# Injibara University

# College of Natural and Computational Science

# Department of Computer Science

## Fundamentals of Software Engineering

*Course Code:CoSc3091*

*Molla K. (MSc)*

# Chapter-4
# Introduction to Software Project Management

# Software Project Management

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

## A Project can be characterized as:

- Project comes with a **start** and **end** time.
- Project is not a routine activity or day-to-day operation.
- Project needs adequate resources in terms of time, manpower, finance, material.

# Software Project

- It is the complete process of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

## Need of Software Project Management

- Software is said to be an intangible product.
- Software development is a kind of all new stream in world business and there is very little experience in building software products.
- Most software products are modify made to fit client's requirements.

# Software Project Manager

- It is a person who undertakes the responsibility of executing the software project.
- SPM is carefully aware of all the phases of SDLC that the software would go through.
- The responsibilities that a project manager carried:

**Managing People**

- Act as project leader.
- Lesion with stakeholders.
- Managing human resources

# Software Project Manager-cont'd

- **Managing Project**
  - Defining and setting up project scope.
  - Managing project management activities.
  - Monitoring progress and performance .
  - Risk analysis at every phase.
  - Take necessary step to avoid or come out of problems.
  - Act as project spokesperson.

# SPM--Cont'd

- SPM contains a number of activities, that includes **planning** of project, deciding **scope** of software product, **estimation** of cost in various terms, **scheduling** of tasks and events, and resource management.
- ***Project management activities may include:***
  - ✓Project **Planning**
  - ✓**Scope** Management
  - ✓ Project **Estimation**

# ➢ Project Planning

- It a is task, which is performed before the production of software actually starts.

## Scope Management

- It defines scope of the project. This includes all the activities, process need to be done in order to make a deliverable software product.
- ✓ It is an essential because it creates **boundaries** of the project by clearly defining what would be done in the project and what would not be done.

# Software Management Activities…..

- **Scope Mgt**

✓During Project Scope management, it is necessary to :

✓Define the scope.

✓Decide its verification and control.

✓ Divide the project into various smaller parts for simplicity of management.

✓Verify the scope.

✓ Control the scope by incorporating changes to the scope.

# Software Management Activities…..

## Project Estimation

- By the correct estimation, managers can manage and control the project more **efficiently** and **effectively**.

Project estimation may involve the following:

### I) Software Size Estimation

- It may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating number of function points in the software.
- Lines of code(KLOC) depend upon coding practices.
- Function points(FP) vary according to the user or Software Requirements.

# Project Estimation ->cont'd

## II) Effort Estimation

- The manager estimates efforts in terms of personnel requirement and man-hour required to produce the software.
- For effort estimation software size should be known.
- This can either be derived by manager's experience, historical data of organization, or software size can be converted into efforts by using some standard formulae.

**Simplistic model for an estimate :**

***Estimated effort=(system size)/productivity***

System size = Lines of code(LOC)

**Productivity = Lines of Code per day**

# Project Estimation ....Cont'd

## III) Time estimation

- Once **size** and **efforts** are estimated, the time required to produce the software can be estimated.
- Efforts required is separated into sub categories as per the requirement specifications and interdependency of various components of software.
- Software tasks are divided into smaller tasks, activities or events by Work Breakthrough Structure (WBS).
- The tasks are scheduled on day-to-day basis or in calendar months.
- The sum of time required to complete all tasks in hours or days is the total time invested to complete project.

# Project Estimation...cont

## IV)Cost Estimation

This might be considered as the most difficult of all because it depends on more elements than any of the previous ones.

- For estimating project cost, it is required to consider:
  - Size of the software
  - Software quality
  - Hardware
  - Additional software or tools, license etc.
  - Experienced personnel with task-specific skills
  - Travel involved
  - Communication
  - Training and support

# Project Estimation Techniques

- Project manager can estimate the listed factors using two broadly recognized techniques :
  - **Decomposition Technique**

This technique assumes the software as a product of various compositions.

- There are two main models –
- **Line of Code(LOC):** The estimation is done on behalf of number of **line of codes** in the software product.
- **Function Points(FP):** The estimation is done on behalf of number of **function points** in the software product.

# Project Estimation Techniques…Con't

## Empirical Estimation Technique

- This technique uses empirically derived formula to make estimation.
- These formula are based on **LOC** or **FPs**.

### Putnam Model

- This model is made by Lawrence H. Putnam, which is based on Norden's frequency distribution (Rayleigh curve).
- Putnam model maps **time** and **efforts** required with software **size**.

### COCOMO

- It stands for **Constructive Cost Model**, developed by Barry W. Boehm.
- It divides the software product into three categories of software: **organic**, **semi-detached**, and **embedded**

# Project Scheduling

- Project Scheduling in a project refers to roadmap of all activities to be done with specified order and within time period allotted to each activity.
- Project managers rise to define various tasks, and project **milestones** and then arrange them keeping various factors in mind.
- They look for tasks like in **critical path** in the schedule, which are necessary to complete in specific manner (because of task interdependency) and strictly within the time allocated.

# Project Scheduling..(cont)

For scheduling a project, it is necessary to :

- ✓ Break down the project tasks into smaller, manageable form.
- ✓Find out various tasks and correlate them.
- ✓Estimate time frame required for each task.
- ✓Divide time into work-units.
- ✓Assign sufficient number of work-units for each task.
- ✓Calculate total time required for the project from start to finish.

# Project Risk Management

- It involves all activities relating to **identification**, **analyzing** and making provision for predictable and non-predictable risks in the project.

## Risk may include the following:

- ✓Experienced staff leaving the project and new staff coming in.
- ✓Change in organizational management.
- ✓Requirement change or misinterpreting requirement.
- ✓Under-estimation of required **time** and **resources**.
- ✓ Technological changes, environmental changes, business competition.

# Risk Management Process

There are following activities involved in risk management process:

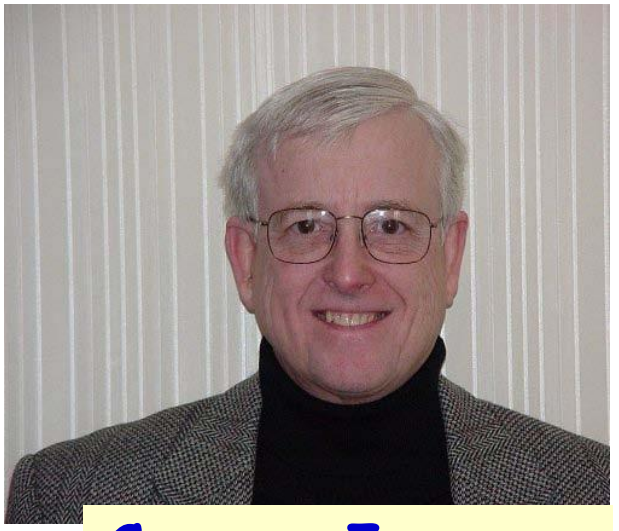
- ***Identification*** - Make note of all possible risks, which may occur in the project.
- ***Categorize*** - Categorize known risks into high, medium and low risk intensity as per their possible impact on the project.
- ***Manage*** - Analyze the probability of occurrence of risks at various phases.
- Make plan to avoid or face risks.
- ***Monitor***-Closely monitor the potential risks and their early symptoms.
- **Risk analysis and prioritization** – which are the most serious risks.

# Project Execution and Monitoring

- In this phase, the tasks described in project plans are executed according to their schedules.
- **Execution:** Needs monitoring in order to check whether everything is going according to the plan.
- **Monitoring:** It is observing to check the probability of risk and taking measures to address the risk or report the status of various tasks.
- These measures include :
  - **Activity Monitoring**- All activities scheduled within some task can be monitored on day-to-day basis.
  - **Status Reports**– It contain status of activities and tasks completed within a given time frame, generally a week.
  - Status can be marked as finished, pending or work-in-progress etc.
  - **Milestones Checklist**- Every project is divided into multiple phases where major tasks are performed (milestones) based on the phases of SDLC.

# Analysis of Failed Projects…

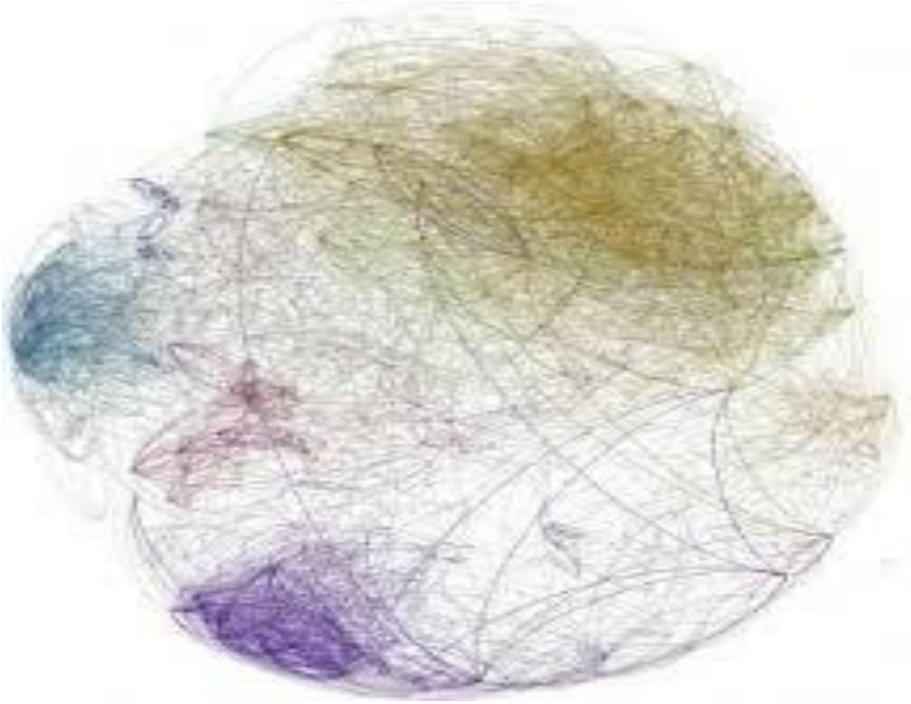
- Investigated 250 large projects.
- Most unsuccessful projects showed weaknesses in:
  - Project Planning
  - Cost Estimation
  - Measurements
  - Milestone Tracking
  - Change Control
  - Quality Control



Capers Jones

…an interesting aspect of the six problem areas is that all are associated with project management rather than with technical aspects.

# In What Ways Management of Software Projects Differ from Management of Engineering Systems?



- **Intangible:**
  - Software is invisible till it is ready and runs
- **Large Change Impact:**
  - A small change request can have poor impact
  - Change is the rule rather than exception
- **Intellectual work**

# Why Software Project Management is Hard?

- Management of intellectual and complex work.
- It is hard to manage anything that you cannot see.
- Changing customer requirements.
- Manpower turnover.

# What Is a Project?

- A project is "a temporary endeavor undertaken to create a unique product, service, or result" (PMBOK® Guide, Fifth Edition, 2012)
- A set of activities undertaken within a defined time period in order to meet a specific set of goals/objectives within a budget.
- A project generally shows most of the following conditions:
  - **It is finite.**
  - **Usually complex.**
  - **Non-repetitive.**
  - **Requires resources.**

- Task: **A small piece of work. It is** usually repetition of a previously accomplished task

# Characteristics of projects

A task is more 'project-like' if it is:

- Non-routine.
- Planned
- Aiming at a specific target
- Carried out for a customer
- Carried out by a temporary work group
- Involving several specialisms
- Made up of several different phases
- Constrained by time and resources
- Large and/or complex

# What is Management?

This involves the following activities:

- **Planning** – Deciding what is to be done
- **Organizing** – Making arrangements
- **Staffing** –Selecting the right people for the job
- **Directing** – Giving instructions continued.
- **Monitoring** – Checking on progress.
- **Controlling**– Taking action to remedy hold-ups.
- **Innovating**–Coming up with solutions when problems emerge
- **Representing**– Communicate with clients, users, developers and other stakeholders.

# Project Stakeholders

- They are individuals and organizations that are actively involved in the project, or whose interests may be positively or negatively affected as a result of project execution or project completion; they may also apply influence over the project and its results.
- Key Stakeholders:
  - **Project manager** —the individual responsible for managing the project.
  - **Customer** —the individual or organization that will use the project's product or service.
  - **Project team members** —the group that is performing the work of the project.
  - **Sponsor**—the individual or group that provides the financial resources for the project.

# Project Stakeholders

- **Internal project stakeholders:**
  - Project manager, Project team, top management.
- **External project stakeholders:**
  - Project's customers (if they are external to the organization), suppliers.

## *Are software* projects really different from other projects?

Not really …but

- Invisibility
- Complexity
- Conformity/consistency
- Flexibility

make software more problematic to build than other engineered artefacts.

# Sounds of Failure's Base Steps…

- Development team doesn't understand customer's needs.
  - Project scope is poorly defined.
  - Changes poorly managed.
  - Chosen technology changes.
  - Business needs change.
  - Unrealistic deadlines.
  - Inexperienced team.
  - Poor project management.

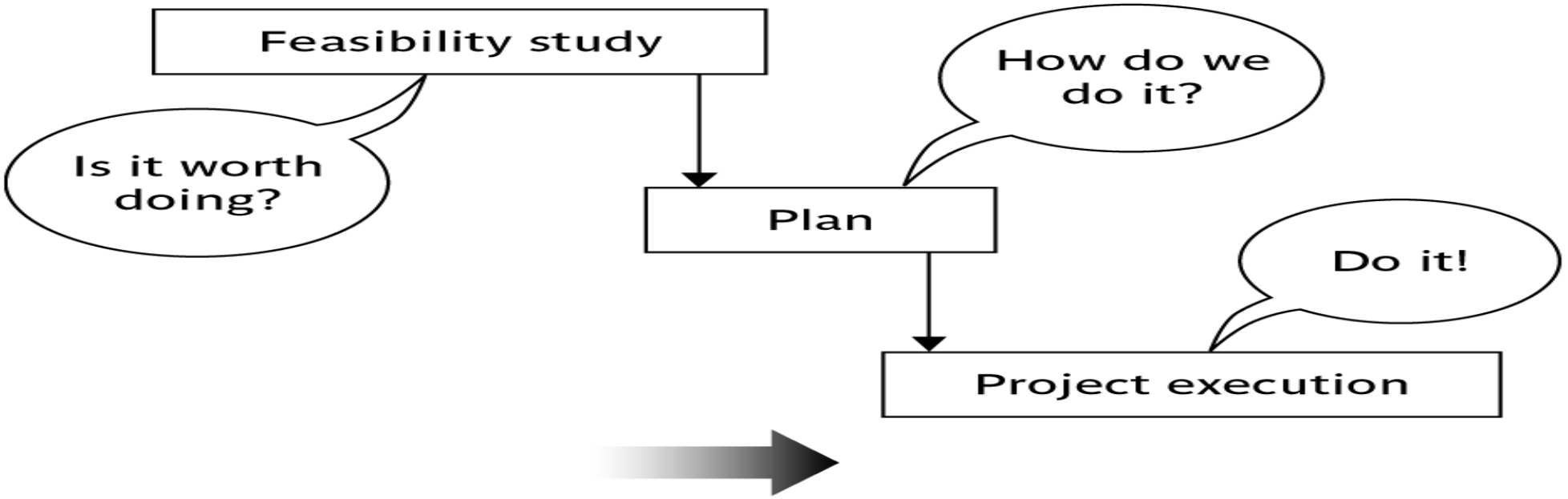
# What is Project Management?

**Project Management = *Plan the work* and *work the plan***

Project Manager's Functions:

- **Planning**: Estimate and schedule resources
- **Organization**: Who does what
- **Staffing**: Recruiting and motivating personnel
- **Directing**: Ensure team acts as a whole
- **Monitoring and Controlling**: Detect plan deviations+ corrective actions

## Major Responsibilities of a Project Manager over Project Life Cycle




### Feasibility study:

Is the project technically feasible and valuable from a business point of view?

### Planning:

Only done if project is feasible

### Execution monitoring and control:

Monitor and control plan implementation, but plan may be changed as execution proceeds

# Project Planning

- Initial plan made before development starts, then updated frequently.
- Important activities:
  - **Estimation**
  - **Scheduling**
  - **Staffing**
  - **Risk management**

# Cont'd

1. Steps required to accomplish the project **objectives**.
2. Tasks needed to be done at each step (using **WBS**).
3. Estimate of how much **effort** each task requires.
4. The **resources** required for each task.
5. (Given 3. and 4.) Calculation of how **long** each task/step will take.
6. (Given 4. and 5.) Calculation of task, step and project **costs**.
7. The inter-dependencies of task.
8. The schedule for each task and the whole project (Milestones, Deliverables, costs, payments).

# Monitoring and Control

- **Lasts for entire active project duration.**
- **Monitoring** – Checking on progress, revising plans.
- **Controlling**–Taking action to remedy/solution hold-ups.
- **Innovating** – Coming up with solutions when problems emerge.
- **Representing** – Linking with clients, users, developers and other stakeholders.

# Setting Objectives

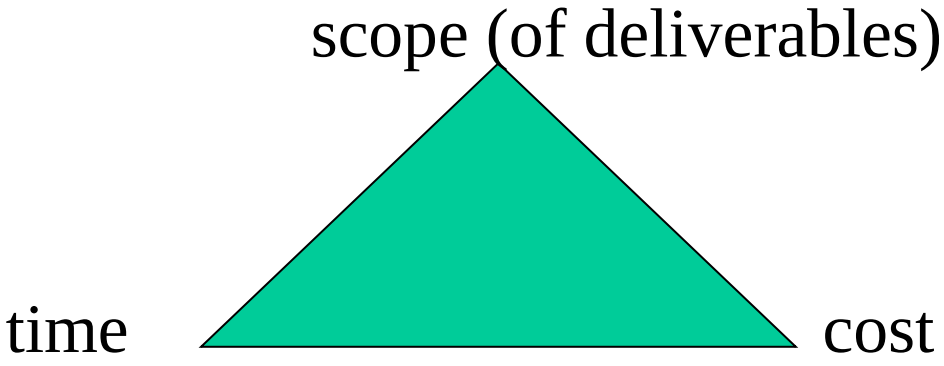
- Answering the question '*What do we have to do to have a success?*'
- Need for a *project authority*
  - Sets the project **scope**.
  - Allocates/approves **costs**.
- Could be one person - or a group
  - Project Board
  - Project Management Board
  - Steering committee

# Objectives should be SMART

- **S – Specific**, that is, concrete and well- defined.
- **M– Measurable**, that is, satisfaction of the objective can be objectively judged.
- **A – Achievable**, that is, it is within the power of the individual or group concerned to meet the target.
- **R – Relevant**, the objective must relevant to the true purpose of the project.
- **T – Time constrained**: there is defined point in time by which the objective should be achieved.

# Project Success/Failure

- **Degree to which objectives are met**




- ✓ In general if, for example, project is running out of time, this can be improved for by **reducing scope** or **increasing costs**.
- ✓ Similarly **costs** and **scope** can be protected by adjusting other corners of the 'project triangle'.

# COCOMO MODEL

- Boehm postulated that any software Dev't project can be classified into one of the following three categories based on the development complexity: **organic**, **semidetached**, and **embedded**.
- In order to classify a product into the identified categories, Boehm not only considered the characteristics of the product but also those of the development **team** and development **environment**.
- Normally, data processing programs are considered to be **application** programs.
- **Compilers**, **linkers**, etc., are **utility** programs.
- **OS** and **real-time** system programs, etc. are system programs.

# Cont'd

- Boehm's[1981] definition of **organic**, **semidetached**, and **embedded** systems are elaborated below.

  - ***Organic***

    - A development project can be considered of organic type, if:
    - The project deals with developing a well understood **application program**,
    - The size of the development team is reasonably **small**,
    - The team members are **experienced** in developing similar types of projects.

# Cont'd

## Semidetached

- A development project can be considered of semidetached type:
  - ✓ If the development consists of a **mixture** of **experienced** and **inexperienced** staff.
  - Team members may have **limited experience** on related systems but may be **unfamiliar** with some aspects of the system being developed.

## Embedded

A development project is considered to be of embedded type:

- ✓ If the software being developed is strongly coupled to complex hardware, or
- ✓ If the stringent regulations on the operational procedures exist.

# Basic COCOMO Model

- The basic COCOMO model gives an approximate estimate of the project parameters.
- The basic COCOMO estimation model is given by the following expressions:

$$\text{Effort} = a_1 \times (\text{KLOC})^{a_2} \text{ PM}$$

$$\text{Tdev} = b_1 \times (\text{Effort})^{b_2} \text{ Months}$$

- Where, KLOC is the estimated size of the software product expressed in Kilo Lines of Code,
- **a1, a2, b1, b2** are constants for each category of software products,
- **Tdev** is the estimated time to develop the software, expressed in months,
- **Effort** is the total effort required to develop the software product, expressed in person months (PMs)

# Cont'd

| Software Project | $a_b$ | $b_b$ | $c_b$ | $d_b$ |
|---|---|---|---|---|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semidetached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Basic COCOMO coefficients

# Example

- Assume that the size of an organic type software product has been estimated to be 32,000 lines of source code. Assume that the average salary of software engineers be Rs. 15,000/- per month. Determine the effort required to develop the software product and the supposed development time.

From the basic **COCOMO** estimation formula for **organic** software:

$$\text{Effort} = 2.4 \times (32)^{1.05} = 91 \text{ PM}$$

$$\text{Dev'tTime} = 2.5 \times (91)^{0.38} = 14 \text{ months}$$

Cost required to develop the product = 14 x 15,000 = Rs. 210,000/-

So, it is mostly used for small medium sized software.

- ✓ It is not detail like intermediate and detail model .
- ✓ It estimates the s/w in a rough and quick manner. It contains 3 modes with their parameters.

# Intermediate COCOMO model

- The basic COCOMO model assumes that **effort** and **development** time are functions of the **product size** only.

**Example** : A new project with estimated **400KLOC embedded** system has to be developed. Project manager has a choice of hiring from 2 pools of developers: Very high capable(with app) with very little experience in programming languages or developer of low quality but a lot of programming languages experienced which is better choice in terms of 2 pool?

## Solution

✓Effort Adjustment Factor(EAF) can be calculated by multiplying all the values that have been obtained after categorizing each cost driver.

$E = a(KLOC)^{b} \times EAF$

$E = 2.8(400)^{1.20X}EAF$

**Case** I: EAF= **0.82*1.14=0.934**

$E = 2.8(400)^{1.20} * 0.934$

**$DT = 2.5(3470)^{0.32} = 33.9M$**

**Case** II: $E = a(KOLC)XEAF$

EAF=1.29*0.95 =1.22 where EAF= LAEXP x HLEXP

**E=3412X1.22=4528PM**

$DT = 2.5(4528)^{0.32} =$ **36.9M**

# COCOMO: Detailed Development Model

✓It calculates the effect of cost Derivers on each phases of SDLC.

✓It is phase sensitive.

✓It is re usability of the code.

✓It uses phases- sensitive **effort multipliers** for each cost driver.

•**To determine the amount of effort required to complete each phase of SDLC.**

✓It establishes **module-subsystem- system** hierarchy.

▪**The rating of cost driver is done at that level only where the cost driver is most affected/ susceptible to variable.**

➢**Adjustment Factor(A)** which describes the reusability of code

$$A= 0.4(DD)+0.3C+0.3I$$

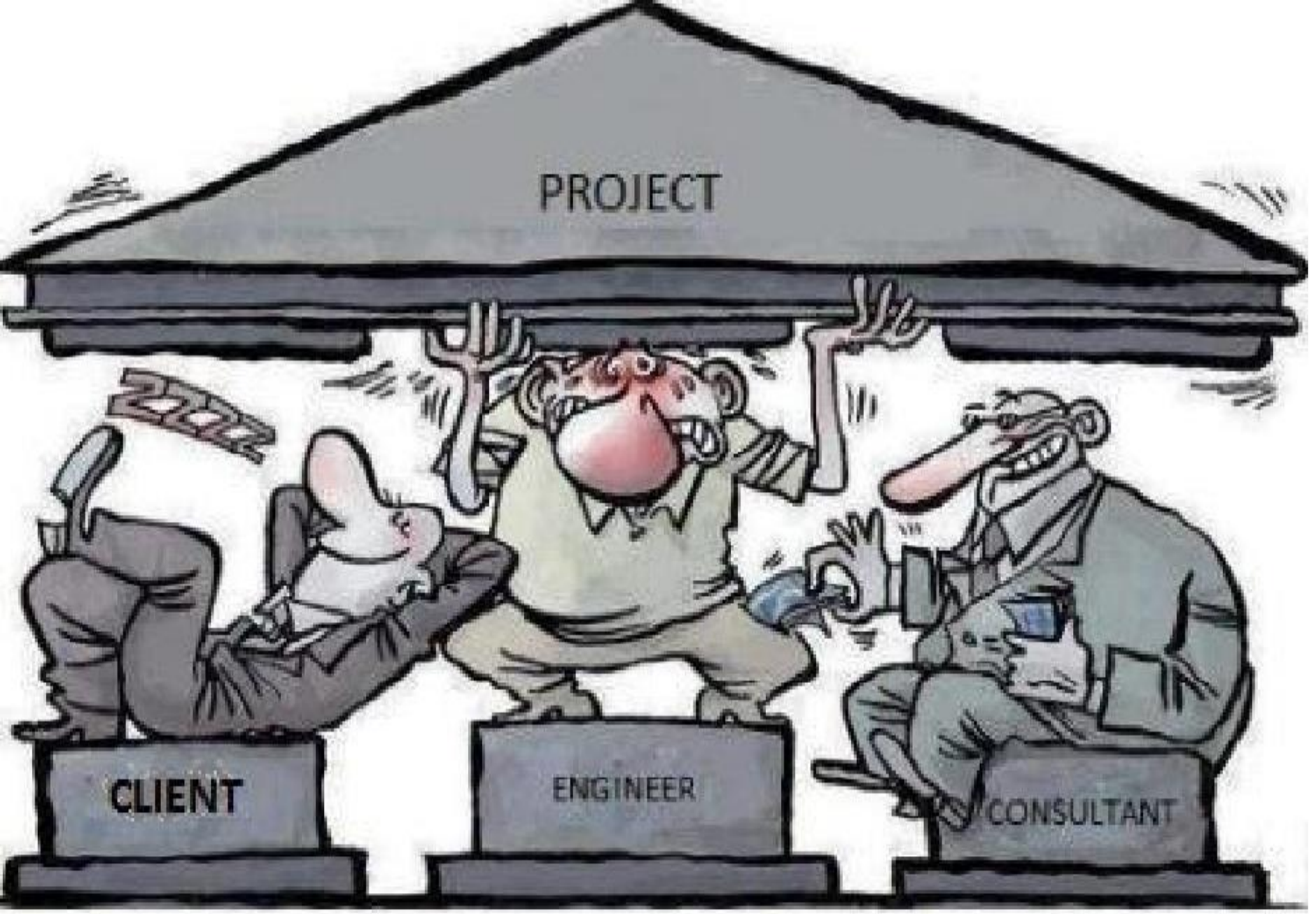
➢**Size Equivalent** : it is affected by Adjustment Factor.

$$\text{Size Equivalent} =(S*A)/100$$

# Summary

## The 3 classes of the software projects

| COCOMO | Project Size | Nature of Project | Innovation | Deadline of project |
|---|---|---|---|---|
| Organic | 2-50KLOC | Small size<br>Experienced developer<br>Eg. payroll system | Title | Not Tight |
| Semidetached | 50-300KLOC | Medium size project.<br>Medium size team<br>Eg. Db system | Medium | Medium |
| Embedded | Over 300KLOC | Large project.<br>Real time system.<br>Eg ATM | Significant | Tight |


PROJECT
ZZZ
CLIENT
ENGINEER
CONSULTANT