BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 6

DCA3245

SOFTWARE PROJECT MANAGEMENT

Unit 4

Estimation and Budgeting of Projects

Table of Contents

1. INTRODUCTION

Dear student, in the last unit you studied about project planning. In this unit, you are going to study about project estimation and budgeting. Software cost estimation plays a vital role in project management. This is because the cost estimation helps to connect the general concepts and techniques of economic analysis of building the service with the software engineering that helps to implement the product. Also, the software cost estimation works as a tool to perform the cost-benefit analysis that will enable the organizations to take the decision to whether to build the service or buy it. Additionally, the cost estimation acts as the basis for good project management as we will be able to estimate the tasks involved in the project as part of cost estimation.

In this unit, we will highlight an introduction to cost estimation, cost estimation methods, comparison of cost estimation methods, COCOMO cost estimation models (Basic, Intermediate and COCOMOII). We are also going to discuss the estimation of software budgeting and its three methods like NPV, payback methods and ROI.

1.1 Objectives:

After studying this unit, you should be able to:

- ❖ *Discuss various issues in software cost estimation*
- ❖ *Explain the cocomo model of software development*
- ❖ *Describe software budgeting*

2. SOFTWARE COST ESTIMATION

Dear student, to implement a project we will require a software (application server, Data base etc.), hardware (computers, network, switch etc.), and the human resources. The bulk of the cost is involved with human resources when compared to the other two aspects. Hence, when we use the term cost estimation, it generally refers to the cost estimation of human resources and will be expressed in terms of person-months (PM).

As we all know, the cost of the project depends on the product or service that we are building. In other words the effort involved to implement a project depends on the nature and complexity of the work and the degree of knowledge we have on the end product to be developed. When we initiate the project and working on the feasibility study of the project, we will have only some idea on the product being built. As we progress through requirements and gather more information the uncertainties around the functionality to be developed reduces and we will be in a better position to accurately specify the system. Despite this fact that the specifications being unclear initially, the cost estimation will be taken up during the initial stage of the project as over the time the cost estimation models have matured significantly so that we can estimate the project with fair degree of accuracy.

In the last section, we saw that the project planning is the key activity that needs to be carried out as initial part of the project. As part of the planning, the project manager and the key team members brainstorm to estimate the amount of work to be carried out. This effort estimation is essentially cost estimation as the effort directly translates to the cost of the project. Using the tasks-efforts listed out the project manager can put the time lines for each activity; allocate the resource come up with schedule as part of the project planning. We have to be aware that there will be some change in the effort estimated because as we gather more information on requirement we will have clearer picture of the end product.

According to Frederick Brooks, estimation is science as well as art. The estimation has to be carried out in an organized way. The past learning from the previous projects executed at the organization and the metrics collected over the time will help immensely during the cost estimation. Also the experience of the team members that worked on similar projects earlier will help to arrive at accurate cost estimations. As we stated earlier, estimation becomes foundation for the future set of activities to be carried out in the project and sets the direction for successful implementation of the project. Hence it is very important that we carry out proper estimation before embarking on the project.

Factors Affecting Software costs:

The vast majority of projects on the market are projects completed based on a fixed price cooperation (when the budget and terms are planned in advance, at the stage of sealing the deal).

- When evaluating a project, the team, besides standard risks and problems, must take into account the "modern and effective" approach of customers who want to combine:
- Flexibility on the part of the development team, the implementation of all customer requirements that appear during the project (since the customer often doesn't know what they want until the middle of the project
- The availability of skilled personnel and their allocation to the project can impact the overall cost. Scarce resources may require higher salaries or longer development times.
- The project management methodology chosen, such as Waterfall, Agile, or a hybrid approach, can influence how resources are allocated and how development progresses, affecting costs.
- Obtaining accurate estimates on the budget and terms before writing a technical task, including them in the contract, and then, during the implementation of the project, strictly controlling the budget and terms;

The three main Factors Affecting the Software cost estimation are as follows:

- Type of Software Project
- Size of Software Project
- Development Team Size

1. Type of Project:

Each of these types of projects typically has a different team makeup and requires a different amount of development effort. Understanding the type of project is the first step in developing a cost estimate. This information will be used in combination with the size of the project and the project team to determine the final estimate.

- From a high level, typical software development engagements tend to break down into the following types:

- New Software Development – new software, involving custom development.
- Software Modification – Enhancement of existing software.

2. <u>Size of Software:</u> Size is a bit of a gut call. There tends to be a tight correlation between the complexity of a project and its size, but that isn't always the case. Generally, project sizes fall into the following categories
   - Small : A small project usually involves minor changes. Typically, things like tweaks to the user interface or bug fixes that are well defined with a known cause. Interaction with the client is limited, i.e. "Is this what you want to be done?" followed up by, "Here is what we did.."
   - Large Project: Large projects may require integration with multiple systems, have a database component, and address security and logging features. An underlying framework and a module-based design are common, taking into consideration scalability and maintainability.

<u>Classes of Cost Estimation:</u>

Estimating the cost involved in software development has been broadly divided into three classes based on the percentage that the project has been defined so as to easier the work to categories ,

Classes of cost estimates are as follows:

- Estimate can be developed when the project is defined by 10-40%, it serves to authorise and control budgets
- Estimate can be obtained when the project is defined by 30-70%, it supports cost control activities and bidding
- Estimate can be received when the project plan is mature by 65-100%, it helps verify cost estimates or design compelling bids

Estimate can be received during the initial phase of project planning, Its only aim is to screen and evaluate different project and cost concepts Estimate can be obtained when the project plan is defined by 1-15%, Its primary function is to analyse performance feasibility

In some cases, especially for business or investment projects, preliminary estimates of the potential benefits or returns on investment may be provided during the initial planning phase. These estimates involve identifying key communication points, channels, and stakeholders. They help in establishing effective communication plans for the project.

Estimate Credibility Factors

Estimating the credibility factors serves best for Construction and architecture, marketing, but works for other industries too. You may use Gantt charts, flow charts, spreadsheets, or lists to show the hierarchical outline of importance and connectivity between the tasks needed to complete the project. Ensure no team member carries the majority of the project's weight by spreading duties and responsibilities across the team.

Follow these steps in order to attain the creditability factors:

1. Form a clear picture of the project's constraints, characteristics, conditions and tasks to have a realistic view of its financial requirements
2. In case of unsuccessful project management (if the team follows the client's lead), the team easily passes the deadlines and budget, because the contract is signed and the budget is agreed upon, thus they just work at a loss.
3. Involve employees from different organisational levels into the estimation process to ensure information applied during decision making is complete and reliable
4. It is clear that the client is the only one to blame. The matter is that the estimated software project cost is often figured out without sufficient analysis of requirements, tasks are insufficiently and incorrectly scheduled, and very often, only programming is included in the estimate, while testing and management don't get proper attention
5. Utilise relevant and high quality data when estimating cost and Implement a standardized and structured approach to estimation
6. Take into account risks, uncertainties and possible environmental changes, including cost escalation
7. Implement a standardized and structured approach to estimation
8. Recognise excluded costs and make sure you have a good reason for not including them in your project budget and Conduct an independent review of final estimates to verify their accuracy.
9. Always update forecasts in case your project plan and design have been subject subjecto any modifications

Adaptive Project Framework (APF): The Adaptive Project Framework (APF) is a method of project management that excels in situations where there is a lot of room for uncertainty, shifting requirements, and improvisation. When conventional approaches to managing projects encounter difficulties in adapting to new or changing conditions, APF can help. The Adaptive Project Framework is characterised by the following characteristics and guiding principles:

- Iterative and Incremental Approach: APF is built around iterative cycles, where work is divided into smaller increments or iterations. Each iteration results in a functional piece of the project, allowing for early feedback and adjustments. The project's scope can evolve based on this feedback.
- Project Scope: APF recognizes that uncertainty and change are natural aspects of many projects. It encourages project teams to embrace learning and discovery throughout the project lifecycle. As the team gains insights and new information, they can adapt their approach accordingly.
- Cycle plan: APF promotes close collaboration with customers, stakeholders, and end-users. This collaboration helps ensure that the project's outcomes align with the evolving needs and expectations of the stakeholders.
- Cycle build : Instead of having a fixed, detailed project plan upfront, APF emphasizes continuous planning. Plans are updated and adjusted as the project progresses, taking into account new information and changing circumstances.
- Client checking Point: APF allows for changes to the project scope, provided they align with project objectives and stakeholder needs. As new requirements emerge or priorities shift, the project scope can be adjusted accordingly.
- Final Review: emphasizes delivering functional pieces of the project at the end of each iteration. This provides tangible results early in the project and helps validate assumptions and decisions.

Best Practices for software estimation: Software cost estimation is a critical aspect of project management in software development. Accurate cost estimation helps ensure that projects are properly budgeted and managed, preventing overruns and financial issues. Here are some best practices to consider when performing software cost estimation:

- understand what needs to be done to achieve the project goal and deliver in the best possible way;
- identify as many requirements and constraints for the project as possible;
- test the received requirements;
- in order to determine how much does it cost to develop a software program, involve all relevant professionals

- test the received requirements;
- in order to determine how much does it cost to develop a software program, involve all relevant professionals

- Utilize a combination of estimation techniques to cross-validate results and gain a more accurate estimate. Common methods include expert judgment, analogous estimation, parametric estimation, and bottom-up estimation.
- Break the project down into smaller work packages or tasks. This allows for more granular estimation, making it easier to assess the effort and resources required for each component.

<u>Parameters- which effects the Costs of Developing Software</u>

To achieve the goal of the project, break it down into custom actions, which break down into tasks, which break into subtasks, etc. And so on until each task becomes understandable to a junior specialist level person and has clear criteria for how to check its implementation. The following are some of the parameters where the costs involves in developing the software:

- Creation of the installation utility.
- Creation of a configuration utility (there may be several of them: initial configuration, a configuration of system parameters, security configuration).
- Creation of a utility for initial data filling and, possibly, a utility for migration to a new version.
- Creation of diagnostic utilities (utilities that will help you check if everything is installed correctly and help you troubleshoot).
- Creation of a logging module (logging). Even if the customer does not need it, it will greatly help to identify errors and shortcomings.
- Creating a security module.

Techniques for Estimation: There are as many ways to estimate the cost and timeline of a software project as there are development languages. Research braas shown that if the project can broken down into small chunks of work, and each chunk estimated, the estimates tend to be more accurate. Involving stakeholders early in the software estimation process helps to define more accurately what is important in the software development cycle. Aids both the business leaders and the technology team gain a shared understanding of the project, it also helps to hold everyone involved accountable to the initial estimate

Estimation of the benefit of the project:

To find out weather the project is beneficial or not we have to follow the below steps:

- Identify benefits: determining and categorizing a business's or project's benefits and the people who will be in charge of handling them
- Execute benefits management: overseeing the management of benefits to avoid risks and find new opportunities
- Sustain benefits realization: monitoring the performance of a project's benefits and ensuring they're valuable even after implementation

Estimation for efforts:

The first part of pricing comes down to how much effort is needed to achieve the desired outcome. i.e. how many engineers and how many of their hours per day will be required to get the job done. Remember that effort estimation is not an exact science, and there will always be some level of uncertainty. As the project progresses, continually monitor and compare actual effort expended to your estimates, and use the feedback to refine your estimation process for future projects.

- The ability of a client to dedicate staff to work with the project team for requirements analysis, design checks and user testing
- The ability of a client to dedicate staff to work with the project team for requirements analysis, design checks and user testing
- What does it take to get database or system access, Is this a quick call to a DBA, or is there an approval process that has to get committee approval?
- How easy is it to get firewall changes
- What needs to be done to get a cloud-based solution approved

<u>Using the project Size and the Project Type we can determine the project time:</u>

We can determine the time estimation required to complete the project based on a parameter called project type:

1. Project type: Now that project types and sizes are defined, they can be combined to put together the following possible timeframes:
2. Project Size: The involvement of the resources describes the project size

For larger, complex projects – team resources usually fulfill only one role to effectively move the project forward.

- Once the project is defined in terms of type and size, the next factor to be determined is the team size.
- Every project requires at least 3 roles – a Project Manager, a Developer, and a QA Tester.
- In a small/medium project, the Project Manager may also fulfill the role of Business Analyst, and so forth.

<u>Following are the steps involved in cost estimation.</u>

Step 1: Establish Objectives

- Document the objectives of the estimation process. This will be helpful in the decision making.
- Evaluate the estimation accuracy objective versus the components of the cost estimates and try to balance out between the two factors.
- As the estimation process progresses, review the estimating objectives and refine them as needed.

Step 2: Plan for Required Data and Resources

Consider the estimation activity itself as a mini project to be carried out. This will ensure that we will then do the proper planning for the estimation process and collect proper data and allocate appropriate resources to carry out the task. Bottom-line is that we need to ensure that the estimating process is carried in an orderly fashion noting down the answers for typical questions like why, who, what, where, how, how much etc. for our estimation activity.

Step 3: Pin Down Software Requirements

It is important that the requirements to be unambiguous to arrive at accurate estimates. Hence, to arrive at proper estimates we need to analyze the requirement repeatedly to clear out the uncertainties around the functionality to be estimated. A measure of how accurate the requirement specification can be obtained by looking at the extent the specification can be tested. If we can clearly specify a pass/fail condition for a test case of a specification, then for such specification we can have accurate estimates.

Step 4: Work Out as Much Details as Feasible

In step 1, we established the objectives of estimation that included the objective on accuracy as well. Using the accuracy objective as guideline, try to work out as much detail as feasible with available resources and timeline for the estimation process. If we have more details on the task we are estimating, it will help in more accurate estimates because of following reasons: a) the more detail we explore, the better we understand the technical aspects of the software to be developed; b) the more pieces of software we estimate, the variance in estimates will be reduced by corresponding factor; c) the more we think through all the functions the software must perform, the less likely we are to miss the feature to be accounted for.

Step 5: Use Several Independent Techniques and Sources

There are several estimation techniques available to carry out the cost estimation process. Each will have their own strengths and weaknesses and will be suitable for certain kind of the projects. We will have to study the different techniques available and follow the process that suites best for our project by getting the inputs from different sources.

Step 6: Compare and Iterate Estimates

If we have multiple independent techniques available for estimation, we need to carry out the estimation using these independent techniques. By this we will have the advantage of comparing the results of estimation using these techniques and carry out investigation on the reason for difference in numbers. This comparison and iterative estimation helps to arrive at more realistic estimates.

Step 7: Follow-up

Once the project gets underway, it is important to monitor the project progress and gather metrics on the actual cost incurred. It is worth comparing the cost against the originally estimated cost as it would give us the knowledge on estimating inputs that needs to be improved upon. The estimates might have been inaccurate because of improper sizing or cost drivers or some other parameters used for costing. Having regular follow up on the actual costing against the estimated cost give us the knowledge on actual cost drivers to be used, sizing that had to be applied and so on. This learning can be useful in implementing proper estimation of future projects in the organization.

Software development is a volatile activity in the sense we can expect a continuous change in the system as the components are added, re-scoped, and re-designed during the course of the project. Hence the project manager will have to be vigilant to identify these changes and generate a more realistic estimate to accommodate such changes. Also, as we know software cost estimation is evolved by studying the trends observed in previous projects and calibrated on past data. Hence the estimation needs continuous observation and needs regular updates from time to time as per the project data collected over the time. For this purpose we will have to compare the actual costs versus the estimated costs and look for opportunities to improve on the estimation techniques.

Software Cost Estimation Methods

Several popular methods are available to carry out the software cost estimation.

Delphi Method : The Delphi method is an interactive forecasting technique carried out by a group of experts. During a series of meetings, each member of the expert panel provides their cost forecasts in the questionnaire format. The answers are then analyzed and anonymously announced to others by a facilitator. This activity is repeated for two or more rounds to provide experts with an opportunity to adjust their forecasts each time until the ultimate goal of the Delphi method – consensus on a final cost estimate – is attained

Algorithimic Models

As the name suggests this method follows the cost estimation in the form of a function that expects a number of variables as input. These input variables will generally be the software metrics to know the sizing and the cost drivers.

Expert Judgement

This method involves getting the estimation carried out by a group of experts in the relevant field of the project being carried out. Group of experts carry out the estimation independently, and later the results are compared and re-worked on until consensus on the estimates are reached. This method is also known as Delphi technique.

Analogy Estimation

In this method, we try to look for one or more similar projects that are of the same nature of the current project being estimated. By using analogy of the past work the estimation is carried out for the current project to be implemented.

Top-down Estimation

In this approach, the project feature to be implemented as a function and estimated based on the size of the function to be implemented.

Bottom-up Estimation

In this approach, the project is divided into sub-tasks and estimated separately for each sub task. The estimate of the overall project then will be the aggregate of efforts estimated for each of the sub tasks.

Parkinson's Principle

Parkinson’s principle states “Work expands to fill the available volume”. This principle will be useful in cost estimation as well as allocating the resources to the tasks

Price to Win

This method solely depends on the pre-determined price that needs to be quoted to win the project. In other words, for a software project to be won we might be thinking that the price should be under some limit. Hence, the estimation will be done to in order to reach a number as determined earlier and not as per the software functionality to be developed.

3-point Estimation

The final project cost value is then calculated by using the following equation, also known as the PERT formula:

[Optimistic Estimate + Pessimistic Estimate + (4 × Best-guess Estimate)] / 6

To calculate a probable project cost with the 3-point estimating technique, one needs to create 3 types of estimates:

- An optimistic estimate – the amount of money your team hopes to spend on project activities,
- A pessimistic estimate – the project's cost in the worst-case scenario when many things go not as expected,
- The best-guess estimate – a realistic figure, the amount of money the team will most likely spend on project activities

Reverse Analysis

As an estimation technique, reserve analysis focuses on factors of cost uncertainty and is primarily concerned with avoiding risks that may lead to cost overruns. The purpose of this method is to determine the size of two types of backup sums:

Contingency reserve – the amount of money allocated for mitigation of various expected project risks (e.g., delays in supply, increased staff turnover, technological holdbacks, etc.).

Management reserve – the amount of money that could be utilized to tackle the outcomes of unidentified risks.

Cost of Quality Analysis

The purpose of the cost of quality (CoQ) analysis if to figure out how much money is needed to meet the preset project quality standardsThe CoQ calculation process comprises the following phases:

- Define project quality requirements and standards;
- Identify the cost of good quality (CoGQ): cost of activities needed to prevent quality failures + costs of quality appraisal activities;
- Identify the cost of poor quality (CoPQ): expenses the project may incur as a result of internal quality failures (e.g., product re-design, waste of time and material resources, etc.) + expenses the project may incur as a result of external quality failures (e.g., customer complaints, shipping errors and damages, etc.);

The received CoQ figure is then used to inform fund allocation decisions in a way that minimizes the risk of financial loss and helps sustain the optimal level of product and service quality.

Conclusion:

As you can see, 10 techniques reviewed in this article produce cost estimates of different accuracy levels. However, if an estimation method is associated with only vague and imprecise calculations, it doesn't mean it should be discarded right away. Estimation methods are techniques used in various fields, such as project management, engineering, and finance, to calculate or predict values based on available data or information. The choice of estimation method depends on the specific project, the availability of data, and the level of accuracy required. Many projects use a combination of these methods to arrive at a more robust estimate.

- Even inaccurate cost estimates are useful when you're still planning your project. Besides, approaches used for their creation are usually easy to implement. Estimation methods assist in resource allocation, ensuring that the right people, materials, and equipment are available when needed. This prevents delays caused by resource shortages. Estimation methods include contingency planning, allowing for the allocation of reserves to manage risks and uncertainties. This proactive approach helps mitigate the impact of unforeseen issues on the project's budget.
- Thus, lower-class estimation techniques will readily support you in the evaluation of multiple cost variants and scenarios. With their help, you can generate and assess new ideas, perfecting the overall strategic and tactical picture of your future project and gathering all the information needed to arrive at more accurate forecasts at later stages in the project development process
- generate and assess new ideas, perfecting the overall strategic and tactical picture of your future project and gathering all the information needed to arrive at more accurate forecasts at later stages in the project development processEstimation methods guide project planning by helping project managers anticipate resource needs and develop realistic project schedules. Effective cost estimates help organizations optimize resource utilization by preventing overallocation or underutilization of resources.

The comparison of different cost estimation methods have been shown in the Table 4.1.

Table 4.1: Comparison of Methods

Method	***Strengths***	***Weaknesses***
Algorithmic model	• Objective, repeatable, analyzable formula • Efficient, good or sensitivity analysis • Objectively calibrated to experience	• Subjective inputs • Assessment of exceptional circumstances • Calibrated to past, not future
Expert judgmen	• Assessment of representative ness, interactions, exceptional circumstances	• No better than participants • Biases, incomplete recall
Analogy	• Based on representative experience	• Representative ness of experience
Parkinson & Price to win	• Correlates with some experience • Often gets the contract	• Reinforces poor practice • Generally produces large overruns
Top-down	• System level focus • Efficient	• Less detailed basis • Less stable
Bottom-up	• More detailed basis • More stable • Fosters individual commitment	• May overlook system level costs • Requires more effort

Cost Estimation Guidelines

The following guidelines will help you in cost estimation.

1) Let the developers carry out the initial round of estimation. This will help the developers to gain better understanding of the system.

2) Perform proper reviews on the estimations carried out and finalize the initial estimates only after thorough study.
3) Anticipate and control user changes.
4) Monitor the progress of the proposed project.
5) Evaluate proposed project progress by using independent auditors.
6) Use the estimate to evaluate project personnel.
7) Computing management should carefully and approve the cost estimate.
8) Rely on metrics and data available rather than on intuitions.
9) Don't rely on cost estimating software for an accurate estimate.

SELF-ASSESSMENT QUESTIONS – 1

1. Cost in a project includes software, hardware and human resources. (True / False)
2. Most cost estimates are determined in terms of ____________________.
3. ‘Work expands to fill the available volume’ is _________________ principle.
   a) Parkinson’s
   b) James’s
   c) John’s
   d) None of the above

Software Engineering Models:

All SDLC models can be structured into several groups depending on how they approach workflow organization – linearly or iteratively – and what kind of relationships are established between the development team and the customer. Software development life cycle (SDLC) models show the ways to navigate through the complex and demanding process of software building. A project’s quality, timeframes, budget, and ability to meet the stakeholders’ expectations largely depend on the chosen model.Today, there are more than 50 recognized SDLC models in use. None of them is perfect, and each brings its favorable aspects and disadvantages for a specific software development project or a team.

➢ The sequence is led by the types in the chart's lower quadrants. You can easily implement, utilise, and manage them. As you level up, the process loosens up and gives you more leeway to adapt to future software's changing needs. Different software development life cycle (SDLC) models provide various strategies for completing software development tasks. The project's peculiarities and the organization's requirements should be taken into account while deciding on a model. Many current development methods are hybrid techniques that pull features from other models to better fit individual projects.Based on whether they take a sequential or iterative approach to organising the workflow, as well as the nature of the interactions forged between the development team and the client, all SDLC models can be categorised into a number of types.

- The sequence is led by the types in the chart's lower quadrants. You can easily implement, utilise, and manage them. As you level up, the process loosens up and gives you more leeway to adapt to future software's changing needs.
- Models on the left of the chart have minimal customer involvement, while those on the right become increasingly "cooperative" and involve customers in various stages of the software development life cycle.

Waterfall model:

Each stage has concrete deliverables and is strictly documented.

Each stage has concrete deliverables and is strictly documented. The next stage cannot start before the previous one is fully completed. Thus, for example, software requirements cannot be re-evaluated further in development.

The Waterfall Model has been criticized for its inflexibility in handling changing requirements and the potential for lengthy delivery times. However, it is still used in situations where project requirements are well-understood and unlikely to change significantly, such as in certain government projects, manufacturing, and highly regulated industries where documentation and traceability are critical. Additionally, some modern development methodologies, such as the V-Model and some forms of agile development, have been influenced by the Waterfall Model and incorporate its structured approach with adaptations to improve flexibility and adaptability.

V-Model:

The V-Model, also known as the Validation and Verification Model or Vee Model, is a software development and testing methodology that emphasizes the relationship between each development phase and its corresponding testing phase. It is an extension of the traditional Waterfall Model and is often used in industries where rigorous testing and quality assurance are critical, such as aerospace, healthcare, and defense. The name "V-Model" is derived from the shape of the diagram that illustrates the model, with the development phases on the left side and the testing phases on the right side, converging toward the delivery phase.

Such workflow organization implies exceptional quality control, but at the same time, it makes the V-model one of the most expensive and time-consuming models.

Moreover, even though mistakes in requirements specifications, code and architecture errors can be detected early, changes during development are still expensive and difficult to implement.

As in the Waterfall case, all requirements are gathered at the start and cannot be changed

Incremental and Iterative Model:

Incremental and Iterative models are software development methodologies that focus on delivering a system in smaller, manageable parts over time rather than in one large, single release. These approaches allow for flexibility, adaptability, and the opportunity to gather user feedback early in the development process. While they share similarities, there are distinct differences between the two models:

Both Incremental and Iterative models offer advantages in terms of adaptability, early feedback, and risk management. The choice between the two depends on project requirements, constraints, and the desired balance between delivering partial functionality early (Incremental) or refining the entire system in iterations (Iterative). Some projects may even combine elements of both models to suit their specific needs.

After qualified suppliers return the filled RFPs, managers should analyze them to identify a possible project cost range and decide on final estimates.

The development process based on the Incremental model is split into several iterations ("Lego-style" modular software design is required!).

New software modules are added in each iteration with no or little change in earlier added modules. The development process can go either sequentially or in parallel.

Scrum:

Scrum is probably the most popular Agile model. The iterations ('sprints') are usually 2-4 weeks long and they are preceded with thorough planning and previous sprint assessment. No changes are allowed after the sprint activities have been defined. Scrum is a popular agile framework for managing and delivering software and other complex projects. It is characterized by its iterative and incremental approach, strong emphasis on collaboration, and adaptability to changing requirements. Scrum provides a structured framework for teams to work together to develop high-quality products efficiently.

Scrum is known for its adaptability, as it allows for changes to be made between sprints based on user feedback and evolving requirements. It promotes transparency, inspection, and adaptation as core principles for managing complex projects. Scrum is widely used in software development but can also be applied to various other fields and industries to improve project management and product development processes.

3. COCOMO MODEL

COCOMO stands for Constructive **CO**st **MO**del. This is a very popular technique of estimation that is used in software development. COCOMO is developed by Barry Bohem of TRW and was first published by Bohem in 1981. This technique estimates the effort and time needed for the project relating it to the size of input and cost drivers. The cost drivers are the parameters that affect the productivity of the project. The below equation (Equation 1) provides the basic effort calculation in COCOMO model. The other specific measurements like the estimates related to requirement, support etc. are derived from the base calculation as derived in Equation 1.

$$PM = C \times (KDSI)^n \quad ------- \quad \textbf{Equation 1}$$

Here

PM represents the effort in Person-Month

C represents a constant,

KDSI is thousands (Kilo) of "delivered source instructions" (DSI), and **n** is a constant.

COCOMO model is typically defined in three different models. They are the **Basic Model**, the **Intermediate model,** and the **Detailed model.** The complexity of models increases with the number of parameters each model takes into account that influences the project while computing the effort estimation. Further, the factors considered for the cost estimation depends on the mode at which the development is done. The development modes are classified as **Organic mode**, **Semidetached mode,** and the **Embedded mode**.

Organic mode – refers to the projects that are developed in a familiar and stable environment, and the nature of the work involved in the product development is similar to the one that the company has previously worked. Further the quantum of the work is small and requires no major innovation.

Embedded mode – This refers to complex projects that are having a tight schedule, stringent constraints and having complex interfaces. Also the team will have to put in lot of innovation in these kinds of projects.

Semidetached mode - In this mode the project complexities falls in between the characteristics of Organic and Embedded modes of the project.

The COCOMO model estimates the cost of the project only in terms of Person-Months. The cost estimation in dollars is avoided as the labor cost has lot of variation across organizations and also across different geographical settings. Also the Person-Month estimation can be conveniently converted to Currency level by applying base labor cost prevailing as per the company set up taking into account all the dynamic parameters like salary, inflation, competition etc. as applicable.

The Basic COCOMO Model

The basic COCOMO model performs the estimation of software development effort by considering only the size of the software in DSI and the three different modes of development. It is a quick and rough way of estimation. The accuracy of the estimation is not so great. Only about 29% of the time the cost estimations will be within the range of 1.3 with the actual cost of the project. It is primarily because the cost factors like hardware constraints, resource personnel quality, tool usage etc. are not considered for the estimation.

Table 4.2 shows the Effort Equations for the three development modes in Basic COCOMO Model. Here, the effort corresponds to the effort in Person-Months and TDEV corresponds to the Time for development of the software product.

Table 4.2: Effort Equations for Three Development Modes in

Development Mode	**Basic Effort Equation**	**Basic Schedule Equation**
Organic:	**Effort = 2.4 x $(KDSI)^{1.05}$**	**DEV = 2.5 x $(Effort)^{0.38}$**
Semidetached:	**Effort = 3.0 x $(KDSI)^{1.12}$**	**DEV = 2.5 x $(Effort)^{0.35}$**
Embedded:	**Effort = 3.6 x $(KDSI)^{1.20}$**	**DEV = 2.5 x $(Effort)^{0.32}$**

The Intermediate COCOMO Model

The Intermediate model uses an **Effort Adjustment Factor (EAF)** and different set of coefficients for the effort equations when compared to the Basic model. We can apply Intermediate COCOMO across the entire software product for easily and roughly cost estimation during the early stage, or apply it at the software product component level for more accurate cost estimation in more detailed stages. The Intermediate COCOMO estimates are within 20% of the actual 68% of the time. Table 4.3 shows the Effort Equations for the three development modes in Intermediate COCOMO Model.

Table 4.3: Effort Equations for Three Development Modes in

Development Mode	Basic Effort Equation	Basic Schedule Equation
Organic:	Effort = EAF x 3.2 * $(KDSI)^{1.05}$	TDEV = 2.5 x $(Effort)^{0.38}$
Semidetached:	Effort = EAF x 3.0 * $(KDSI)^{1.12}$	TDEV = 2.5 x $(Effort)^{0.35}$
Embedded:	Effort = EAF x 2.8 * $(KDSI)^{1.20}$	TDEV = 2.5 x $(Effort)^{0.32}$

EAF = Effort Adjustment Factor, when the EAF is 1, we call the Effort is nominal.

The Intermediate model produces better result of estimation because along with the size, the model considers 15 cost drives while computing the effort. These cost drivers are grouped into four categories: software product attributes, computer attributes, personnel attributes, and project attribute. These cost parameters take into consideration the factors that will influence the project during its implementation stage. The factors vary from issues like hardware constraints, limitations of the personnel employed, software tool usage, etc. The EAF is the product of the effort multiplier corresponding to each of the cost factors. Additional advantage with the Intermediate model over the Basic model is that we can estimate the cost for individual component like staffing, cost, duration instead of system as a whole.

The Detailed Model improves on the Intermediate model. It encompasses all the characteristics of the intermediate model, and additionally includes the assessment of the cost driver's performance on each phase (analysis, design etc.) of the software development process.

Introduction to COCOMO II

For the traditional software development the original COCOMO model suited very well and also it was most widely accepted across the software industry. However for the modern software development practices the original COCOMO estimation model's cost drivers and effort adjustment factors are not very well adaptable. Hence it has evolved into a more comprehensive estimation model, called COCOMO II. COCOMO II developed based on changes in software development practices of the 1990s and 2000s, and will continue to evolve over the next few years.

The primary objectives of the COCOMO II effort are::

- To develop a software cost and schedule estimation model tuned to the life cycle practices of the 1990's and 2000's.
- To develop software cost database and tool support capabilities for continuous model improvement.

COCOMO II is actually a hierarchy of estimation models that address the following areas:

Application Composition Model: Used during the early stages of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount.

Early Design Stage Model: Used once requirements have been stabilized and basic software architecture has been established.

Post-architecture-stage Model: Used during the construction of the software. Like all estimation models for software, the COCOMO II models require sizing information. Three different sizing options are available as part of the model hierarchy: object points, function points, and lines of source code.

SELF-ASSESSMENT QUESTIONS – 2

4. COCOMO model was developed by Parkinson. (True / False)
5. COCOMO stands for ______________.
6. In _________ mode the project is characterized by tight, stringent constraints and interface requirements.
   a) Organic
   b) Embedded
   c) Semidetached
   d) None of the above

4. BUDGETING

In business terminology, the Budgeting of a company means the planned allocation of available funds and resources in each department of that company. Budgeting is aimed at controlling the spending of the company. The goal of budgeting is that overspending on less productive areas of the company is curbed and company assets and resources are invested on more productive areas that generate more income and better public relation to the company. Budgeting is handled by executives of the company comprising of Accountants, senior managers, experts in financial domain, and the representatives of each department of the company. Software Budgeting is also similar activity planning the spending of the software organization.

Capital Budgeting

Capital Budgeting or the investment appraisal is used to shape the company's long term investments such as the new products to be released, new projects to be worked on, new domains to be serviced, new technology to be adapted, new researches to be undertaken etc.

Many formal methods are used in capital budgeting, including the techniques such as

- Net present value
- Profitability index
- Internal rate of return

- Modified Internal Rate of Return, and
- Equivalent annuity.

These methods use the incremental cash flows from each potential investment, or project. Simplified and hybrid methods are used as well, ***such as payback period and discounted payback period.***

1) Net Present Value (NPV)

Net present value (NPV) that measures the present value of net cash flow is a standard method for the financial appraisal of long-term projects. This is used for capital budgeting, and widely throughout economics, it measures the excess or shortfall of cash flows, in present value (PV) terms, once financing charges are met. The below equation gives the expression for evaluating the NPV

Definition: NPV

The net-present-value (NPV) method is a discounted-cash-flow approach to capital budgeting that computes to all expected future cash flows using a minimum desired rate of return.

Formula:

Each cash inflow/outflow is discounted back to its PV. Then they are summed. Therefore

$$\mathrm{NPV} = \sum_{t=1}^{n} \frac{C_t}{(1+r)^t} - C_0$$

Where

t – the time of the cash flow

n– the total time of the project r – the discount rate

C_t – the net cash flow (the amount of cash) at time t.

C_0 – the capital outlay at the beginning of the investment time (t = 0)

In the above formula, choosing an appropriate discount rate is crucial to the NPV calculation. A good practice of choosing the discount rate is to decide the rate which the capital needed for the project could return if invested in an alternative venture. If, for example, the capital required for Project A can earn five percent elsewhere, use this discount rate in the NPV calculation to allow a direct comparison to be made between Project A and the alternative.

Obviously, NPV value obtained using variable discount rates with the years of the investment duration is more reflecting to the real situation than that calculated from a constant discount rate for the entire investment duration.

With a particular project, if C_t is a positive value, the project is in the status of cash inflow in the time of t. If C_t is a negative value, the project is in the status of cash outflow in the time of t. Appropriately risked projects with a positive NPV should be accepted. This does not necessarily mean that they should be undertaken since NPV at the cost of capital may not account for opportunity cost, i.e. comparison with other available investments. In financial theory, if there is a choice between two mutually exclusive alternatives, the one yielding the higher NPV should be selected.

Table 4.4 sums up the NPV's various situations.

Table 4.4: Decision Making based on NPV's Value

If...	*It means...*	*Then...*
NPV>0	the investment would add value to the firm	the project should be accepted
NPV<0	the investment would subtract value from the firm	the project should be rejected
NPV=0	the investment would neithergain nor lose value for the firm	We should be indifferent in the decision whether to accept or reject the project. This project adds no monetary value. Decision should be based on other criteria, e.g. strategic positioning or other factors not explicitly included in the calculation.

Example:

ABC Company is deciding on feasibility of introducing a new product line. Launching a new product involves initial startup expenses, operational expenses. The incoming cash flow is over 6 years. This product line introduction will have an immediate (t=0) cash outflow of

$50000 (the expenses towards machinery, new skilled resources, training costs etc.). Other cash outflows for years 1-6 are expected to be $2500 per year. Cash inflows are expected to be $15000 per year for years 1-6.

All cash flows are after-tax, and there are no cash flows expected after year 6. The required rate of return is 12%. The present value (PV) can be calculated for each year:
T=0 - $50000 / 1.12^0 = -$50000 PV.
T=1 ($15000 - $2500)/ 1.12^1 = $11160 PV.
T=2 ($15000 - $2500)/ 1.12^2 = $9965 PV.
T=3 ($15000 - $2500)/ 1.12^3 = $8897 PV.
T=4 ($15000 - $2500)/ 1.12^4 = $7943 PV.
T=5 ($15000 - $2500)/ 1.12^5 = $7092 PV.
T=6 ($15000 - $2500)/ 1.12^6 = $6332 PV.

The sum of all these present values is the **net present value,** which equals $1390. Since the NPV is greater than zero, the corporation should invest in the project.

2) Rate of return (ROR)

In finance, rate of return (ROR) or return on investment (ROI), or sometimes just return, is the ratio of money gained or lost on an investment relative to the amount of money invested. The amount of money gained or lost may be referred to as interest, profit/loss, gain/loss, or net income/loss. The money invested may be referred to as the asset, capital, principal, or the cost basis of the investment.

ROI is also known as **rate of profit. Return** can also refer to the monetary amount of gain or loss. ROI is the return on a past or current investment, or the estimated return on a future investment. ROI is usually given as a percent rather than decimal value.

ROI does not indicate how long an investment is held. However, ROI is most often stated as an annual or annualized rate of return, and it is most often stated for a calendar or fiscal year.

ROI is used to compare returns on investments where the money gained or lost – or the money invested – are not easily compared using monetary values. For instance, a $1,000 investment that earns $50 in interest obviously generates more cash than a $100 investment that earns $20 in interest, but the $100 investment earns a higher return on investment.

- \$50/\$1,000 = 5% ROI
- \$20/\$100 = 20% ROI

3. Payback Model

Payback time, or payback period, gives the measure of period it takes for a project to start making profit. It is expressed in terms of cash inflows to the project from the initial money that is invested into the project.

$P=I\div O$

Where I is the cash inflow and O is the cash outflow.

Example:

Consider a scenario in which the company buying a new machinery. The cost of machine is \$9000 and annual savings is about \$4500 in the cash outflows. The machine has useful life span of about 6 years. What would be the payback period in this case?

$P = I \div O = \$9000 \div \$4500 = 2$ years

SELF-ASSESSMENT QUESTIONS – 3

7. Budgeting in a business sense is the planned allocation of available funds to each department within a company. (True / False)
8. ROI stands for ______________.
9. ______________ is a standard method for the financial appraisal of long-term projects.

5. SUMMARY

Let's summarize the important points covered in this unit:

- The software project planner must estimate three things before a project begins: how long it will take, how much effort will be required, and how many people will be involved. In addition, the planner must predict the resources (hardware and software) that will be required and the risk involved.
- Accurate project estimates generally use at least two of the three techniques just noted.
- By comparing and reconciling estimates derived using different techniques, the planner is more likely to derive an accurate estimate.
- In this unit we have highlighted the COCOMO cost estimation models. It is the model used so frequently in the industry.
- We have discussed the estimation of software budgeting and its three methods like NPV, payback methods and ROI.

6. TERMINAL QUESTIONS

1. What is cost estimation? Explain different Cost Estimation methods.
2. Compare the Cost Estimation Methods in detail.
3. What is COCOMO? Explain COCOMO model in detail.
4. What is COCOMO II? Explain in brief.
5. What is budgeting? Explain the following budgeting techniques

   a) NPV b) ROI c) Payback Models

7. ANSWERS

Self Assessment Questions

1. True
2. Person-Months (PM)
3. a) Parkinson's
4. False
5. COnstructive COst MOdel
6. b) Embedded
7. True
8. Return On Investment
9. Net Present Value (NPV)

Terminal Questions

1. The project cost includes the requirements for software, hardware and human resources. The bulk of the cost of software development is due to the human resources needed, and most cost estimation procedures focus on this aspect. Most cost estimates are determined in terms of person-months (PM). (Refer Section 2)
2. A number of methods have been used to estimate software costs such as Algorithmic Models, Expert Judgment, Analogy Estimation etc. (Refer Section 2)
3. COCOMO stands for COnstructive COst MOdel, which is the most widely used software estimation model in the world. It was developed by Barry Boehm of TRW and first published in his book Software Engineering Economics in 1981. The COCOMO model predicts the effort and duration of a project based on inputs relating to the size of the resulting systems and a number of "cost drives" that affect productivity. (Refer Section 3)
4. The original COCOMO model became one of the most widely used and has evolved into a more comprehensive estimation model, called COCOMO II. COCOMO II is tuned to modern software life cycles. (Refer Section 3)
5. Budgeting in a business sense is the planned allocation of available funds to each department within a company. Budgeting allows executives to control overspending in less productive areas and put more company assets into areas which generate significant income or good public relations. (Refer Section 4)