

Chapter 2

(continue)

Feasibility Analysis :

- The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.
- Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system.
- **Types of Feasibility :**
 - Technical Feasibility
 - Operational Feasibility
 - Economic Feasibility
 - Legal Feasibility
 - Schedule Feasibility

- **Technical Feasibility :** Technical feasibility study gives report whether there exists correct required resources and technologies which will be used for project development. Along with this, feasibility study also analyzes technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc.
 - **Operational Feasibility :** Operational feasibility is operational scopes are determining usability of product, Determining suggested solution by software development team
-
- **Economic Feasibility –**
In Economic Feasibility study cost and benefit of the project is analyzed. Means under this feasibility study a detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost.
 - **Legal Feasibility –**
In Legal Feasibility study project is analyzed in legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc
 - **Schedule Feasibility –**
In Schedule Feasibility Study mainly timelines/deadlines is analyzed for proposed project which includes how many times teams will take to complete final project which has a great impact on the organization as purpose of project may fail if it can't be completed on time.

Cost-Benefit Analysis :

- Cost-benefit analysis compares the expected financial gain derived from a particular set of actions with the expected cost of providing each action to determine the most profitable option. The projected benefits of a plan or program are divided by its estimated total long-term cost.
- **Costs**
- Direct costs
- Indirect costs
- Intangible costs
- Opportunity costs
- Costs of potential risks

- **Benefits**

- Direct

- Indirect

- Total benefits

- Net benefits

- The determination of costs and benefit entails following steps :

1. Identify the costs and benefits pertaining to given project.
2. Categorize the various costs and benefits for analysis.
3. Select a method of evaluation.
4. Interpret the results of the analysis.
5. Take action.

1. Hardware cost –

Hardware cost includes actual purchase and peripherals (external devices) that are connected to computer. For example, printer, disk drive etc. Actually, finding actual cost of hardware is generally more difficult especially, when system is shared by various users so as to compared to a system which dedicated stand alone . In some case, best way is to treat it as operating cost.

2. Personnel costs –

Personnel costs includes EDP staff salaries and benefits as well as pay for those who are involved in process of development of system. Cost occurred during development of system which are one time costs and are also called development cost. Once system is installed, cost of operating and maintaining system becomes recurring cost that one has to pay very frequently based on requirement.

3. Facility cost –

Facility cost is amount of money that is spent in preparation of a site that is physical where application or computer will be in operation. This includes wiring, flooring, lighting and air conditioning. These costs are treated as one- time costs and are included into overall cost estimate of candidate system.

4. Operating costs –

These includes all costs associated with day-to-day(everyday) operation of system and amount depends on number of shifts, nature of applications. There are various ways of covering operating costs. One approach is to treat operating costs as an overhead. Another approach is to charge money from each authorized user for amount of processing they require from system. Amount charged is based on computer time or time they spend on system, staff time ad volume of output produced .

- **Supply costs –**

Supply cost are variable costs that increase with increased use of paper, disks and like. They should be estimated and included in overall cost of system.

- **Requirements modeling** is the process of identifying the requirements this software solution must meet in order to be successful. Requirements modeling contains several sub-stages, typically: scenario-based modeling, flow-oriented modeling, data modeling, class-based modeling, and behavioral modeling.
 - While technically there is no right way to go through the stages of requirements modeling, it typically begins with scenario-based modeling.
-

- **Scenario-based Model :**

Using a scenario-based approach, system is described from user's point of view. **For example**, basic use cases and their corresponding use-case diagrams evolve into more elaborate template-based use cases. There are three levels of elaboration.

Scenario-Based Modeling

- For the building of analysis and design models, it is essential for software engineers to understand how end users and other actors want to interact with the system
- Analysis Modeling with UML begins with the creation of scenarios in the form of use-cases, activity diagrams and swim-lane diagrams
- Use cases can be represented as a text narrative or as an ordered sequence of user actions or by using a template
- Example, consider the use case Access Camera Surveillance – Display Camera Views (ACS – DCV) of the Safe Home Security System

-
- **Class-based model :**
 - Class-based modeling begins by identifying the classes in the use case.
 - A collection of things that have similar attributes and common behaviors i.e., objects are categorized into classes. In addition to class diagrams, other analysis modeling elements depict manner in which classes collaborate with one another and relationships and interactions between classes.

7. Class-Based Modeling

- The section that follow present a series of informal guidelines as how do we go about developing the class-based elements of an analysis model:
 - classes and objects
 - attributes
 - operations
 - packages
 - CRC models, and
 - collaboration diagrams.

- **Behavioral elements :**

Effect of behavior of computer-based system can be seen on design that is chosen and implementation approach that is applied. Modeling elements that depict behavior must be provided by requirements model.

Behavioral Modeling

- The behavioral model indicates how software will respond to external events or stimuli. To create the model, the analyst must perform the following steps:
 - Evaluate all use-cases to fully understand the sequence of interaction within the system.
 - Identify events that drive the interaction sequence and understand how these events relate to specific objects.
 - Create a sequence for each use-case.
 - Build a state diagram for the system.
 - Review the behavioral model to verify accuracy and consistency.