**MODULE 3**

**Managing Software Projects**

Contents:

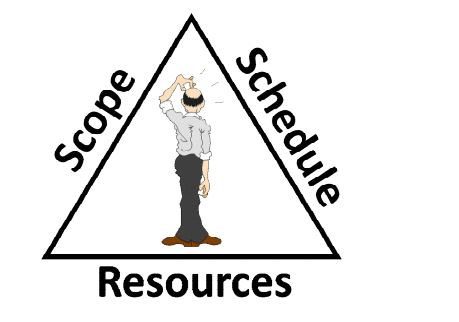
* Project Management Concepts,
* Process and Project Metrics
* -Estimation for Software Projects
* Project Scheduling- Maintenance and Reengineering.

**1)Software Project Management (SPM)**

Software Project Management (SPM) is a proper way of planning and leading software projects. It is a part of project management in which software projects are planned, implemented, monitored, and controlled.

**Need for Software Project Management**

Software is a non-physical product. [Software development](https://www.geeksforgeeks.org/what-is-software-development/) is a new stream in business and there is very little experience in building software products. Most of the software products are made to fit clients’ requirements. The most important is that basic technology changes and advances so frequently and rapidly that the experience of one product may not be applied to the other one. Such types of business and environmental constraints increase[risk in software development](https://www.geeksforgeeks.org/different-types-of-risks-in-software-project-development/) hence it is essential to manage software projects efficiently. It is necessary for an organization to deliver quality products, keep the cost within the client’s budget constraint, and deliver the project as per schedule. Hence, in order, software project management is necessary to incorporate user requirements along with budget and time constraints.





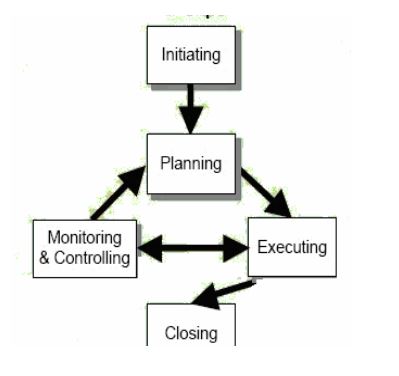
Software Project:

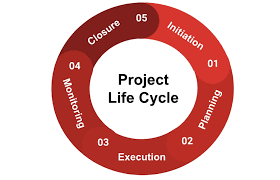
A project is well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery).

A Project can be characterized as:

* Every project may has a unique and distinct goal.
* Project is not routine activity or day-to-day operations.
* Project comes with a start time and end time.
* Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
* Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

**2)PROJECT LIFE CYCLE**

The Project Life Cycle refers to a logical sequence of activities to accomplish the project’s goals or objectives**.**

****

**1)Initiation Phase**

* The start of the project for the project manager, who is responsible for defining the project at a high level.
* This usually begins with a business case, feasibility study, cost-benefit analysis and other types of research to determine whether the project is feasible and should or shouldn’t be undertaken.
* Stakeholders provide input.
* If the project is approved, then a project charter is created, which provides an overview of the project and sets up the stage for your project plan.

**2)** Planning Phase

* This is where the project plan is created, and all involved in the project will follow it. This phase begins by setting SMART (specific, measurable, attainable, realistic, timely) goals.
* The scope of the project is defined and a project management plan is created, identifying cost, quality, resources and a timetable.
* Some of the features of this phase include a scope statement, setting of milestones, communication with stakeholders, risk management plans and a work breakdown structure.

**3)Execution Phase**

The execution step includes allocating resources, conducting meetings, communicating updates, and delivering performance reports.

**4)Monitoring & Controlling Phase**

The project monitoring and controlling phase consists of setting up controls and key performance metrics to measure the effectiveness of the project execution. The monitoring and controlling project phase is very important to make sure the execution goes as planned in terms of schedule, scope and budget baselines.

5) **Closing Phase**

* Completing the deliverables to the satisfaction of your stakeholders is key, of course, but the project manager must now disassemble the apparatus created to fulfill the project. That means closing out work with contractors, making sure everyone has been paid and ensuring that all project documents are signed off on and archived to help with planning future projects.
* Once this has been done, the project manager often has a post-mortem with the project team to highlight what worked and what didn’t work, so that successes can be repeated and mistakes avoided.

**Types of Management in SPM**

**1)Conflict Management**

* [Conflict management](https://www.geeksforgeeks.org/organisational-conflicts-consequences-and-types/) is the process to restrict the negative features of conflict while increasing the positive features of conflict. The goal of conflict management is to improve learning and group results including efficacy or performance in an organizational setting. Properly managed conflict can enhance group results.

**2. Risk Management**

* [Risk management](https://www.geeksforgeeks.org/risk-management-software-engineering/) is the analysis and identification of risks that is followed by synchronized and economical implementation of resources to minimize, operate and control the possibility or effect of unfortunate events or to maximize the realization of opportunities.

**3. Requirement Management**

* It is the process of analyzing, prioritizing, tracking, and documenting requirements and then supervising change and communicating to pertinent stakeholders. It is a continuous process during a project.

**4. Change Management**

* [Change management](https://www.geeksforgeeks.org/change-management-in-software-engineering/) is a systematic approach to dealing with the transition or transformation of an organization’s goals, processes, or technologies. The purpose of change management is to execute strategies for effecting change, controlling change, and helping people to adapt to change

**5.Software Configuration Management**

* Software configuration management is the process of controlling and tracking changes in the software, part of the larger cross-disciplinary field of configuration management. [Software configuration management](https://www.geeksforgeeks.org/software-engineering-system-configuration-management/) includes revision control and the inauguration of baselines**.**

**6.Release Management**

* [Release Management](https://www.geeksforgeeks.org/release-management-in-software-engineering/) is the task of planning, controlling, and scheduling the built-in deploying releases. Release management ensures that the organization delivers new and enhanced services required by the customer while protecting the integrity of existing services.

**Aspects of Software Project Management**

**1. Planning**

* The[software project manager](https://www.geeksforgeeks.org/software-engineering-role-and-responsibilities-of-a-software-project-manager/) lays out the complete project’s blueprint. The project plan will outline the scope, resources, timelines, techniques, strategy, communication, testing, and maintenance steps. SPM can aid greatly here.

**2. Leading**

* A software project manager brings together and leads a team of engineers, strategists, programmers, designers, and data scientists. Leading a team necessitates exceptional communication, interpersonal, and leadership abilities. One can only hope to do this effectively if one sticks with the core SPM principles.

**3. Execution**

* SPM comes to the rescue here also as the person in charge of software projects (if well versed with SPM/[Agile methodologies](https://www.geeksforgeeks.org/software-engineering-agile-software-development/)) will ensure that each stage of the project is completed successfully. measuring progress, monitoring to check how teams function, and generating status reports are all part of this process.

**4. Time Management**

* Abiding by a timeline is crucial to completing deliverables successfully. This is especially difficult when managing software projects because changes to the original project charter are unavoidable over time. To assure progress in the face of blockages or changes, software project managers ought to be specialists in managing risk and emergency preparedness.

**5. Budget**

* [Software project managers](https://www.geeksforgeeks.org/software-engineering-role-and-responsibilities-of-a-software-project-manager/), like conventional project managers, are responsible for generating a project budget and adhering to it as closely as feasible, regulating spending, and reassigning funds as needed. SPM teaches us how to effectively manage the monetary aspect of projects to avoid running into a financial crunch later on in the project.

**6. Maintenance**

* Software project management emphasizes continuous product testing to find and repair defects early, tailor the end product to the needs of the client, and keep the project on track. The [software project manager](https://www.geeksforgeeks.org/a-day-in-the-life-of-a-project-manager/) makes ensuring that the product is thoroughly tested, analyzed, and adjusted as needed. Another point in favor of SPM

PROJECT, PROGRAM, PORTFOLIO, AND OPERATIONS MANAGEMENT

* A project may be managed in three separate scenarios: as a stand-alone project (outside of a portfolio or program), within a program, or within a portfolio.
* Project managers interact with portfolio and program managers when a project is within a program or portfolio.
* For example, multiple projects may be needed to accomplish a set of goals and objectives for an organization. In those situations, projects may be grouped together into a program.
* A program is defined as a group of related projects, subsidiary programs, and program activities managed in a coordinated manner to obtain benefits not available from managing them individually. Programs are not large projects.

A portfolio is defined as projects, programs, subsidiary portfolios, and operations managed as a group to achieve strategic objectives.

* Operations management is an area that is outside the scope of formal project management.
* Operations management is concerned with the ongoing production of goods and/or services. It ensures that business operations continue efficiently by using the optimal resources needed to meet customer demands.
* It is concerned with managing processes that transform inputs (e.g., materials, components, energy, and labor) into outputs (e.g., products, goods, and/or services).

Knowledge Areas & Process Groups

A Project Management Process Group is a logical grouping of project management processes to achieve specific project objectives. Process Groups are independent of project phases. Project management processes are grouped into the following five Project Management Process Groups:

* **Initiating Process Group.** Those processes performed to define a new project or a new phase of an existing project by obtaining authorization to start the project or phase.
* **Planning Process Group.** Those processes required to establish the scope of the project, refine the objectives, and define the course of action required to attain the objectives that the project was undertaken to achieve.
* **Executing Process Group.** Those processes performed to complete the work defined in the project management plan to satisfy the project requirements.
* **Monitoring and Controlling Process Group.** Those processes required to track, review, and regulate the progress and performance of the project; identify any areas in which changes to the plan are required; and initiate the corresponding changes.
* **Closing Process Group.** Those processes performed to formally complete or close the project, phase, or contract.

A Knowledge Area is an identified area of project management defined by its knowledge requirements and described in terms of its component processes, practices, inputs, outputs, tools, and techniques.

Although the Knowledge Areas are interrelated, they are defined separately from the project management perspective. The ten Knowledge Areas identified in this guide are used in most projects most of the time.

The ten Knowledge Areas are:

* **Project Integration Management:** Includes the processes and activities to identify, define, combine, unify, and coordinate the various processes and project management activities within the Project Management Process Groups.
* **Project Scope Management:** Includes the processes required to ensure the project includes all the work required, and only the work required, to complete the project successfully.
* **Project Schedule Management:** Includes the processes required to manage the timely completion of the project.
* **Project Cost Management:** Includes the processes involved in planning, estimating, budgeting, financing, funding, managing, and controlling costs so the project can be completed within the approved budget.
* **Project Quality Management:** Includes the processes for incorporating the organization’s quality policy regarding planning, managing, and controlling project and product quality requirements, in order to meet stakeholders’ expectations.
* **Project Resource Management:** Includes the processes to identify, acquire, and manage the resources needed for the successful completion of the project.
* **Project Communications Management:** Includes the processes required to ensure timely and appropriate planning, collection, creation, distribution, storage, retrieval, management, control, monitoring, and ultimate disposition of project information.
* **Project Risk Management:** Includes the processes of conducting risk management planning, identification, analysis, response planning, response implementation, and monitoring risk on a project.
* **Project Procurement Management:** Includes the processes necessary to purchase or acquire products, services, or results needed from outside the project team.
* **Project Stakeholder Management:** Includes the processes required to identify the people, groups, or organizations that could impact or be impacted by the project, to analyze stakeholder expectations and their impact on the project, and to develop appropriate management strategies for effectively engaging stakeholders in project decisions and execution.

**Project Metrics**

The project metrics describes the characteristic and execution of a project. Examples include effort estimation accuracy, schedule deviation, cost variance, and productivity. Usually measures-

* + Number of software developer
  + Staffing patterns over the life cycle of software
  + Cost and schedule
  + Productivity

Project metrics are quantitative measures used to assess various aspects of a project's performance, progress, and quality. These metrics help project managers and stakeholders make informed decisions, identify issues, and improve processes.

The different project metrics can be categorized into several key areas:

**1. Schedule Metrics**

These metrics focus on the timing aspects of the project, ensuring that tasks and milestones are completed on time.

Planned Value (PV): The budgeted cost for work scheduled to be completed by a specific date.

Actual Cost (AC): The actual cost incurred for work completed by a specific date.

Schedule Variance (SV): The difference between the Planned Value (PV) and the Earned Value (EV). SV = EV - PV.

Schedule Performance Index (SPI): A measure of schedule efficiency, calculated as SPI = EV / PV.

On-Time Completion Rate: The percentage of tasks completed on or before the scheduled time.

**2. Cost Metrics**

These metrics track the financial performance of the project, ensuring that it stays within budget.

Budget at Completion (BAC): The total budget allocated for the project.

Cost Variance (CV): The difference between the Earned Value (EV) and the Actual Cost (AC). CV = EV - AC.

Cost Performance Index (CPI): A measure of cost efficiency, calculated as CPI = EV / AC.

Estimate at Completion (EAC): The expected total cost of the project at completion.

Estimate to Complete (ETC): The expected cost to finish all remaining project work.

Cost of Quality (CoQ): The total cost of ensuring good quality, including prevention, appraisal, and failure costs.

**3. Quality Metrics**

These metrics assess the quality of the project's deliverables and processes.

Defect Density: The number of defects per unit size of the deliverable (e.g., defects per 1,000 lines of code).

Defect Leakage: The percentage of defects that were not found in a particular phase but were discovered later.

Customer Satisfaction Index: A measure of customer satisfaction with the project deliverables, often collected through surveys.

First Pass Yield (FPY): The percentage of deliverables that meet quality standards without requiring rework.

**4. Scope Metrics**

These metrics ensure that the project stays within its defined scope and meets its objectives.

Scope Creep: The extent of uncontrolled changes or continuous growth in a project’s scope.

Requirements Stability Index: The degree to which the project requirements remain unchanged over time.

Completion Rate of Deliverables: The percentage of deliverables completed relative to the total number planned.

**5. Resource Metrics**

These metrics evaluate the utilization and efficiency of resources in the project.

Resource Utilization: The percentage of time that project resources are actively working on project tasks.

Team Productivity: Output per unit of input, such as the number of tasks completed per person-hour.

Resource Availability: The amount of time that resources are available for project work.

**6. Risk Metrics**

These metrics track the project's exposure to risks and the effectiveness of risk management strategies.

Risk Exposure: The potential impact of identified risks on the project.

Risk Mitigation Effectiveness: The success rate of implemented risk mitigation strategies.

Number of Identified Risks: The total number of risks identified during the project lifecycle.

Number of Realized Risks: The number of risks that have occurred and affected the project.

**7. Communication Metrics**

These metrics measure the effectiveness of communication within the project team and with stakeholders.

Stakeholder Engagement Level: The degree of involvement and satisfaction of stakeholders in the project.

Communication Frequency: The regularity of project communication updates (e.g., weekly meetings, status reports).

Feedback Turnaround Time: The time taken to respond to feedback or inquiries from stakeholders.

**8. Process Metrics**

These metrics evaluate the efficiency and effectiveness of the processes used in the project.

Process Cycle Time: The time taken to complete a specific process from start to finish.

Process Adherence Rate: The percentage of processes followed as per defined standards and procedures.

Process Improvement Rate: The rate at which processes are improved over time, often measured by reductions in cycle time or defect rates.

**Project Estimation**

Project estimation is the process of forecasting the time, cost, and resources needed to deliver a project. It typically happens during project initiation and/or planning and takes the project's scope, deadlines, and potential risks into account.

**1)CoCoMo Model**

The Cocomo Model is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time, and quality. The key parameters that define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort and schedule:

\*)**Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.

**\*)Schedule:** This simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks, and months.

**The key parameters that define the quality of any software products in COCOMO Model**

**1)Product Attributes**

* **Size (SLOC or Function Points):** The size of the software, usually measured in Source Lines of Code (SLOC) or Function Points, is a fundamental parameter in COCOMO. Larger software size generally implies greater complexity and effort.
* **Required Software Reliability (RELY):** The level of reliability required from the software. High reliability often increases the effort due to the need for thorough testing and validation.
* **Database Size (DATA):** The size of the database that the software will handle. Larger databases require more effort for design, implementation, and maintenance.
* **Product Complexity (CPLX):** The complexity of the software in terms of control, computational, device-dependent, and data management characteristics. Higher complexity typically increases the effort needed.

**2. Hardware Attributes**

* **Execution Time Constraint (TIME):** The constraints on the execution time of the software. Tight time constraints can increase the development effort.
* **Main Storage Constraint (STOR):** The constraints on the main memory available for the software. Severe constraints may require more effort for optimization.
* **Virtual Machine Volatility (VIRT):** The volatility of the hardware environment. High volatility can increase the effort required for development due to the need for frequent adjustments.
* **Computer Turnaround Time (TURN):** The time taken to get responses from the computer. Longer turnaround times can lead to increased development time and effort.

**3. Personnel Attributes**

* **Analyst Capability (ACAP):** The capability and experience of the systems analysts. Higher capability generally reduces the required effort.
* **Programmer Capability (PCAP):** The capability and experience of the programmers. More capable programmers typically reduce development effort.
* **Personnel Continuity (PCON):** The continuity of the personnel involved in the project. Higher continuity reduces the effort needed as less time is spent on knowledge transfer and ramp-up.

**4. Project Attributes**

* **Use of Modern Programming Practices (MODP):** The degree to which modern programming practices are applied. More modern practices can reduce the effort needed.
* **Use of Software Tools (TOOL):** The extent to which software tools are used during development. Effective use of tools can significantly reduce effort.
* **Required Development Schedule (SCED):** The schedule constraints imposed on the project. Tighter schedules can increase the effort as tasks may need to be expedited.

**3 software projects types of COCOMO Model**

**1. Organic Projects**

Characteristics:

* Small to Medium Size: These projects are typically small to medium in size.
* Well-Understood Requirements: The project requirements are well-understood and stable.
* Experienced Teams: The development team is highly experienced with similar types of projects.
* Less Complex: The software is relatively simple, involving little innovation or new technology.

**Examples:**

* Business applications such as payroll systems, inventory management systems.
* Data processing systems.
* Effort Estimation:
* *E*=2.4×(*KLOC*)^1.05  
  Where 𝐸 is the effort in person-months and 𝐾𝐿𝑂𝐶*KLOC* is the size of the software in thousands of lines of code.

**2. Semidetached Projects**

**Characteristics:**

* Intermediate Size: These projects are of medium size.
* Moderate Requirements Stability: The project requirements may change moderately over time.
* Mixed Experience Levels: The development team has a mix of experience levels.
* Moderate Complexity: The software is of moderate complexity, potentially involving some innovation or new technology.

**Examples:**

* Embedded systems with moderate complexity, such as real-time processing systems.
* Complex business applications with some degree of user interaction.
* Equation:
* 𝐸=3.0×(𝐾𝐿𝑂𝐶)^1.12
* Where 𝐸*E* is the effort in person-months and 𝐾𝐿𝑂𝐶*KLOC* is the size of the software in thousands of lines of code.

**3. Embedded Projects**

**Characteristics:**

* Large Size: These projects are usually large and complex.
* Complex Requirements: The project requirements are often complex and may change frequently.
* Highly Specialized Teams: The development team requires specialized skills and may need to work with unfamiliar technology or systems.
* High Complexity: The software involves high innovation, new technology, or stringent performance requirements.

**Examples:**

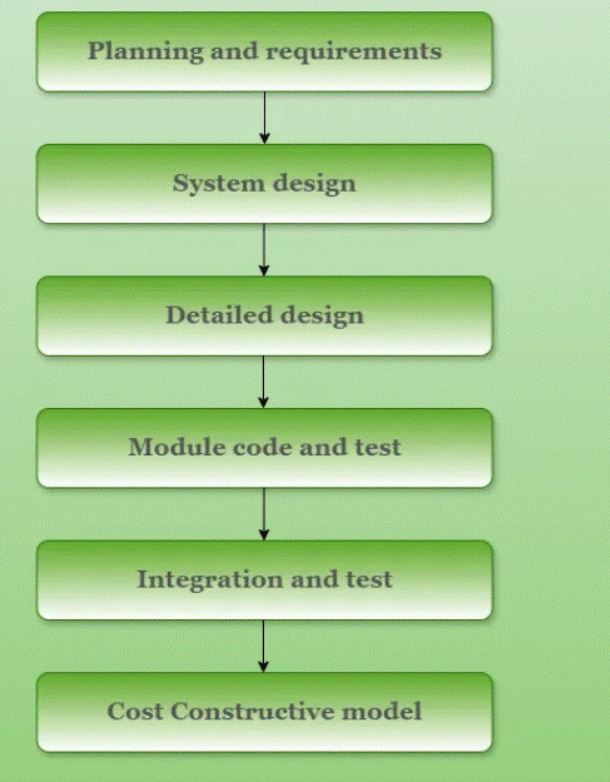
* Military defense systems.
* Real-time flight control systems.
* Telecommunications software.

**Effort Estimation:**

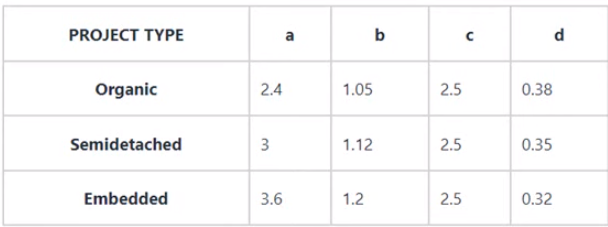
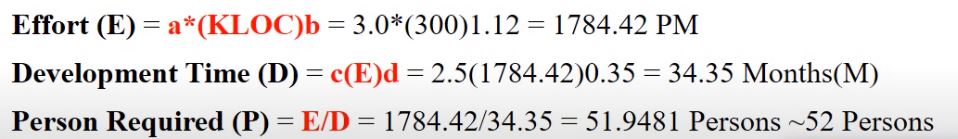
*E*=3.6×(*KLOC*)^1.20 Where 𝐸is the effort in person-months and 𝐾𝐿𝑂𝐶*KLOC* is the size of the software in thousands of lines of code.

**Structure of COCOMO Model**

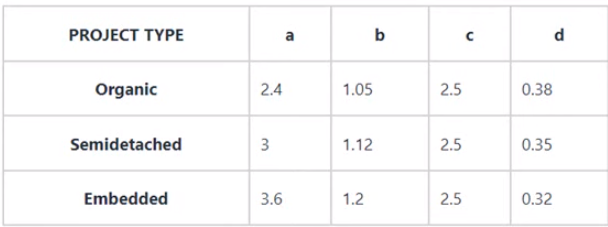
**COCOMO Model Basics**

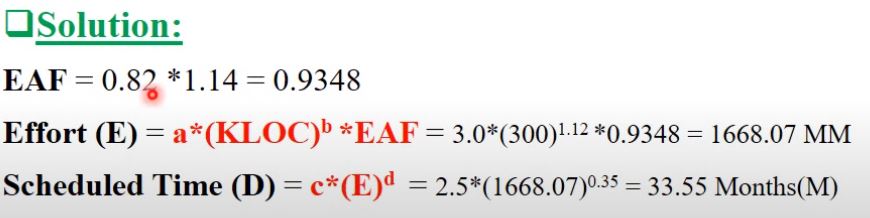
* *E = a(KLOC)^b*
* *Time = c(Effort)^d*
* *Person required = Effort/ time*

1)Consider a Software Project using semi detached mode using 300KLOC.Find out Effort estimation, Development time and person estimation.

 Solution:

2) For a given semidetached project was estimated with a size of 300KLOC.Calculate the effort, scheduled time for development by considerin developer having high application experience and very low experience in programming.





**2)Functional Point (FP) Analysis**

* [**Functional Point**](https://www.geeksforgeeks.org/software-engineering-calculation-of-function-point-fp/) Analysis gives a dimensionless number defined in function points which we have found to be an effective relative measure of function value delivered to our customer.

**Objectives of Functional Point Analysis**

* **Encourage Approximation:** FPA helps in the estimation of the work, time, and materials needed to develop a software project. Organizations can plan and manage projects more accurately when a common measure of functionality is available.
* **To assist with project management:** Project managers can monitor and manage software development projects with the help of FPA. Managers can evaluate productivity, monitor progress, and make well-informed decisions about resource allocation and project timeframes by measuring the software’s functional points.
* **Comparative analysis:** By enabling benchmarking, it gives businesses the ability to assess how their software projects measure up to industry standards or best practices in terms of size and complexity. This can be useful for determining where improvements might be made and for evaluating how well development procedures are working.
* **Improve Your Cost-Benefit Analysis:** It offers a foundation for assessing the value provided by the program concerning its size and complexity, which helps with cost-benefit analysis. Making educated judgements about project investments and resource allocations can benefit from having access to this information.

**Comply with Business Objectives:** It assists in coordinating software development activities with an organization’s business objectives. It guarantees that software development efforts are directed toward providing value to end users by concentrating on user-oriented functionality

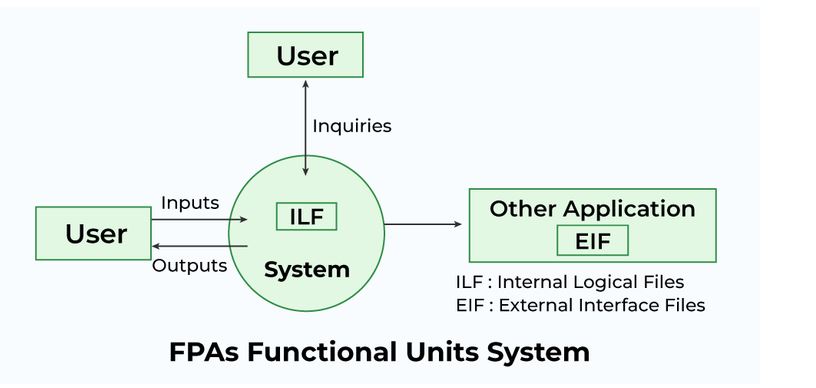
**Types of Functional Point Analysis**

There are two types of Functional Point Analysis:

**1. Transactional Functional Type**

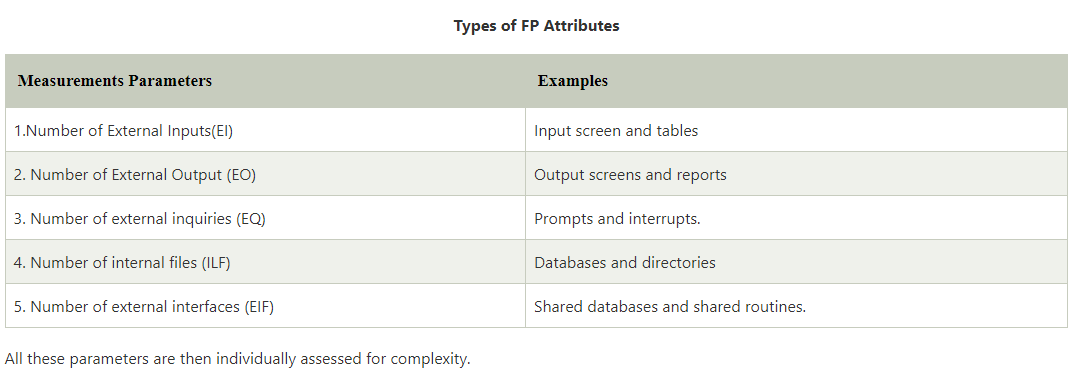
* **External Input (EI):** EI processes data or control information that comes from outside the application’s boundary. The EI is an elementary process.
* **External Output (EO):** EO is an elementary process that generates data or control information sent outside the application’s boundary.
* **External Inquiries (EQ):** EQ is an elementary process made up of an input-output combination that results in [data retrieval.](https://www.geeksforgeeks.org/what-is-information-retrieval/)

**2. Data Functional Type**

* **Internal Logical File (ILF):** A user-identifiable group of logically related data or control information maintained within the boundary of the application.
* **External Interface File (EIF):** A group of users recognizable logically related data allusion to the software but maintained within the boundary of another software.

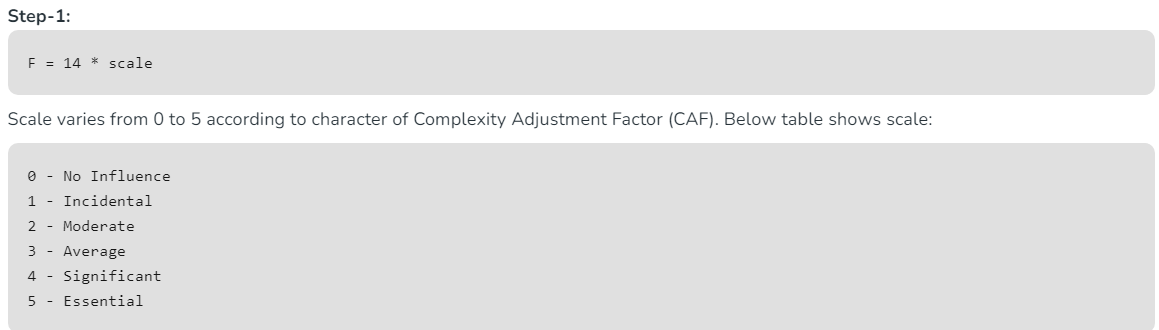
**FPA Procedure at a glance**



Functional Point Estimation

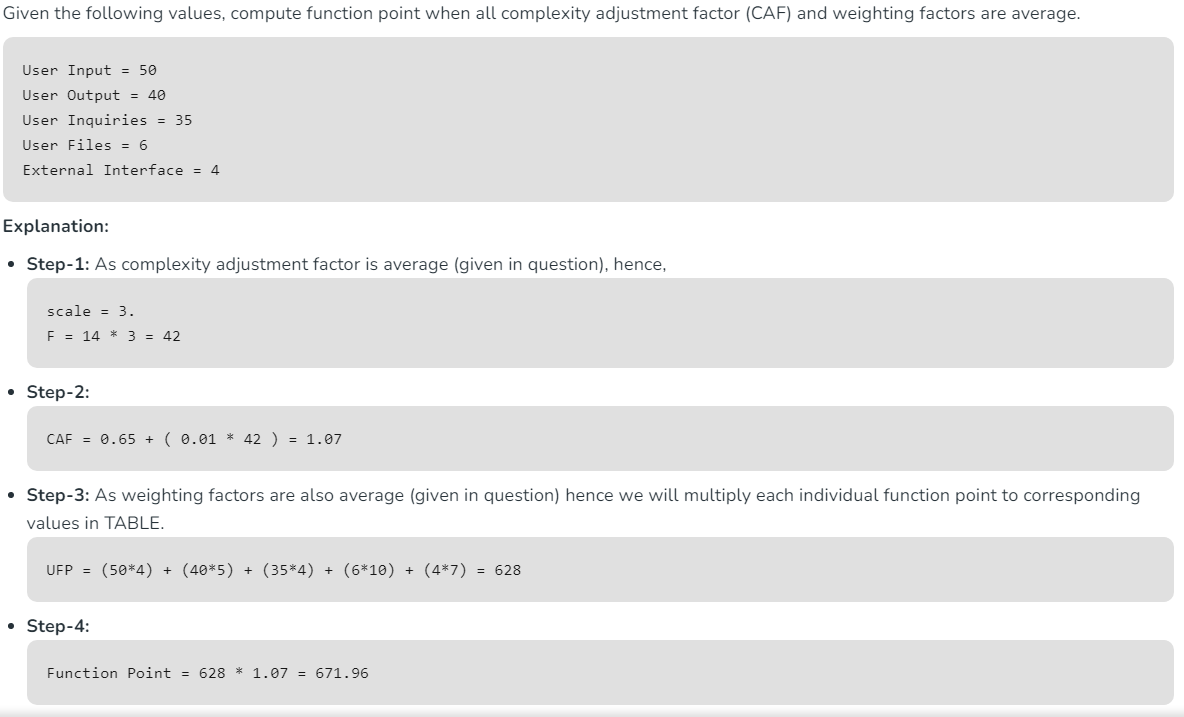
**Calculation of Function Point (FP)**

Counting Function Point (FP):







 Problem1: Given average values: 4,5,4,10,7

Problem 2:

Given the following values, calculate the functional point when complexity adjustment factors are significantly complex product and weighting factors are high.

User input=55

User Output=35

User Enquires =40

User files = 8

External Interfaces=5

High values values: 6,7,6 ,15,10

Solution:



**Cost-benefit analysis in project management**

A cost-benefit analysis in project management is a tool to evaluate the costs vs. benefits of an important project or business proposal. It is a practical, data-driven approach for guiding organizations and managers in making solid investment decisions. It helps determine if a project or investment is financially feasible and beneficial for the organization.

Cost may include the following:

* **Direct costs:** These are costs that are directly related to the proposed project or investment, e.g., materials, labor, and equipment.
* **Indirect costs:**These are related fixed costs that contribute to bringing the project or investment to life, e.g., overhead, administrative, or training expenses.
* **Opportunity costs:**These are the benefits or opportunities foregone when a business chooses one project or opportunity over others. To quantify opportunity costs, you must weigh the potential benefits of the available alternatives.
* **Future costs:**These are costs that may come up later in the project. These costs depend on certain factors happening, e.g., costs of mitigating potential risks.

Benefits may include:

* **Tangible benefits:** These are measurable outcomes that can be easily quantified in monetary terms, e.g., increased revenue or reduced costs.
* **Intangible benefits:** These benefits are difficult to measure in monetary terms. They are indirect or qualitative outcomes, such as improved customer satisfaction or increased employee morale.

**Project Scheduling**

* Scheduling in project management is the listing of activities, deliverables, and milestones within a project.
* A schedule usually includes a planned start and finish date, duration, and resources assigned to each activity.
* Effective project scheduling is a critical component of successful time management.
* The processes for building a schedule are:
  + Plan schedule management
  + Define project activities
  + Sequence activities
  + Estimate resources
  + Estimate durations
  + Develop the project schedule

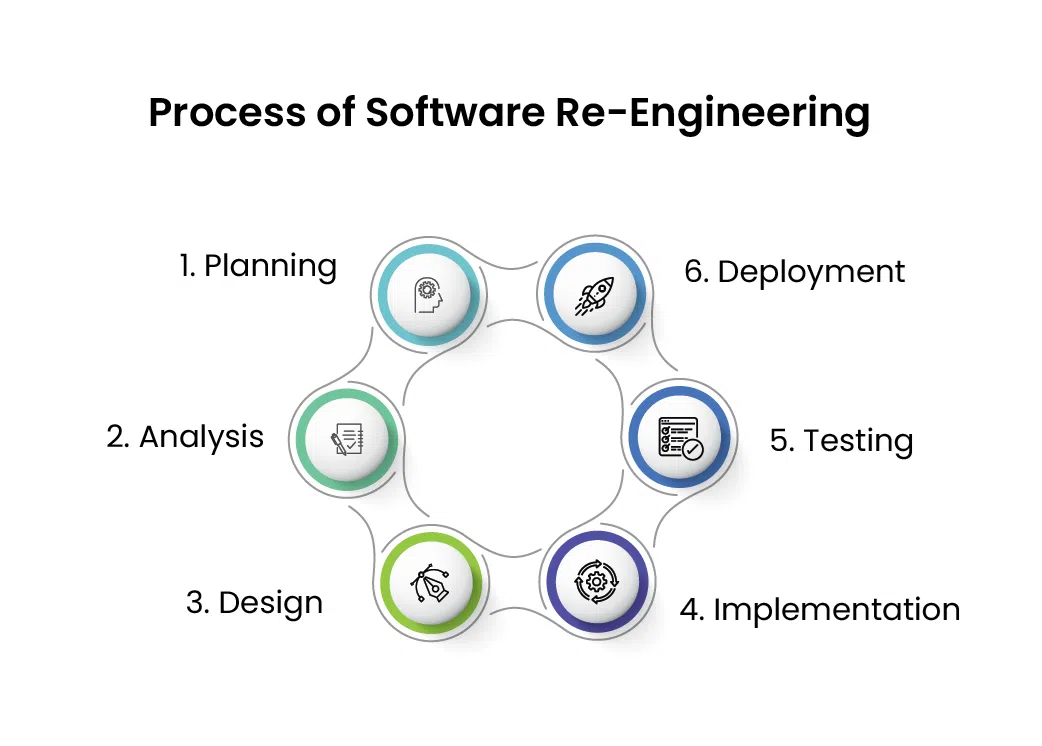
Three main types of project schedules:

* + **Master project schedule:** A master schedule tends to be a simplified list of tasks with a timeline or project calendar.
  + **Milestone schedule or summary schedule:** This type of project schedule tracks major milestones and key deliverables, but not every task is required to complete the project.
  + **A detailed project schedule:** This is the most thorough project schedule, as it identifies and tracks every project activity. If you have a complex, large, or lengthy project, it’s important to have a detailed project schedule to help track everything.The most common form of project schedule is a **Gantt chart**. Both a milestone schedule and a detailed project schedule can be created as a Gantt chart. When choosing scheduling software, look for project scheduling tools that allow you to create different views from the same schedule.

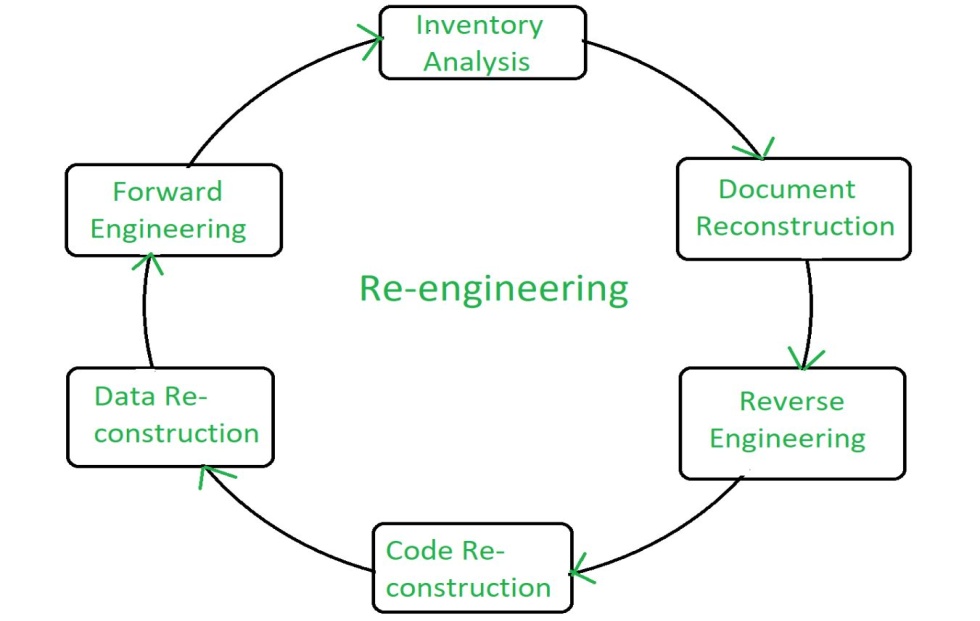
**Re-engineering**

* Software Re-engineering is a process of software development that is done to improve the maintainability of a software system. Re-engineering is the examination and alteration of a system to reconstitute it in a new form.
* This process encompasses a combination of sub-processes like reverse engineering, forward engineering, reconstructing, etc.
* It is the process of analyzing, designing, and modifying existing software systems to improve their quality, performance, and maintainability.

1. This can include updating the software to work with new hardware or software platforms, adding new features, or improving the software’s overall design and architecture.
2. Software re-engineering, also known as software restructuring or software renovation, refers to the process of improving or upgrading existing software systems to improve their quality, maintainability, or functionality.
3. It involves reusing the existing software artifacts, such as code, design, and documentation, and transforming them to meet new or updated requirements.



* **Planning:** The first step is to plan the re-engineering process, which involves identifying the reasons for re-engineering, defining the scope, and establishing the goals and objectives of the process.
* **Analysis:** The next step is to analyze the existing system, including the code, documentation, and other artifacts. This involves identifying the system’s strengths and weaknesses, as well as any issues that need to be addressed.
* **Design:** Based on the analysis, the next step is to design the new or updated software system. This involves identifying the changes that need to be made and developing a plan to implement them.
* **Implementation:** The next step is to implement the changes by modifying the existing code, adding new features, and updating the documentation and other artifacts.
* **Testing:** Once the changes have been implemented, the software system needs to be tested to ensure that it meets the new requirements and specifications.
* **Deployment:** The final step is to deploy the re-engineered software system and make it available to end-users

**Steps in Re-engineering**