables are identified. This helps in identifying and separating the independent variables. Finally, it adds milestones to the project.

Scheduling Techniques

b. Program Evaluation and Review Technique (PERT)

PERT is a way to schedule the flow of tasks in a project and estimate the total time taken to complete it. This technique helps represent how each task is dependent on the other. To schedule a project using PERT, one has to define activities, arrange them in an orderly manner and define milestones. You can calculate timelines for a project on the basis of the level of confidence:

Optimistic timing

Most-likely timing

Pessimistic timing

Weighted average duration and not estimates are used by PERT to calculate different timeframes.

Scheduling Techniques

Gnatt Chart

For tracking progress and reporting purposes, the Gantt Chart is a visualization technique used in project management.

It is used by project managers most of the time to get an idea about the average time needed to finish a project.

A project schedule Gantt chart is a bar chart that represents key activities in sequence on the left vs time. Each task is represented by a bar that reflects the start and date of the activity, and therefore its duration. The chart shows all the activities, when they’re set to start and end, how long each activity will last, where there are overlaps of activities, dependencies between activities, which are connected with arrows and the start/end date of the entire project.

Scheduling Techniques

- **Pros:**
  - Adaptable to all industries and projects
  - Easy to view progress
  - Ability to set accurate deadlines and define dependencies
  - Easily modified
  - Can be created in Microsoft Excel or in a project management system
  - Ability to assign tasks to resources
- **Cons:**
  - If the Gantt chart is not part of a project management system, it is flat
  - No versioning
  - Limited collaboration
  - No progress tracking

Project Monitoring & Controlling

UNIT – 3

Contents

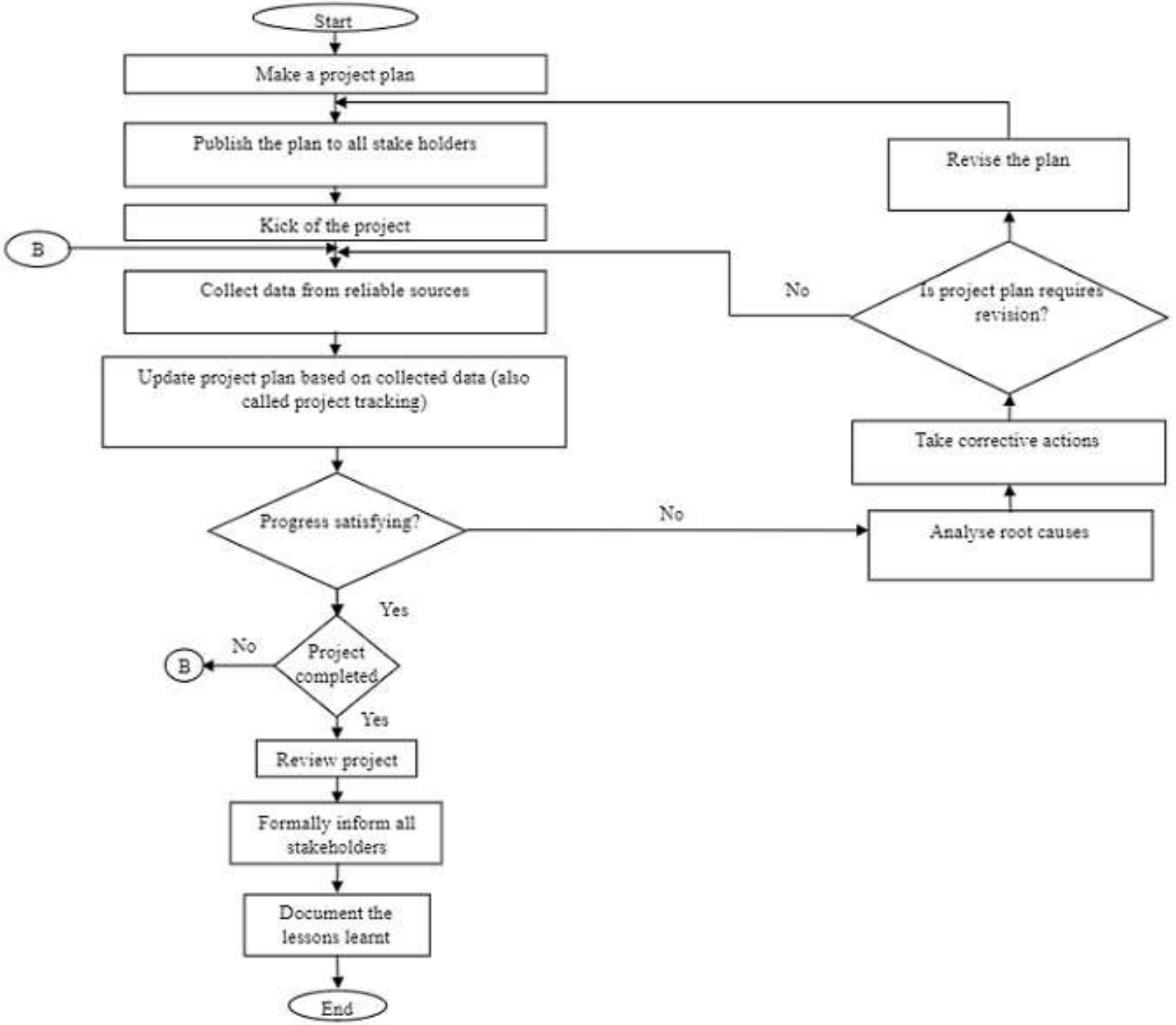
- Framework for Management and control – Collection of data – Visualizing progress – Cost monitoring
- Earned Value Analysis
- Prioritizing Monitoring
- Project tracking, Change control
- Software Configuration Management
- Contract Management.
- MS Project

Creating Framework

Project Framework consists of

1. Understanding the Project Control Cycle
2. Establishing the Project structure

Project Control Cycle



Assessing Project

Project Tracking

- Collect Data
- Cross check its validity
- Update project plan
  - If it involves time & cost

Collect Data – Reports

Project progress report	Example	Remarks
• Verbal • Formal • Regular	Periodic (weekly/ fortnightly/ monthly) meetings	Whatever is verbally told in the meeting must be recorded in the minutes of the meetings
• Verbal • Formal • Adhoc	Milestone achievements/ any completion of stage etc.	Must be followed up with written report
• Written • Formal • Regular	• Weekly/fortnightly/monthly project progress reports • Job sheets etc	Generally, these reports are sent in a predefined format
• Written • Formal • Adhoc	• Flash reports (triggered by an event) • Change report • Exception report	Project manager reviews the impact on the project due to the event.
• Verbal • Informal • Adhoc	• Corridor talk • Social interaction • Coffee break discussion	Only gives advance information to be alert and take any preventive action. Not official without written report.

Collect Data – Reports

Time Sheet

Staff John Smith **Week ending** 26/3/99

Rechargeable hours

Project	Activity code	Description	Hours this week	% Complete	Scheduled completion	Estimated completion
P21	A243	Code mod A3	12	30	24/4/99	24/4/99
P34	B771	Document take-on	20	90	1/4/99	29/3/99

Total recharged hours	32

Non-rechargeable hours

Code	Description	Hours	Comment & authorization
z99	day in lieu	8	Authorized by PB

Total non-rechargeable hours	8

Activity Assessment Sheet

Staff Justin

Ref: IoE/P/13 **Activity: Code & test module C**

Week number	13	14	15	16	17	18	
Activity Summary	G	A	A	R			
Component							*Comments*
Screen handling procedures	G	A	A	G			
File update procedures	G	G	R	A			
Housekeeping procedures	G	G	G	A			
Compilation	G	G	G	R			
Test data runs	G	G	G	A			
Program documentation	G	G	A	R			

Collect Data – Risk Reports

- "Traffic light technique' for risk reporting followed by IBM

- Step 1: Identify the first level (higher level) key elements to assess the work
- Step 2: Break down the first level key elements to second level elements
- Step 3: Judge each of the second level element's progress in 3 scales as below.

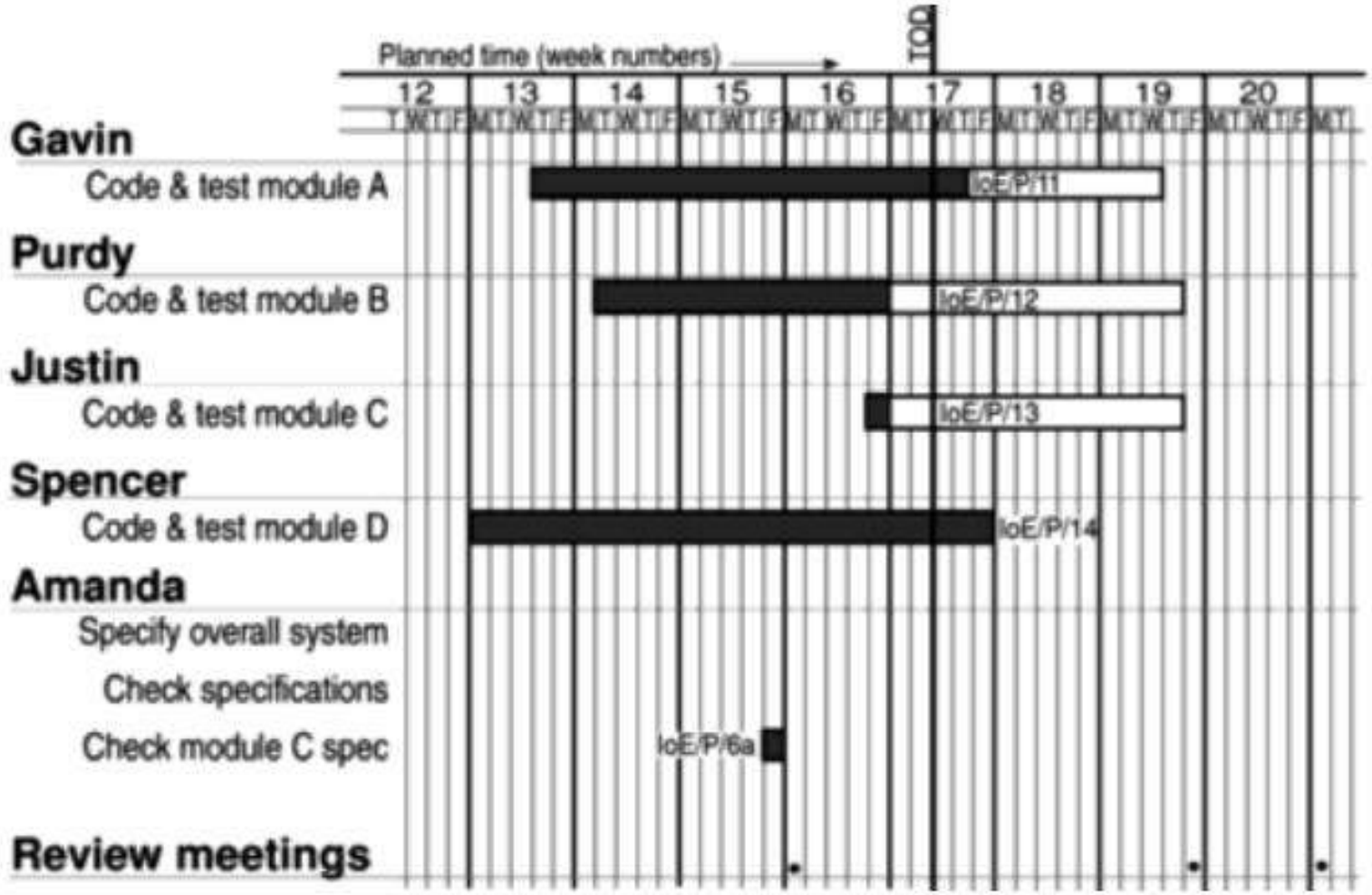
Green(G)	– As per target
Amber (A)	– Not as per target but can be brought back to control
Red (R)	– Not as per target and cannot be brought back to control without involving additional cost/resource/time

- Step 4: Based on the second level assessment, judge the first level on the same 3 point scale (Green/Amber/Red)
- Step 5: Review all the first level assessments to decide on the overall assessment of the project.

Visualizing Progress

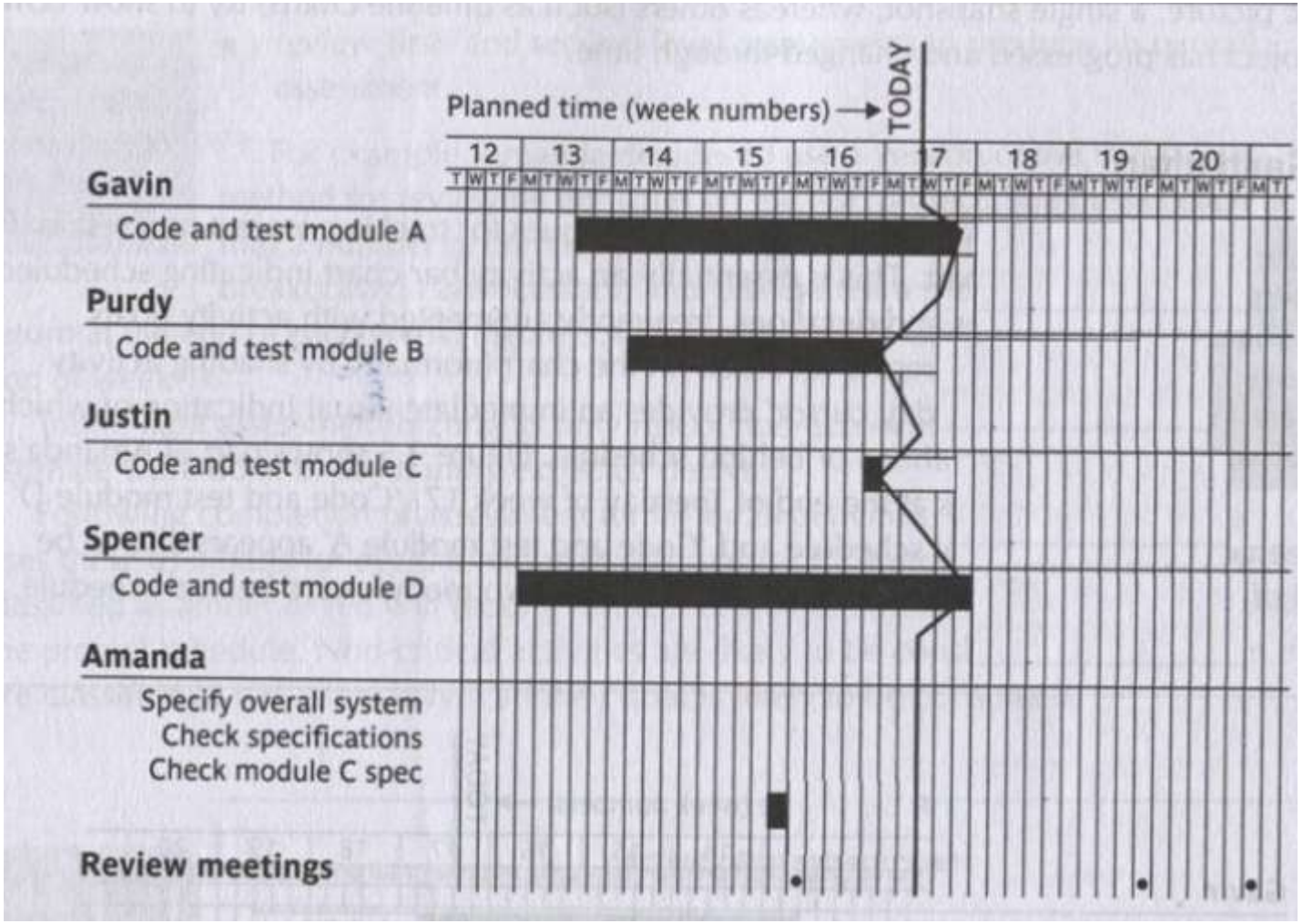
- Gantt Chart
- Slip Chart
- Ball Chart
- Timeline Graph

Gantt Chart


Planned time (week numbers)
TOD
12
13
14
15
16
17
18
19
20
Gavin
Code & test module A
IoE/P/11
Purdy
Code & test module B
IoE/P/12
Justin
Code & test module C
IoE/P/13
Spencer
Code & test module D
IoE/P/14
Amanda
Specify overall system
Check specifications
Check module C spec
IoE/P/6a
Review meetings

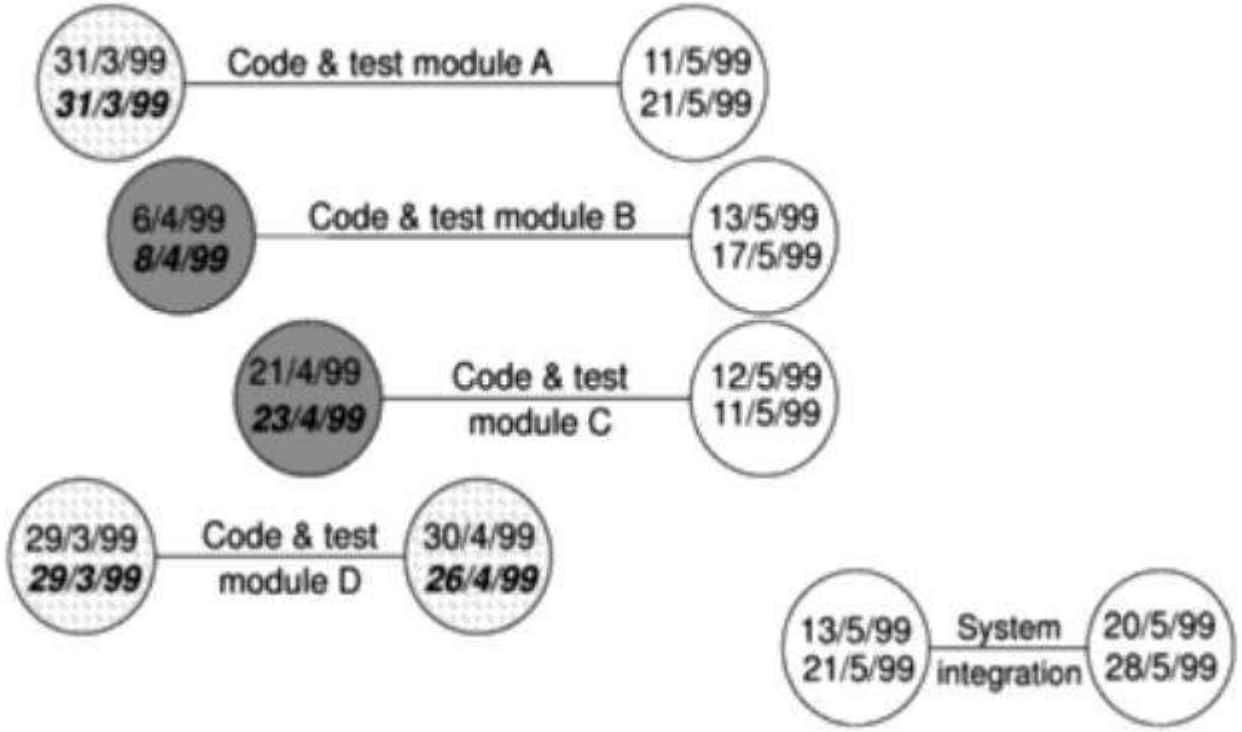
Slip Chart

Provides visual indication of activities which are not progressing in schedule

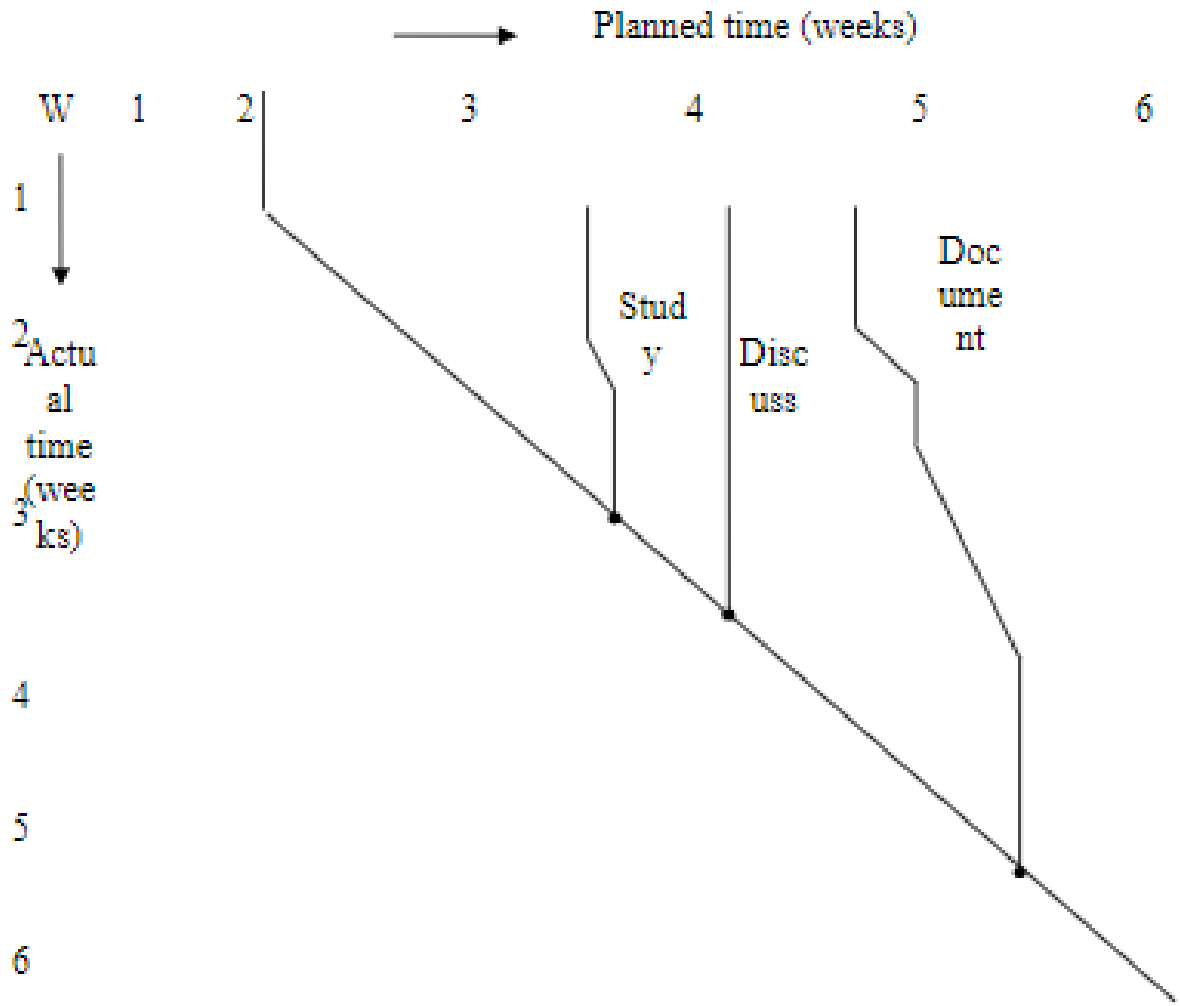


Ball Chart

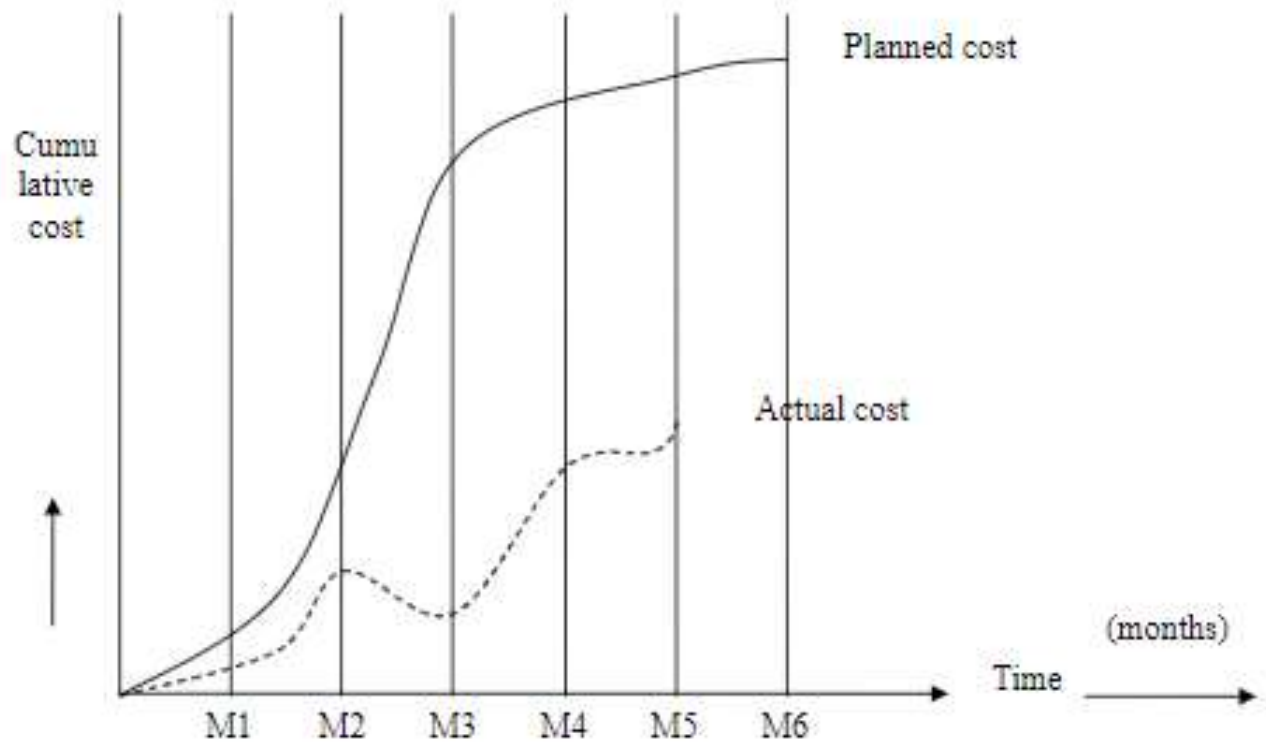
Shows whether or not targets have been met



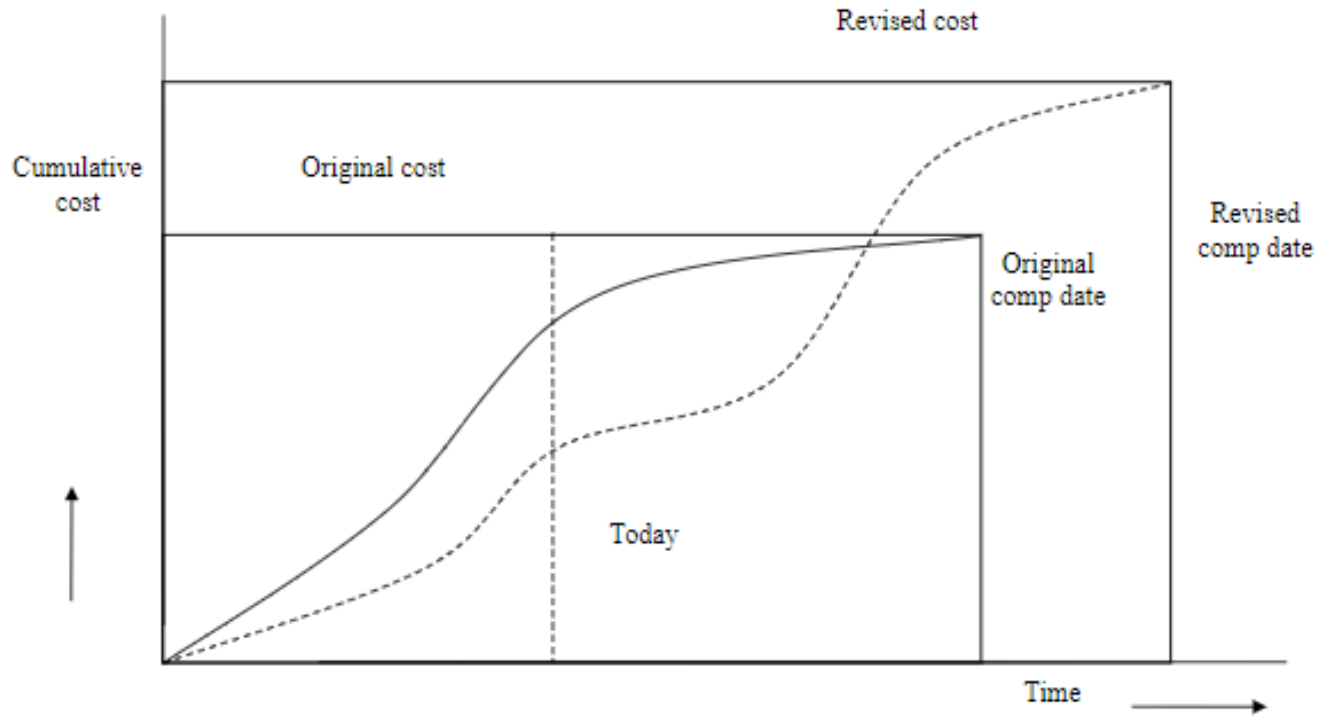
Timeline Graph



Cost Monitoring



Cost Graph with cost /time extension



Earned Value Analysis (EVA

- Earned Value Analysis (EVA) is an industry standard method of measuring a project's progress at any given point in time, forecasting its completion date and final cost, and analyzing variances in the schedule and budget as the project proceeds.
- It compares the planned amount of work with what has actually been completed, to determine if the cost, schedule, and work accomplished are progressing in accordance with the plan.
- As work is completed, it is considered "earned".
- EVA answers 2 key questions
  - At the end of the project, is it likely that the cost will be less than, equal or greater than the original estimate?
  - Will the project likely to be completed on time?

Calculating Earned Value

- Earned Value Management measures progress against a baseline. It involves calculating three key values for each activity in the WBS:
  - **The Planned Value (PV),** (formerly known as the *budgeted cost of work scheduled* or *BCWS*)—that portion of the approved cost estimate planned to be spent on the given activity during a given period.
  - **The Actual Cost (AC),** (formerly known as the *actual cost of work performed* or *ACWP*)—the total of the costs incurred in accomplishing work on the activity in a given period. This Actual Cost must correspond to whatever was budgeted for the Planned Value and the Earned Value (e.g. all labor, material, equipment, and indirect costs).
  - **The Earned Value (EV),** (formerly known as the *budget cost of work performed* or *BCWP*)—the value of the work actually completed.
- These three values are combined to determine *at that point in time* whether or not work is being accomplished as planned.

Calculating Earned Value

- The most commonly used measures are the cost variance:

  Cost Variance (CV) = EV - AC

- and the schedule variance:

  Schedule Variance (SV) = EV - PV

- These two values can be converted to efficiency indicators to reflect the cost and schedule performance of the project. The most commonly used cost-efficiency indicator is the cost performance index (CPI). It is calculated thus:

  CPI = EV / AC

- The sum of all individual EV budgets divided by the sum of all individual AC's is known as the cumulative CPI, and is generally used to forecast the cost to complete a project.

Calculating Earned Value

- The schedule performance index (SPI), calculated thus:

  SPI = EV / PV

- is often used with the CPI to forecast overall project completion estimates.
- A negative schedule variance (SV) calculated at a given point in time means the project is behind schedule, while a negative cost variance (CV) means the project is over budget.

Earned Value Management Formula

Name	Formula
Cost Variance (CV)	EV – AC
Schedule Variance (SV)	EV – PV
Time Variance (TV)	Difference between the time when the achievement of the current earned value was planned to occur and the time now
Cost Performance Index (CPI)	EV / AC
Schedule Performance Index (SPI)	EV / PV

Prioritizing monitoring

- Critical path activities
- Activities with no free float
- Activities with less than a specified float
- High risk activities
- Activities using critical resources

Getting the project back to target

- Strategies
  - Shortening the critical path
  - Altering the activity precedence requirements
- **Shorten critical path**
  - Increased required resources
  - Make available resources, work overtime to meet the requirement
  - Ensure more efficient resources (specialists and experts)
- **Reconsider precedence requirements**
  - Consider possibility of parallel activities
  - Consider training outside business hours

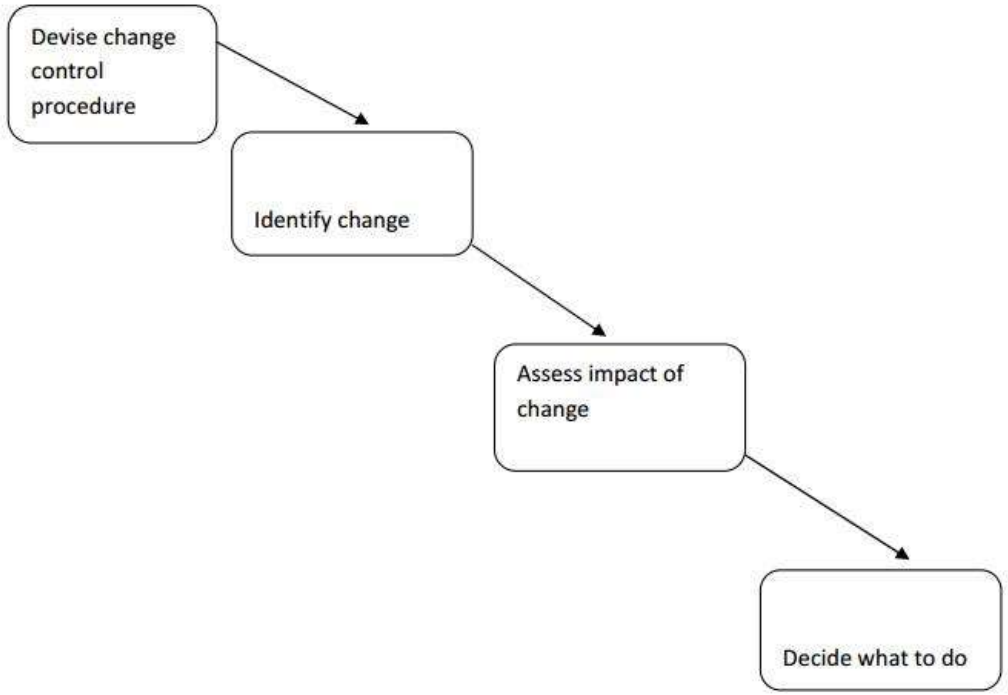
Change Control

- Change control is a methodology used to manage any change requests that impacts the your project.
- The change control process in project management ensures that each change proposed during a project is adequately defined, reviewed and approved before implementation.
- The change control process helps avoid unnecessary changes that might disrupt services and also ensures the efficient use of resources.
- That includes evaluating the request and then approving, rejected or deferring it.
- Scientific techniques to control software changes is called Software Configuration Management (SCM)
- Change control
  - Set of procedures to ensure that changes made only after a consideration of the full impacts.
- Configuration management
  - Version control to ensure that all changes are properly recorded and managed – and so that knock-on effects on other projects can be identified.

Need for Change

- Changes in business strategy
- New customer needs due to market driving forces
- Reorganization of the business
- Budgeting or scheduling constraints
- New regulations imposed by the Government
- Changes in Technology

Change Control



- Change control contains five stages:
  - Proposing a Change
  - Summary of Impact
  - Decision
  - Implementing a Change
  - Closing a Change
- There are two documents used during the process:
  - **Change Log:** used to provide a record of all changes requested and decisions made
  - **Change Request Form:** used to document details of the change, including the business case

Change Control

- Proposing a Change
  - This process gives the ability for anyone in the project team (including the customer) to suggest a change to the project. The proposal must include a description of the change and expected benefits or other reason for the change. The change is presented using the Change Request Form and added to the Change Log for the project.
- Summary of Impact
  - This process is carried out by the project manager, who will consider the overall effect on the project, covering the following items:
    - Quantifiable cost savings and benefits
    - Legal, regulatory or other unquantifiable reason for change
    - Estimated cost of the change
    - Impact on timescales
    - Extra resources needed
    - Impact on other projects and business activities
    - New risks and issues
  - After this assessment, the project manager recommends whether to carry out the change.

Change Control

- Decision
  - This process involves a review of the change request by an approved authority who will consider all the information provided by the project manager and person making the request. The decision will usually be:
    - Accept
    - Accept with comments and special conditions
    - Reject
    - Defer (change is not approved, but is left for consideration later)
- Implementing a Change
  - If the change is approved it is planned, scheduled and executed at a time agreed with the stakeholders.
  - As part of the planning, a regression test plan is needed in case the change needs to be backed out.
  - After implementation, it is usual to carry out a post-implementation review.
- Closing a Change
  - Once implemented, the requester checks and agrees on the change, and it is closed in the Change Log by the project manager.

Managing contracts

- Contract administration is the management of contracts made with customers, vendors, partners, or employees.
- The personnel involved in Contract Administration required to negotiate, support and manage effective contracts that are expensive to train and retain.
- Contract management includes negotiating the terms and conditions in contracts and ensuring compliance with the terms and conditions, as well as documenting and agreeing on any changes or amendments that may arise during its implementation or execution.
- It can be summarized as the process of systematically and efficiently managing contract creation, execution, and analysis for the purpose of maximizing financial and operational performance and minimizing risk.

Types of contract

- Fixed Price
- Time & material
- Fixed price per delivered unit

Fixed Price Contracts

- Supplier must execute all the commitments as described in the contract for a specific amount of money as decided in the contract.
- The price is fixed and cannot be altered unless the contract is renegotiated.
- Advantage:
  - Customer understanding
  - Known income/expenditure
  - Supplier efficiency
- Disadvantages
  - Difficulties in changes to environment
  - Risk absorption
  - Threat to quality

Time and Material Contract

- Supplier will charge at a fixed rate per unit of effort
  - E.g. different rates for man programmer hours, man-analyst hours etc. all already fixed
- Supplier and the acquirer will guess estimate the efforts and the time required at various levels to complete the project. These are only estimates to plan resources and activities and not the basis for final payment.
- Advantages
  - Ease of changing requirements
  - Lack of price pressure
- Disadvantages
  - No supplier role for cost effectiveness
  - Customer liability

Fixed price per delivered unit

- Prices are fixed for design, coding, implementation and support based on function points.
- Advantages
  - Customer clarity
  - Competitive nature
  - Changing requirements
  - Supplier efficiency
  - Flexibility
- Disadvantages
  - Difficulty in measurements
  - Changing requirements

Fixed price per delivered unit

FP count	Design cost/FP (Rs)	Coding cost/FP (Rs)	Implementation cost/FP (Rs)	Support cost/FP/year (Rs)
Up to 1500	10,000	7000	18,000	5000
1500 – 2000	12,000	8,000	20,000	7,000
2000 – 3000	15,000	10,000	22,000	10,000
3000 – 4000	18,000	12,000	25,000	13,000
4000 – 5000	20,000	15,000	30,000	15,000

Contract Management

- Responsibility
  - Supplier
    - Work execution
  - Acquirer
    - Managing and ensuring that the project is on right track
- Approvals
  - Every milestone
- Change Control
  - Use of SCM
- Sync of both parties
  - Periodic meetings and signed minutes
  - Periodic reports and acknowledgements

MS Project

- Microsoft **Project** (MSP) is a **project** management software made for **project** managers so they can control their projects.
- Depending on your plan, Microsoft **Project** lets you plan projects, assign tasks, manage resources, make reports and more.

Thank You!

RISK MANAGEMENT

UNIT – IV

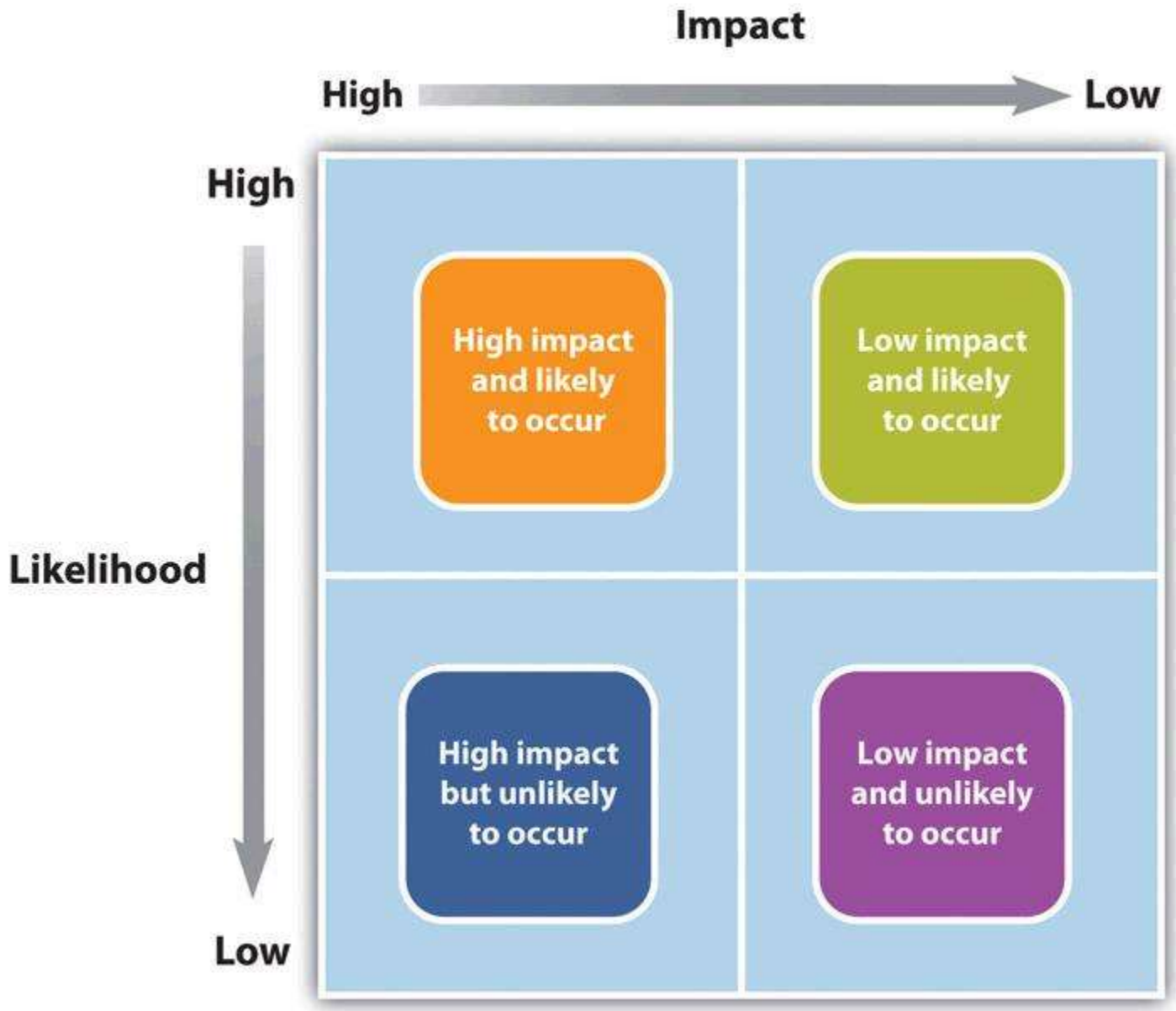
Contents

- Risk management planning
- Risk Management Activities
- Common sources of risk on information technology projects
- Risk identification
- Qualitative risk analysis & Quantitative risk analysis
- Risk response planning
- Risk monitoring and control

Risk

- A risk is any uncertain event or condition that might affect your project.
- Risk is a function of two components:
  - The ***Probability*** of occurrence
  - The ***Consequences*** (what's at stake)
- Risk is inevitable in a business organization when undertaking projects. However, the project manager needs to ensure that risks are kept to a minimal.
- Risks can be mainly divided between two types, negative impact risk and positive impact risk.
- Some events (like finding an easier way to do an activity) or conditions (like lower prices for certain materials) can help your project. When this happens, we call it an opportunity; but it's still handled just like a risk.
- Once the risk has been identified, project managers need to come up with a mitigation plan or any other solution to counter attack the risk.

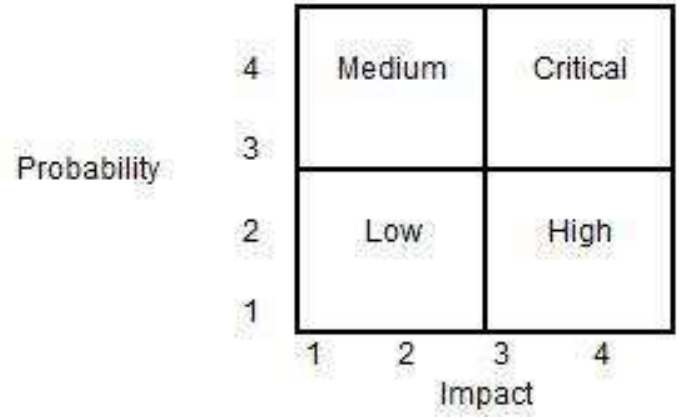
Risk and Impact



Risk Quantification

Risks can be evaluated based on quantity. Project managers need to analyze the likely chances of a risk occurring with the help of a matrix.

Using the matrix, the project manager can categorize the risk into four categories as Low, Medium, High and Critical. The probability of occurrence and the impact on the project are the two parameters used for placing the risk in the matrix categories. As an example, if a risk occurrence is low (probability = 2) and it has the highest impact (impact = 4), the risk can be categorized as 'High'.



Risk Management

- Project risk management is the process of identifying, analyzing and responding to any risk that arises over the life cycle of a project to help the project remain on track and meet its goal.
- Risk management isn't reactive only; it should be part of the planning process to figure out risk that might happen in the project and how to control that risk if it in fact occurs.
- A risk is anything that could potentially impact your project's timeline, performance or budget. Risks are potentialities, and in a project management context, if they become realities, they then become classified as "issues" that must be addressed. So risk management, then, is the process of identifying, categorizing, prioritizing and planning for risks before they become issues.
- On large-scale projects, risk management strategies might include extensive detailed planning for each risk to ensure mitigation strategies are in place if issues arise. For smaller projects, risk management might mean a simple, prioritized list of high, medium and low priority risks.

The Risk Planning Process

- ***Risk planning*** is the process of identifying, prioritizing, and managing risk.
- Every project or initiative has ***objectives***, that is, goals that it seeks to accomplish. These are often called Critical Success Factors
- ***Risk events*** threaten the successful completion of these critical success factors. Thus, risk planning involves identifying the most important risk events in advance, prioritizing them, and developing the appropriate risk response plans. There are three steps to risk planning:
  - Identifying Risks
  - Prioritizing Risks
  - Determining Response Plans

Risk Planning: Identifying Risks

- A strong risk identification process is important to the successful completion of the critical success factors.
- A Risk Management Plan is prepared which includes items such as:
  - Risk Register
  - Risk Breakdown Structure
  - Risk Analysis
- The risk register is the itemized listing of most important risks and it becomes the cornerstone of the Risk Management Plan. It requires careful consideration of the project risks and what could affect the project's critical success factors.
- Here are a few ideas to ensure that each risk is identified: Use a Risk Breakdown Structure, develop a checklist, look at assumptions, previous project experience, expert judgment.
- Obviously, it is not possible to list all project risks. Although you should endeavor to identify the most important ones, you cannot predict everything and your stakeholders do not expect you to.

Risk Planning: Prioritizing Risks

- To create a strong risk management plan, risks must be analyzed and prioritized to determine which require the project manager's time and attention, how often, and what resources are required.
- Prioritizing risks ensures that stakeholders recognize the importance placed on their areas of concern which goes a long ways toward placating them.
- Since risk has two components:

  Risk = Probability x Impact

- Each of these factors should be prioritized. The scale is not important, but it is often 1-10, low-medium-high, or a similar scale.
- If your risk register is a table with the risks listed vertically (in rows), you would add two columns labelled probability and impact. Each risk gets a ranking of 1-10, or whatever scale you choose.
- After the initial ranking, an overall prioritization is often helpful to stakeholders. You would multiply the probability and impact to get a risk level, and then sort the table from highest to lowest. Clearly you will be able to see which risks to focus your attention on.

Sample Risk Register

Risk	Probability	Impact	Priority	Trigger	Response Plan
Miss completion date due to inclement weather	3	6	18	Site foreman to decide	See below
Deficient materials arrive at site	1	8	8	Site foreman inspects material upon arrival and decides	Send material back for full refund

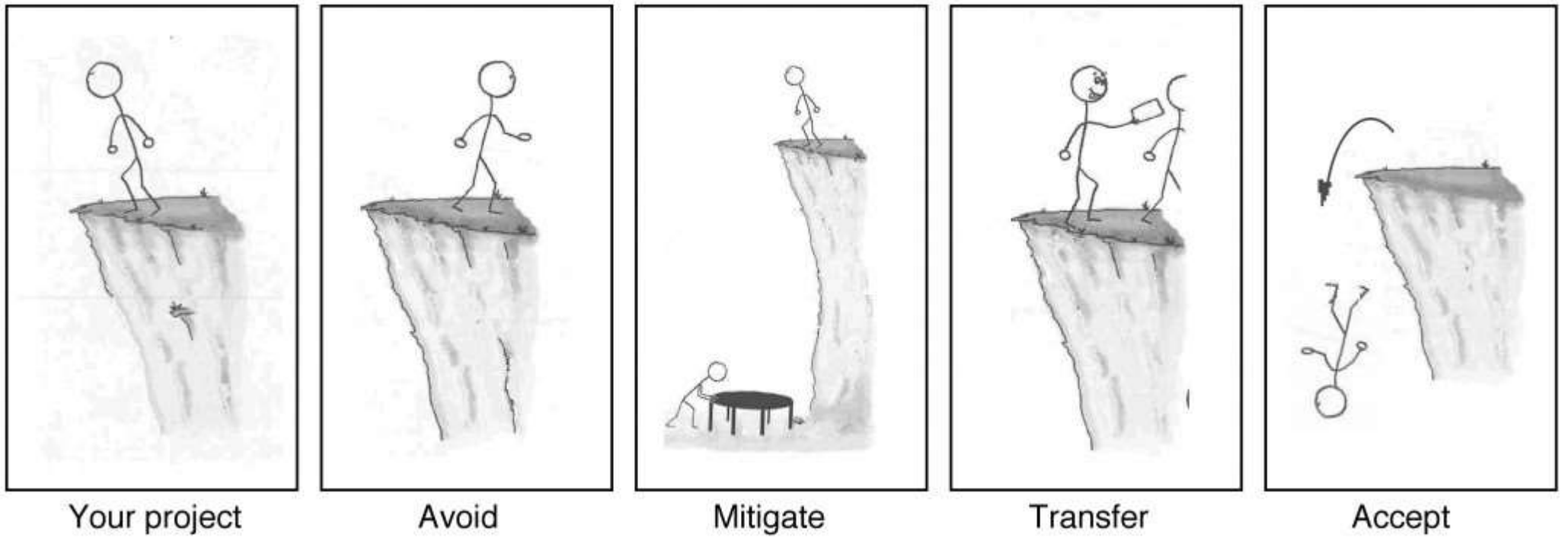
Risk Planning: Determine Risk Response Plans

- The final piece of information that completes the risk register is a risk response plan.
- Now that you've identified the triggers that allow you to quickly identify when a risk has occurred (or is occurring), the response plan gives you a head start in the response.
- Some responses occur at the beginning of the project (when the risk planning process is taking place) and others occur when the risk event occurs.
- They must contain an appropriate level of detail. For major risks a good action plan is necessary in advance and could warrant its own write up. For medium risks a small action plan could be placed within the risk register, and for small risks there could be no action plan at all.

Risk Planning: Determine Risk Response Plans

- There are four possible responses to risk events:
  - **Avoid.** Eliminate the threat. For example, change the scope of the project, spin off a certain business unit, or change the objectives that the risk event is threatening.
  - **Transfer.** Off-load the risk to a third party. For example, buy insurance, issue a performance bond, or change the contract from a lump sum to a unit price (or vice versa).
  - **Mitigate.** Reduce the probability or impact of the risk event. For example, cover the project area to prevent work stoppages due to inclement weather, or purchase materials in advance to ensure they can be returned without threatening the project completion date.
  - **Accept.** Sometimes there is no other alternative than to proceed with the project and accept the risk. But producing documentation, holding meetings, and communicating the risk with stakeholders can go a long ways toward minimizing the damage.

Risk Planning: Determine Risk Response Plans



Risk Management Process

- The risk management process involves five components.

The five components are:

1. Planning for risk
2. Identifying risks
3. Analyzing risks
4. Developing risk response strategies
5. Monitoring and controlling risks

Good risk management is proactive, not reactive, and seeks to reduce the probability of an adverse event occurring as well as the magnitude of its impact.

Risk Planning

- The project manager should develop a written risk management strategy which includes the methods used to execute a project's risk management plan.
- Adequate resources need to be available to manage risk.
- The key to writing a good Risk Management Plan is to provide the necessary information so the project team knows the objectives, goals, tools and techniques, reporting, documentation, and communication roles and responsibilities.

Identifying Risks

- Project risks should be examined to a level of detail that permits an evaluator to understand the significance of the risk and its root causes and to potentially examine those root causes. Surveys of customers, end users, and other stakeholders could be beneficial.
- Some typical risk categories are:
  - **Cost** – the cost of the project is higher than forecast, or increases during the project (scope creep)
  - **Schedule** – customers or end users are not given the final product within the agreed upon time frame
  - **Technical** – performance objectives are not met
  - **Feasibility** – the product is does not turn out to meet financial and/or business objectives.
  - **Logistics** – components do not arrive in time
  - **Human Resources** – project staff are not available, or lose availability
  - **Production** – concerns over packaging, manufacturing
  - **Support** – maintainability, operability, and trainability
  - **Engineering** – technical requirements for the product are too onerous, or not physically possible

Identifying Risks

- **Business** – the financial metrics of the project change (demand slows, market prices change, etc.)
- **Contract** – third party consultants/contractors/suppliers do not perform as anticipated, or did not interpret the contract the same way
- **Funding** – the project cannot be funded to completion, or funding is removed part-way
- **Management** – meddling in the project causes complications
- **Political** – regulations change, or were not fully considered
- **Threat** – Security, survivability, and vulnerability
- **Test** – product tests are not set up correctly

- Cost and Schedule risk are often the most likely to happen and require the most management resources.

Analyzing Risks

- Risk analysis is the systematic process to estimate the level of risk for identified and approved risks.
- Normally, this involves the creation of a risk matrix which quantifies the probability and consequence of the defined risks and a conversion to an overall risk level.
- **Qualitative Analysis**
  - A commonly used qualitative risk analysis method involves risk scales for estimating probability of occurrence and a risk mapping matrix.
  - For each identified risk a probability and a consequence is assigned in the form of letters A to E. Each letter should be defined by a verbal description.
  - Then a risk mapping matrix is drawn up to categorize each risk.

Analyzing Risks

- **Quantitative Analysis**
  - Two primary methods exist in order to perform a quantitative risk analysis:
    - Decision Tree Analysis
    - Monte Carlo Analysis
  - In a decision tree, the various outcomes are analyzed according to probability to come up with overall probabilities of all of the possible permutations.
  - The Monte Carlo process is an attempt to create probability distributions for potential risks and randomly sample them to quantify the risk.

Developing Response Startegies

- In the Risk Management Plan (within the Project Management Plan) strategies to deal with each risk should be developed.
- There are four ways to handle risk: Avoid, Transfer, Metigate, Accept

Monitoring and Controlling Risks

- Within the Risk Management Plan, provisions should be in place to systematically track and evaluate the effectiveness of the risk response actions against established metrics.
- Some techniques that can be used for monitoring and controlling risk:
  - **Earned Value:** This method compares the value of work completed to date (earned) with the value of work supposed to be performed at that point in the schedule. It is a technique to manage budget and schedule risk.
  - **Program Metrics:** Formal, periodic performance assessments evaluating whether the risk management plan is achieving its objectives.
  - **Technical Performance Measurement (TPM):** A way to measure the technical performance of a project and compare with the specifications required for project success.

Thank You!

Configuration Management

UNIT – V

Contents

- Software Configuration Management (SCM)
- Software Configuration Items (SCI)
- SCM Process & Goals of SCM
- Identification of Objects in the Software Configuration
- Configuration management & Maintenance plan
- Change Management Version and Release Management
- Configuration Management Tools

SCM

- **Software Configuration Management(SCM)** is a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle.
- When we develop software, the product (software) undergoes many changes in their maintenance phase; we need to handle these changes effectively.
- Several individuals (programs) works together to achieve these common goals. This individual produces several work product (SC Items) e.g., Intermediate version of modules or test data used during debugging, parts of the final product.
- The elements that comprise all information produced as a part of the software process are collectively called a software configuration.
- As software development progresses, the number of Software Configuration elements (SCI's) grow rapidly. These are handled and controlled by SCM. This is where we require software configuration management.

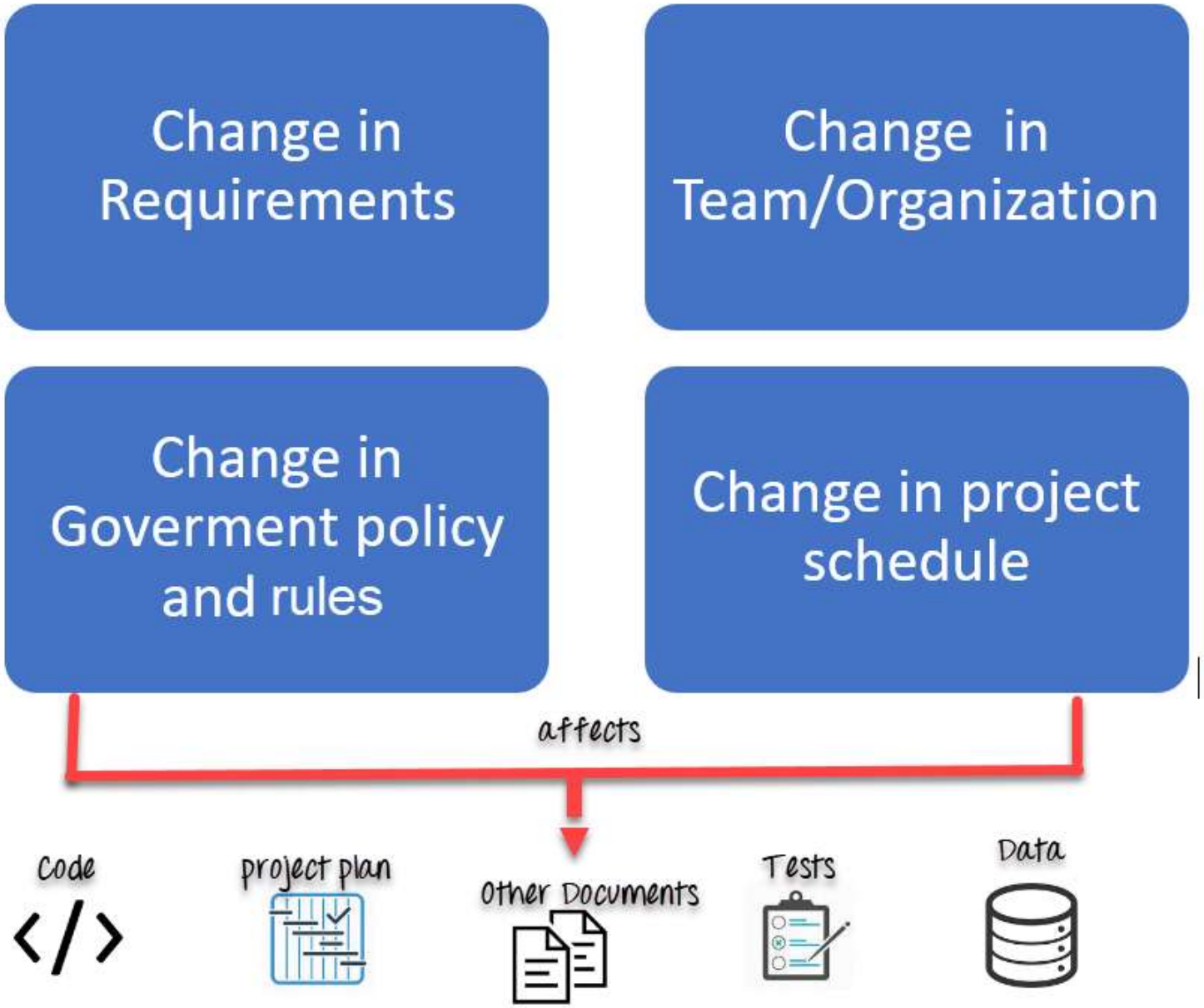
SCM

- A configuration of the product refers not only to the product's constituent but also to a particular version of the component.
- Therefore, SCM is the discipline which
  - Identify change
  - Monitor and control change
  - Ensure the proper implementation of change made to the item.
  - Auditing and reporting on the change made.
- Configuration Management (CM) is a technic of identifying, organizing, and controlling modification to software being built by a programming team.
- **The objective is to maximize productivity by minimizing mistakes (errors).**

Need for SCM

- There are multiple people working on software which is continually updating
- It may be a case where multiple version, branches, authors are involved in a software config project, and the team is geographically distributed and works concurrently
- Changes in user requirement, policy, budget, schedule need to be accommodated.
- Software should able to run on various machines and Operating Systems
- Helps to develop coordination among stakeholders
- SCM process is also beneficial to control the costs involved in making changes to a system
- Any change in the software configuration Items will affect the final product. Therefore, changes to configuration items need to be controlled and managed.

Need for SCM


Change in Requirements
Change in Team/Organization
Change in Goverment policy and rules
Change in project schedule
affects
Code
project plan
Other Documents
Tests
Data

Importance of SCM

- It is practical in controlling and managing the access to various SCIs e.g., by preventing the two members of a team for checking out the same component for modification at the same time.
- **It provides the tool to ensure that changes are being properly implemented.**
- It has the capability of describing and storing the various constituent of software.
- SCM is used in keeping a system in a consistent state by automatically producing derived version upon modification of the same component.

Tasks in SCM process

- Configuration Identification
- Baselines
- Change Control
- Configuration Status Accounting
- Configuration Audits and Reviews

Configuration Identification:

- Configuration identification is a method of determining the scope of the software system. With the help of this step, you can manage or control something even if you don't know what it is. It is a description that contains the CSCI type (Computer Software Configuration Item), a project identifier and version information.
- **Activities during this process:**
  - Identification of configuration Items like source code modules, test case, and requirements specification.
  - Identification of each CSCI in the SCM repository, by using an object-oriented approach
  - The process starts with basic objects which are grouped into aggregate objects. Details of what, why, when and by whom changes in the test are made
  - Every object has its own features that identify its name that is explicit to all other objects
  - List of resources required such as the document, the file, tools, etc.
- Example: Instead of naming a File login.php its should be named login_v1.2.php where v1.2 stands for the version number of the file. Or instead of naming folder "Code" it should be named "Code_D" where D represents code should be backed up daily.

Baseline

- A baseline is a formally accepted version of a software configuration item. It is designated and fixed at a specific time while conducting the SCM process. It can only be changed through formal change control procedures.
- **Activities during this process:**
  - Facilitate construction of various versions of an application
  - Defining and determining mechanisms for managing various versions of these work products
  - The functional baseline corresponds to the reviewed system requirements
  - Widely used baselines include functional, developmental, and product baselines
- In simple words, baseline means ready for release.

Change Control

- Change control is a procedural method which ensures quality and consistency when changes are made in the configuration object. In this step, the change request is submitted to software configuration manager.
- Activities during this process:
  - Control ad-hoc change to build stable software development environment. Changes are committed to the repository
  - The request will be checked based on the technical merit, possible side effects and overall impact on other configuration objects.
  - It manages changes and making configuration items available during the software lifecycle

Configuration Status Accounting

- Configuration status accounting tracks each release during the SCM process. This stage involves tracking what each version has and the changes that lead to this version.
- **Activities during this process:**
  - Keeps a record of all the changes made to the previous baseline to reach a new baseline
  - Identify all items to define the software configuration
  - Monitor status of change requests
  - Complete listing of all changes since the last baseline
  - Allows tracking of progress to next baseline
  - Allows to check previous releases/versions to be extracted for testing

Configuration Audits and Reviews

- Software Configuration audits verify that all the software product satisfies the baseline needs. It ensures that what is built is what is delivered.
- **Activities during this process:**
  - Configuration auditing is conducted by auditors by checking that defined processes are being followed and ensuring that the SCM goals are satisfied.
  - To verify compliance with configuration control standards. auditing and reporting the changes made
  - SCM audits also ensure that traceability is maintained during the process.
  - Ensures that changes made to a baseline comply with the configuration status reports
  - Validation of completeness and consistency

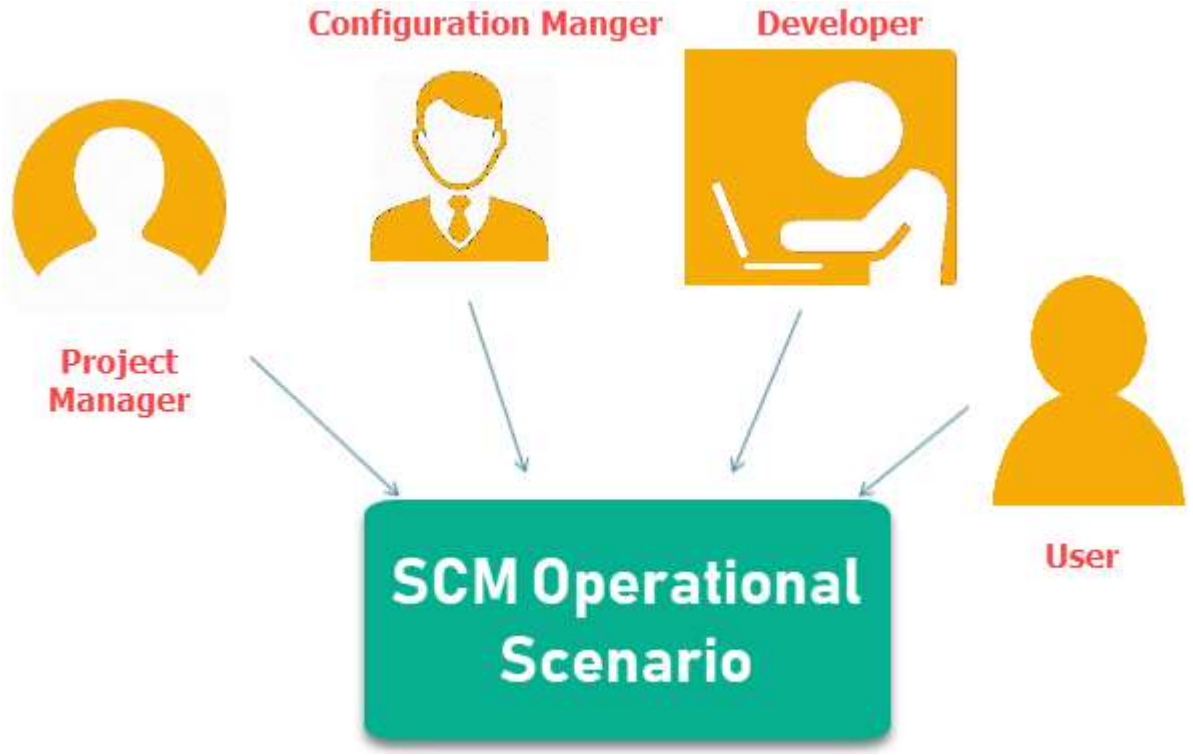
Participant of SCM process

1. **Configuration Manager**
   - Configuration Manager is the head who is Responsible for identifying configuration items.
   - CM ensures team follows the SCM process
   - He/She needs to approve or reject change requests
2. **Developer**
   - The developer needs to change the code as per standard development activities or change requests. He is responsible for maintaining configuration of code.
   - The developer should check the changes and resolves conflicts
3. **Auditor**
   - The auditor is responsible for SCM audits and reviews.
   - Need to ensure the consistency and completeness of release.

Participant of SCM process

4. **Project Manager:**
   - Ensure that the product is developed within a certain time frame
   - Monitors the progress of development and recognizes issues in the SCM process
   - Generate reports about the status of the software system
   - Make sure that processes and policies are followed for creating, changing, and testing

5. **User**
   - The end user should understand the key SCM terms to ensure he has the latest version of the software



Software Configuration Management Plan

- The SCMP (Software Configuration management planning) process planning begins at the early coding phases of a project. The outcome of the planning phase is the SCM plan which might be stretched or revised during the project.
  - The SCMP can follow a public standard like the IEEE 828 or organization specific standard
  - It defines the types of documents to be management and a document naming. Example Test_v1
  - SCMP defines the person who will be responsible for the entire SCM process and creation of baselines.
  - Fix policies for version management & change control
  - Define tools which can be used during the SCM process
  - Configuration management database for recording configuration information.

Software Configuration Management Tools

- Any Change management software should have the following 3 Key features:
- **Concurrency Management:**
  - When two or more tasks are happening at the same time, it is known as concurrent operation. Concurrency in context to SCM means that the same file being edited by multiple persons at the same time.
  - If concurrency is not managed correctly with SCM tools, then it may create many pressing issues.
- **Version Control:**
  - SCM uses archiving method or saves every change made to file. With the help of archiving or save feature, it is possible to roll back to the previous version in case of issues.
- **Synchronization:**
  - Users can checkout more than one files or an entire copy of the repository. The user then works on the needed file and checks in the changes back to the repository.They can synchronize their local copy to stay updated with the changes made by other team members.

Thank You!