COST ESTIMATION AND MAINTENANCE

COST ESTIMATION

Cost estimation can be defined as the approximate judgment of the costs for a project. Cost estimation will never be an exact science because there are too many variables involved in the calculation for a cost estimate, such as human, technical, environmental, and political. Furthermore, any process that involves a significant human factor can never be exact because humans are far too complex to be entirely predictable. Furthermore, software development for any fair-sized project will inevitably include a number of tasks that have complexities that are difficult to judge because of the complexity of software systems.

➢ COST ESTIMATION IN PROJECT PLANNING

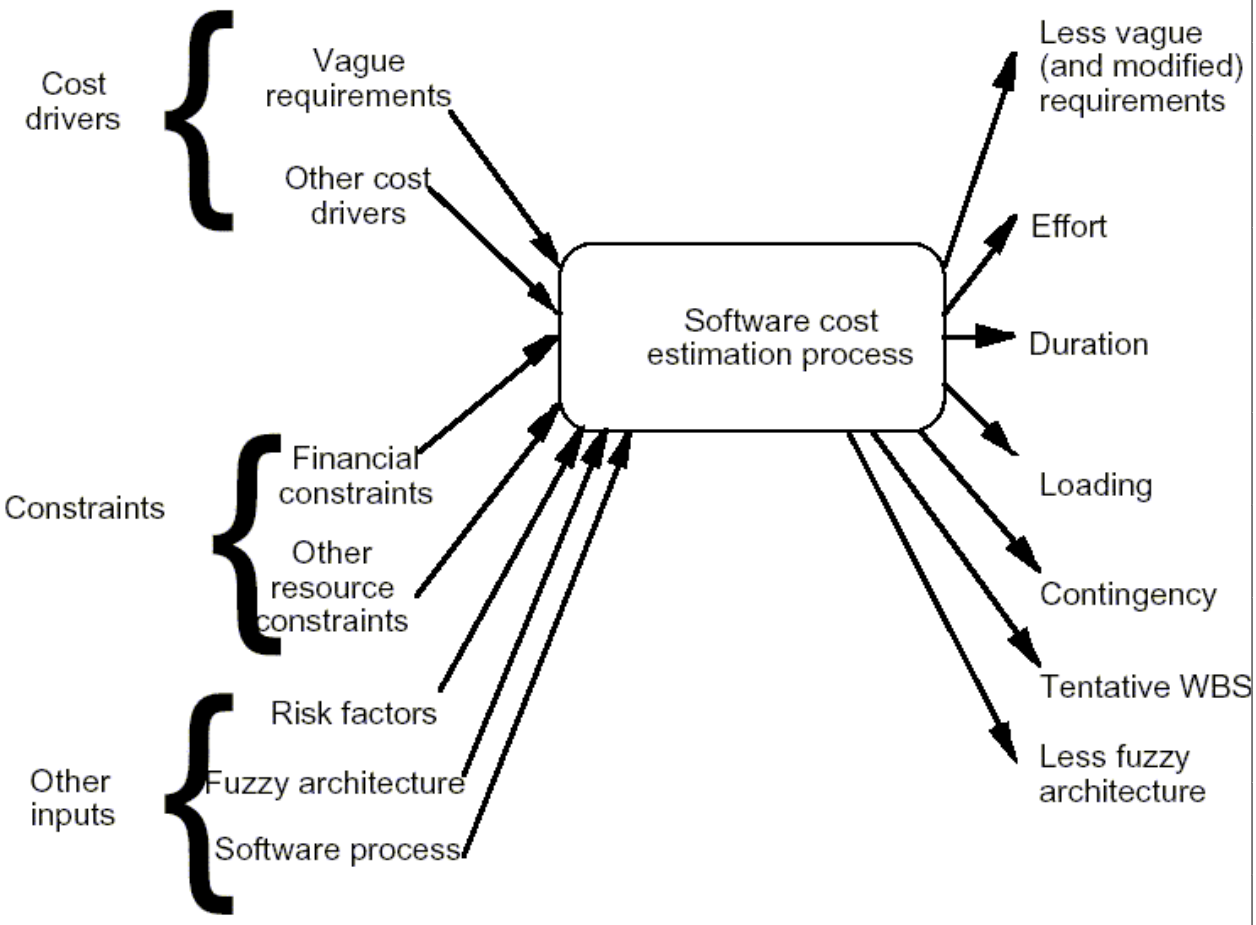
Cost estimation is an important tool that can affect the planning and budgeting of a project. Because there are a finite number of resources for a project, all of the features of a requirements document can often not all are included in the final product. A cost estimate done at the beginning of a project will help determine which features can be included within the resource constraints of the project (e.g., time). Requirements can be prioritized to ensure that the most important features are included in the product. The risk of a project is reduced when the most important features are included at the beginning because the complexity of a project increases with its size, which means there is more opportunity for mistakes as development progresses. Thus, cost estimation can have a big impact on the life cycle and schedule for a project.

Cost estimation can also have an important effect on resource allocation. It is prudent for a company to allocate better resources, such as more experienced personnel, to costly projects. Manpower loading is a term used to measure the number of engineering and management personnel allocated to a project in a given amount of time. Most of time, it is worse for a company if a costly project fails than if a less costly project fails. When tools are used for estimation, management and developers can even experiment with trading off some resources (or factors) with others while keeping the cost of the

project constant. For example, one tradeoff may be to invest in a more powerful integrated development environment (IDE) so that the number of personnel working on a project could be reduced. Cost estimation has a large impact on project planning and management.

➢ COST ESTIMATION PROCESS

In order to understand the end result or the outputs of the software cost estimation process we must first understand what is software cost estimation process. By definition, software cost estimation process is a set of techniques and procedures that is used to derive the software cost estimate. There is usually a set of inputs to the process and then the process uses these inputs to generate or calculate a set of outputs.



❖ TECHNIQUES USED IN COST ESTIMATION

There is an abundance of techniques and models which help in cost estimation in the software industry. Here's a brief outline of the various techniques with a mention of their specialties and limitations.

- **Algorithmic (Parametric) model**
- **Expert Judgment (Expertise Based)**
- **Top – Down**
- **Bottom - Up**
- **Estimation by Analogy**
- **Price to Win Estimation**

1. Algorithmic (Parametric) Model

This software cost estimation technique use the mathematical equations to perform the software estimation. The mathematical equations are based on historical data or theory. SLOC (source line of code), function points, and other cost drivers are the inputs. For most algorithmic model, calibration to the specific software environment can be performed to improve the estimation. Examples of the parametric models are COCOMO (Constructive Cost Model), COCOMO II, Putnam's software life-cycle model (SLIM).

Advantages:

- Generate repeatable estimations
- Easy to modify input data
- Easy to refine and customize formulas
- Objectively calibrated to experience

Disadvantages:

- Unable to deal with exceptional conditions
- Some experience and factors cannot be quantified
- Sometimes algorithms may be proprietary

2. Expert Judgment

This technique captures the experience and the knowledge of the estimator who provides the estimate based on their experience from a similar project to which they have participated. Examples are the Delphi, Wideband Delphi and Work Breakdown Structure (WBS).

Advantages:

- Useful in the absence of quantified, empirical data.
- Can factor in differences between past project experiences and requirements of the proposed project
- Can factor in impacts caused by new technologies, applications and languages.

Disadvantages:

- Estimate is only as good expert's opinion
- Hard to document the factors used by the expert.

3. Top-Down

This technique is also called Macro Model, which utilizes the global view of the product and then partitioned into various low level components. Example of this technique is the Putnam model.

Advantages:

- Requires minimal project detail
- Usually faster and easier to implement
- Focus on system level activities

Disadvantages:

- Tend to overlook low level components
- No detailed basis

4. Bottom-Up

The cost of each software components is estimated first and then the results are combined to derive the final cost estimation for the project. An example is the COCOMO's detailed model.

Advantages:

- More stable
- More detailed
- Allow each software group to hand an estimate

Disadvantages:

- May overlook system level costs
- More time consuming

5. Estimation by Analogy

This technique utilizes the actual data that is extrapolated from a previous completed project and compare that with the proposed project in the same application domain to derive the cost estimate.

Advantages:

- Based on actual project data

Disadvantages:

- Impossible if no comparable project had been tackled in the past.
- How well does the previous project represent this one

6. Price to Win

The cost estimate is the price that is necessary to win the contract or the project.

Advantages:

- Often rewarded with the contract

Disadvantages:

- Time and money run out before the job is done

MAINTENANCE

Over a period of time, the developed software system may need modifications according to the changing user requirements. Such being the case, maintenance becomes essential. The software maintenance process comprises a set of software engineering activities that occur after the software has been delivered to the user.

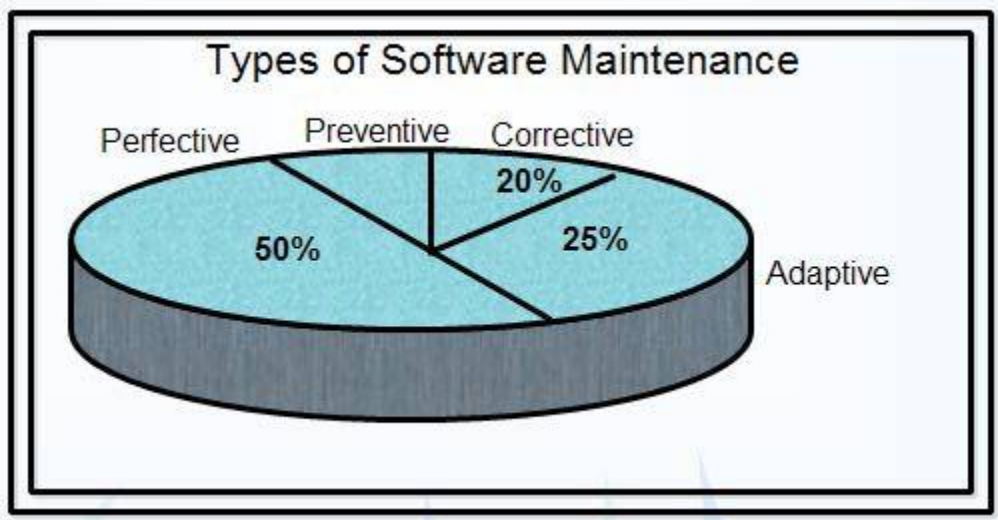
Sometimes, maintenance also involves adding new features and functionalities (using latest technology) to the existing software system. The primary objective of software maintenance is to make the software system operational according to the user requirements and fix errors in the software. The errors arise due to nonfunctioning of the software or incompatibility of hardware with the software. When software maintenance is to be done on a small segment of the software code, software patches are applied. These patches are used to fix errors only in the software code that contains errors.

Software maintenance is affected by several constraints such as increase in cost and technical problems with hardware and software. This chapter discusses how software maintenance assists the present software system to accommodate changes according to the new requirements of users.

❖ TYPES OF MAINTENANCE

There are four types of maintenance, namely, corrective, adaptive, perfective, and preventive. Corrective maintenance is concerned with fixing errors that are observed when the software is in use. Adaptive maintenance is concerned with the change in the software that takes place to make the software adaptable to new environment such as to run the software on a new operating system. Perfective maintenance is concerned with the change in the software that occurs while adding new functionalities in the software. Preventive maintenance involves implementing changes to prevent the

occurrence of errors. The distribution of types of maintenance by type and by percentage of time consumed.



- **Adaptive Maintenance**
- **Perfective Maintenance**
- **Preventive Maintenance**

1. **Adaptive Maintenance**

   Adaptive maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns, and government policies have a significant impact on the software system.

2. **Perfective Maintenance**

   Perfective maintenance mainly deals with implementing new or changed user requirements. Perfective maintenance involves making functional enhancements to the system in addition to the activities to

increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users' changing needs.

Examples of perfective maintenance include modifying the payroll program to incorporate a new union settlement and adding a new report in the sales analysis system. Perfective maintenance accounts for 50%, that is, the largest of all the maintenance activities.

3. **Preventive Maintenance**

Preventive maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understandability and increasing software maintainability. It comprises documentation updating, code optimization, and code restructuring. Documentation updating involves modifying the documents affected by the changes in order to correspond to the present state of the system. Code optimization involves modifying the programs for faster execution or efficient use of storage space. Code restructuring involves transforming the program structure for reducing the complexity in source code and making it easier to understand.

Preventive maintenance is limited to the maintenance organization only and no external requests are acquired for this type of maintenance. Preventive maintenance accounts for only 5% of all the maintenance activities.

THANK YOU….

TOPIC: - Institute Administrator System

BY : Kirti Rathore [0827CS193D09]

Aleena Syed [0827CS193D09]

Navni Pandya [0827CS193D01]