Chapter 5. Acquisition, Development and Implementation of Information Systems

5.1 Introduction

5.2 Business Process Design

- Business process design means **structuring or restructuring** the tasks, functionalities and activities **for improvising a business system**.
- Business Process Design is a **critical step to understand the requirements** of the system.
- Business Process Design **needs a lot of intellectual capability** from team of developers doing the same.

Business process design involves a sequence of the steps, briefly explain each one of them.

5.2.1 Present Process Documentation

In this step, the present business process is **analyzed and documented** *(let's take an example of manual billing system in which the billing clerk checks the price list of products before s/he bills the same to customer. S/he checks the approved price list of the products as is applicable on the date of billing, checks for discounts and bills the products to customer.)*

The **key deliverable** of this step includes the **well-defined short- comings** of the present processes *(In the above process, the key issue being that the billing clerk needs to possess, applicable and authorized price list with him)* **and the overall business requirements**.

This step includes the following activities:

- **Understanding** the business and the objectives for which it exists.
- **Documenting** the existing business processes; and
- **Analysis** of the documented processes.

5.2.2 Proposed Process Documentation

This step is to **design the new process requirements** for the system*(say an automatic billing system in which system, every bill is generated through system. System has in its database of approved price list.)*
The design is **based on the new system requirements and the changes proposed.**
The activities include the following:

- **Understanding** of the business processes necessary to achieve the business objectives. ~~(flash back of stage1: **Understanding** the business and the objectives for which it exists)~~
- **Designing** the new processes; and
- **Documentation** of the new process, preferably using of CASE tools.

\

5.2.3 Implementation of New Process

This step is to implement the new as well as modified processes in the entity.

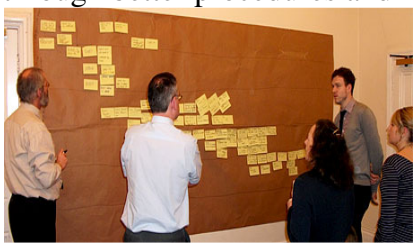
The critical activities may include the following:

- **Validating** the new process;
- **Implementing** the new process; and
- **Testing** the new process.

5.3 System Development

SYSTEMS DEVELOPMENT PROCESS

In business, **systems development** refers to the process of **EXAMINING** a business situation with the intent of improving it through better procedures and methods.


System development can generally be thought of as having two major components:

- **System analysis** is the process of **gathering** and **interpreting** facts, diagnosing problems *(identify problem)*, and using the information to **recommend improvements** to the system. *(Analysis include Preliminary investigation, requirement analysis)*
- **System design** is the process of **planning** a new business system or one to replace or complement an existing system. *(Design include design, development, testing, implementing, maintaining)*

ACHIEVING SYSTEM DEVELOPMENT OBJECTIVES

Nov 2003: Bring out the reasons as to why organizations fail to achieve their system development objectives.

Management related issues:

Management na toh support karta ha na hi system development me involve hota ha aur bolta ha muje strategic system bana k do.

1. Lack of senior management support and involvement in information systems development.

Developers and users of information systems **watch senior management** to determine which systems development projects are important **and act accordingly by shifting their efforts** away **from any project which is not receiving management attention.**

In addition, management can see that adequate resources, as well as budgetary control over use of those resources, are dedicated to the project.

2. Development of strategic systems.

Because strategic decision making is unstructured, the **requirements, specifications, and objectives**

for such development projects are **difficult to define.**

Developer Related Issues:

Ab devloper paresan bolta ha,mere pas kam jyada aur training kam ha toh muje standard method bhi nai pata software banana ka.

3. Overworked or under-trained development staff.

In many cases, systems developers often **lack sufficient education background**. Furthermore, many companies **do little to help their development personnel stay technically sound**. Currently in these organizations, a training plan and training budget do not exist.

4. Lack of standard project management and systems development methodologies.

Some organizations **do not formalize their** project **management and** system **development methodologies,** thereby making it very difficult to consistently complete projects on time or within budget.

User Related Issues:

Users ki needs badal rai ha but unko change nai lana,isliye vo log participate bhi nai karte, unko pucha participate kyu nai karte toh bole hum gavar ha hume training toh di nai ha naya system use kaise kare.

5. Shifting user needs.

User requirements for information technology are **constantly changing.**

When these changes occur during a development process, the **development team faces the challenge of developing systems whose very purposes might change** since the development process began.

6. Resistance to change.

People have a **natural tendency to resist change**, and information systems development projects signal changes.

7. Lack of user participation.

Users must participate in the development effort to **define their requirements, feel ownership for project success**, and work to resolve development problems.

User participation also **helps reduce user resistance** to change.

8. New technologies.

When an organization tries to create a competitive advantage by applying advance Information technology, it generally finds that attaining system development objectives is more difficult because **personnel are not as familiar with the technology.**

9. User training.

Users must be **trained to effectively** utilize the new system.

(Intentionally left blank)

5.3.2 System Development Team

Several people in the organization are responsible for systems development.


1. **Steering committee** consisting of a **group of key** Information Systems services **users that acts as a review body** for Information Systems **plans** and **applications development.**

2. A **project management team** generally **consists of** both **computer professionals and key users.**
3. **System analysts** are responsible to **determine user requirements, design the system and assist** in development and implementation activities.

4. In any systems organization, however, **systems designers** take a lead role during the design, development and implementation stages.

5. In end-user developed systems, the **end-user** is ultimately responsible for the system.

5.3.3 Accountants' Involvement in Development Work (RTP)

Most accountants are **uniquely qualified** to participate in systems development because they may be among the few people in an organization who can combine knowledge of IT, business, accounting, and internal control, as well as behavior and communications, to ensure that new systems meet the needs of the user and possess adequate internal controls.

They **have *specialized skills*** - such as accounting and auditing - that can be applied to the development project.
For example, an accountant might **perform** the **analysis of** a proposed system's **costs and benefits.**

An accountant can help in various related aspects during system development; some of them are as follows:

(i) Return on Investment (referred as ROI):

- This **defines the return, an entity shall earn on a particular investment** i.e. capital expenditure.
- This financial data is a prime consideration for any capital expenditure entity decides to incur.
- The important data required for this analysis being the cost of project, the expected revenue/benefit for a given period.
- The **analysis ideally needs to be done before the start of the development efforts** for better decision making by management.

For this analysis following data needs to be generated.

(a) Cost: This includes estimates for typical costs involved in the development, which are given as follows:

- *Development Costs:* Development Costs for a computer based information system include costs of the system development process, like salaries of developers.

- *Operating Costs:* Operating Costs of a computer based information system including hardware/software rental or depreciation charges; salaries of computer operators and other data processing personnel, who will operate the new system.

- *Intangible Costs:* Intangible Cost that cannot be easily measured. For example, the development of a new system may disrupt the activities of an organization and cause a loss of employee productivity or morale.

(b) Benefits: The benefits, which result from developing new or improved information systems that can be subdivided into tangible and intangible benefits. A post implementation analysis is also done to see how the system development effort has benefitted an organization. For example: A large oil company in public sector, few years back implemented an ERP system at a total cost of ` 100 crores. The calculated benefits from the project were ` 40 crores per annum. Above data gives an actual ROI of 40%, which is tremendous for any business. Same also tells the payback period is around 2.5 years.

(ii) Computing Cost of IT Implementation and Cost Benefit Analysis:

- For analysis of ROI, accountants need the costs and returns from the system development efforts.
- For correct generation of data, proper accounting needs to be done. Accountants shall be the person to whom management shall look for the purpose.

(iii) Skills expected from an Accountant:

- An accountant, being an expert in accounting field must possess skills to understand the system development efforts and nuances *(degree)* of the same.
- S/he is expected to have various key skills, including understanding of the business objectives, expert book keeper, and understanding of system development efforts etc.

5.4 Systems Development Methodology

A system development methodology is a **formalized, standardized; well- organized and documented set of activities** used to manage a system development project.

It refers to the **framework** that is **used to structure, plan and control** the **process of developing** an information system.

Write few characteristics of system development methodology.

1. The project is **divided into a number of identifiable processes**, and each process has a starting point and an ending point. Each process **comprises several activities**, one or more deliverables, and several management control points.

1 

2 

The **division of the project** into these small, manageable steps **facilitates** both **project planning and project control.**

2. Specific reports and other documentation, called **Deliverables must be produced periodically** during system development **to make development personnel accountable** for faithful execution of system development tasks.

3. Users, managers, and auditors are required to **participate in the project** which generally provides approvals, often called signoffs, at pre-established management control points. **Signoffs signify approval** of the development process and the system being developed.




4. The system must be **tested** thoroughly prior to implementation **to ensure that it meets users' needs.**

5. A **training plan is developed** for those who will operate and use the new system.

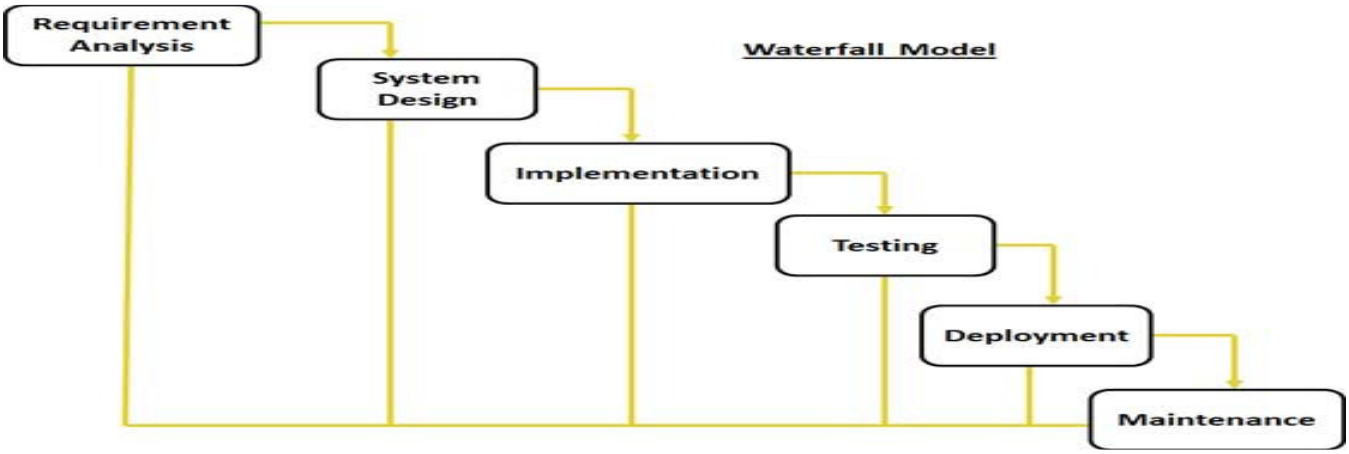
6. Formal **program change controls are established to preclude unauthorized changes** to computer programs. *(No image—Change Control means ek bar system implement hone k bad usme koi change nai honga)*

7. A **post-implementation review** of all developed systems must be performed to assess the effectiveness and efficiency of the new system and of the development process.

5.4.1 Waterfall Model/Traditional Development / Sequential Approch

- This is a traditional development approach in which **each developer work in different phases.**
- These phases include requirement analysis, coding, final testing and release.
- **Used only on small Projects because testing is delayed** until full system is developed, so problems cannot be detected in earlier phases.
- Here **activities are performed in sequence**.
- Also **when prior step is fully completed next activity commences.***(eg: relay race)*

Framework type: **Linear**



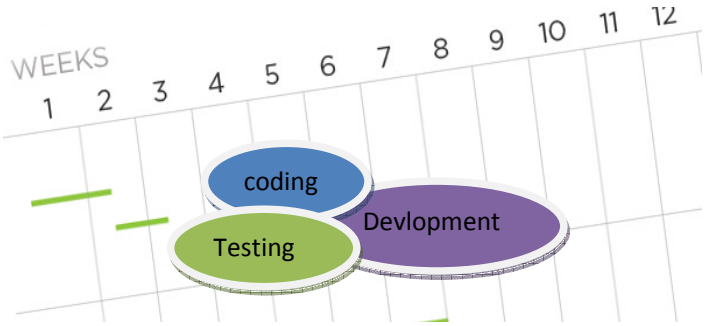
Waterfall Model

What are the characteristics of waterfall model, that have influenced development team to use this model?

OR

Basic Principles of waterfall model:

Code: CDE



1. Tight **C**ONTROL is maintained over the life of project through use of extensive written documentation, as well as through formal reviews and signoffs by users and management occurring at the end of most phases before beginning next phases.

2. Project is **D**IVIDED into sequential phases (eg: requirement analysis, coding, testing etc) with ***some overlaps****(I.e. Project progresses forward with a little movement backward.)* acceptable between phases.

3. **E**MPHASIS is on planning, time schedules, target dates, budgets and implementation of entire project at one time.

List out the benefits of waterfall model.

The fundamental strength of the waterfall model has made it **quite popular and handy among the fraternity***(organizations).*

Major strengths are given as follows:

Code: CPS Quality

1. **C**onserves Resources.
2. **P**rogress of system development measurable.
3. Ideal for **S**upporting less experienced project teams and project managers.
4. An orderly sequence of development steps and design review ENSURE QUALITY and reliability of developed software.

List few weaknesses of waterfall model:

Though it is highly useful model, it suffers from various weaknesses too. Experts and practitioners identify a number of weaknesses including the following:

4 Project:

1. **Inflexible, slow, costly and time consuming** due to tight controls.
2. Project **progress forward with slide movement backward**.

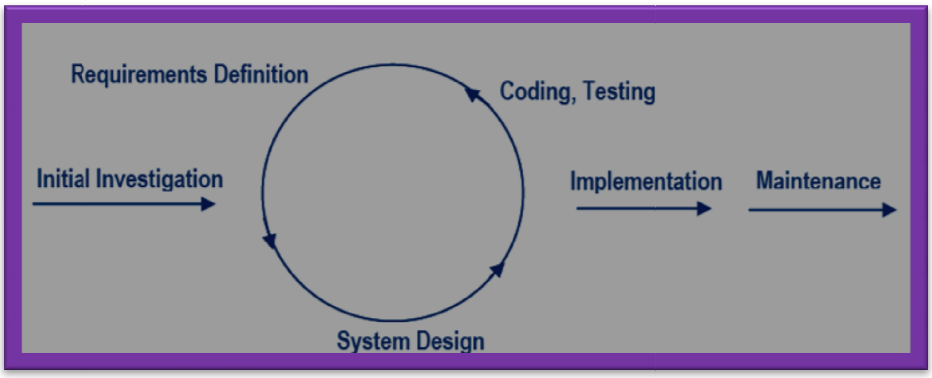
3. There is a **little to iterate**, which may be essential in situations.

4. Produces **excessive documentation** and keeping it updated as project progresses is time consuming.

5. ***USERS:*** Depend upon early identification of requirements, yet users may not be able to clearly define what they need early in the project.

6. ***DESIGN AND CODING:*** Requirement inconsistencies, missing system components and unexpected development needs are often discovered during **design and coding.**
7. ***SYSTEM TESTING:*** Problems are often not discovered until **final testing.**

5.4.2 The Prototyping Model



May 2000: Explain the prototype approach to system development along with its advantages.
May 2013: Goal of Prototyping approach:

- **Develop a small or pilot version** of a system or part of system.
- A prototype is usable system or system component that is built quickly and at a lesser cost, and with the intention of being modified or replacing it by a fully operational system.
- As users work with the prototype they make suggestions about the ways to improve it.
- This suggestion is than incorporated into another prototype, which is also used and evaluated and so on.
- **Finally when a prototype is developed that satisfy all user requirements**, either **it is refined** and turned into final **or scrapped**. If scrapped, the knowledge gained from building prototype is used to develop real system.

Framework Type: Iterative

Basic Principles: Prototyping can be viewed as a series of four steps:
Nov 2002: Describe the steps involved in prototyping for system development.

Step 1: Identifying the information system requirement:
The design team here identifies only **fundamental system requirement** ~~**(all requirements)**~~ to build the initial prototype.

Step 2: Develop the initial prototype:
- In this step designer creates initial base model and gives little or **no consideration to internal controls**.
- Emphasis is on system characteristics such as **simplicity, flexibility, ease of use.**
- **These characteristics enable users to interact with the tentative versions** of data entry screens, menus, and source documents.

Step 3: Test and Revise:
- After finalizing the initial prototype, designers first **demonstrate the model to users and then give it them to experiment and record their likes and dislikes** about the system and recommend changes.
- Using this feedback design team modifies the prototype as necessary and resubmits the revised model to system users for reevaluation.
- **This iterative model** of modification and reevaluation are **continued until user needs are satisfied.**

Step 4: Obtain user signoff of the approved prototype:
- At the end of step 3, user finally **approve the final version of prototype**, which commits them to current design and establishes a contractual obligation about what system will, and will not, do or provide.

Strengths:
3 Users:
Code:
- *Mann me laddu futa-participate karne ko mila(kaise participate kare? – evaluate kar k)*
- *mann me aur ek laddu futa- ☺dekho maine kitna quick implement kar dia ☺*

1. Improves **User participation**
2. A very short period (eg: week) is required to develop and start experimenting with prototype. This small time **allows** system **users to** immediately **evaluate proposed system changes.**
3. Provides **quick implementation** of an incomplete, but functional, application.

3 Engineers:
Code:
- *Mann me laddu futa- knowledge exploit karne ko mila.*

• *mann me aur ek laddu futa- ☺ missing functions easily mil jayenga. ☺ bolo kaise.? Toh bola user bata denga..*

4. Potential exist for **exploiting knowledge** gained at early iteration as later iterations are developed.
5. Helps to **easily identify missing functionalities**.
6. Since, system users experiment with each version of prototype; **errors are hopefully detected** at early stage of development process.

2 Management:

Code:

Management bola wah kya flexible ha prototype like ☺ Dabbang ☺.

7. Especially **useful for solving unclear objectives**, experimenting with and comparing various design solutions.
8. Encourages **flexible design**

1 stakeholder:

9. **Improves communication** to shareholders.

Weakness:

1. Approval Process and **control not strict**.*(laddu futtne k chakkar me kisi ne approval pe dhyan nai dia)*

Users ki dairy milk shots khaatm ho gait oh ab dekho uski demand:
Kuch user bolte ha humare pas time nai ha aur baki k roj subah ate ha ek nayi demand k sath.

2. **Requirements** of users may **change frequently**.

3. Prototyping will be **successful if the system users are willing to devote significant time** in experimenting with prototype and provide the system developers with change suggestions. The users may not be able or willing to spend time required under the prototyping approach.

And finally jab sab user requirement satisfy nai honga toh, users aur engineer ka babal ho gaya:

4. Prototyping may **cause behavioral problems** with system users. These problems include dissatisfaction by users if system developers are unable to meet all user demands for improvements

Ab engineer gusse me honga toh dekho kya karenga.

5. Prototype may not have **sufficient checks incorporated**. *(Engineer ne socha jitney kam check honge utni jyada user galti karenga)*

6. The interactive process of prototyping causes the prototype to be experimented with quite extensively. Because of this system developers are tempted to minimize the testing and documentation process of ultimately approved system. Inadequate testing can make the system error-prone, and inadequate documentation makes this system difficult to maintain.*(pak gaya experiment kar kar k toh testing bhi kam kar denga aur documentation bhi.)*

Nov 2002: If you are the project manager of the software company with the responsibility for developing a break-through product, combining state of art hardware and software, will you opt for prototyping as a process model for a product meant for the intensely competitive market?

Answer:

Reason 1:

Prototyping require user involvement. Here the users are the consumers of the product who are diffused and may not be inclined to join it.

Reason 2:

When we try to test the product with the involvement of customers, confidential or critical information might get leaked to the competitors on our line of thinking. The element of surprise and also the opportunity to capture the market will be lost.

Reason 3:

Prototyping require significant time for experimenting. Since the product is meant for the intensely competitive market, the project manager may not have much time to experiment, the competitor may capture the market by entering market in advance.

5.4.3 The Incremental Model

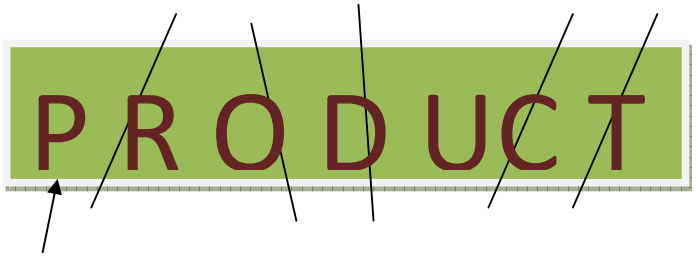
Framework Type: Combination Linear and Iterative.

Basic Principles:

The Incremental build model is a method of software development where the model is **designed, implemented and tested incrementally** (a little more is added each time) until the product is finished.

The product is defined as finished when it satisfies all of its requirements.

This model combines the elements of the waterfall model with the iterative philosophy of prototyping.



The product is **decomposed** into a number of components, each of which are designed and built separately (termed as builds)*(above diagram has 6 builds, which will be developed one after another and delivered to client as they are complete).*

Each component is delivered to the client when it is complete.

- Avoids a long development time
- Ease the traumatic effect of introducing completely new system all at once
- So allows Partial utilization

Basic Principles:

- A series of mini-waterfalls are performed, where all phases of the waterfall development model are completed for a small part of the system, before proceeding to the next increment. *(only waterfall on every build i.e. full~~overall~~ requirement are defined)*

- Overall requirements are defined before proceeding to evolutionary, mini – Waterfall development of individual increments of the system. *(define overall requirement than only waterfall, jo nayi requirement yengi vo next mini waterfall me puri kar dene ki)*

- The initial software concept, requirement analysis, and design of architecture and system core are defined using the Waterfall approach, followed by iterative Prototyping, which culminates in installation of the final prototype (i.e. Working system). *(designing tak waterfall bad me prototype)*

Strengths

(i) Helps to mitigate integration risks earlier in the project. *(Think of a situation of waterfall where you have to integrate 6 builds into one viz-a-viz incremental where you have to integrate two builds.)*

Engineers:

(ii) Potential exists for exploiting knowledge gained in an early increment as later increments are developed.

Users:

(iii) After every build is complete, approval/sign offs is obtained from users so moderate control is maintained over the life of the project.

Stakeholders:

(iv) Stakeholders can be given concrete evidence of project status throughout the life cycle.

Management:

(v) More flexible and also less costly to change requirements.

Company:

(vi) Gradual implementation provides the ability to monitor the effect of incremental changes, isolated issues and make adjustments before the organization is negatively impacted.

Weaknesses:

Company:

(i) When utilizing a series of mini-waterfalls for a small part of the system before moving onto the next increment, there is usually a lack of overall consideration of the business problem and technical requirements for the overall system.

Management:-*company ka problem sun k management bola ha bat toh sahi ha overall problem ka dhyan nai rakha toh architectural problem ana hi ha.*

(ii) Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.

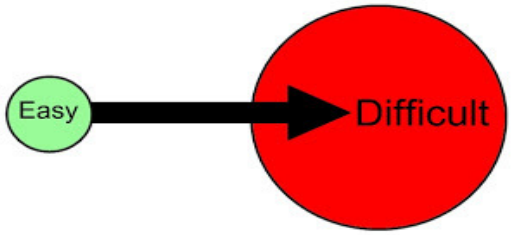
(iii) Since some modules will be completed much earlier than others, well-defined interfaces are required.

Waterfall:

(iv) Each phase of an iteration is rigid and do not overlap each other.*(waterfall me ye limitation tha ki overlap each other aur yaha ye limitation ha k do not overlap each other)*

Engineers:

(v) Difficult problems pushed to the future to demonstrate early success to management.



5.4.3 Spiral Model

Nov 10: As a person in charge of SDLC, you are assigned the job of developing a model for a new system which combines the features of prototyping and waterfall. Which will be the model of your choice and what are its strengths and weakness?

Framework Type: Combination Linear and Iterative.

Basic Principles:

The Spiral model is a software development process combining elements of both waterfall and prototyping.

Also known as the **Spiral Lifecycle**.

The spiral model is intended for large, expensive and complicated projects.

(i) The **new system requirements are defined** in as ***much detail as possible***.
This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.

(ii) A **preliminary design is created** for the new system.
This phase is the most important part of "Spiral Model" in which all possible alternatives, that can help in developing a cost effective project are analyzed and strategies are decided to use them.
This phase has been added specially in order to identify and resolve all the possible risks in the project development.
If risks indicate any kind of uncertainty in requirements, prototyping may be used to proceed with the available data and find out possible solution in order to deal with the potential changes in the requirements.

(iii) A **first prototype of the new system in constructed** from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

(iv) A second prototype is evolved by a fourfold procedure :

1. **Evaluating the first prototype** in terms of its strengths, weaknesses, and risks
2. **Defining the requirements of the second prototype**
3. **Planning and designing the second prototype**
4. **Constructing and testing** the second prototype.

Game development is a main area where the spiral model is used and needed, that is because of the size and the constantly shifting goals of those large projects.

Strengths

(i) Enhance risk avoidance.

(ii) Useful in helping to select the best methodology to follow for development of a given software iteration based on project risk.

(iii) Can incorporate Waterfall, Prototyping, and Incremental methodologies as special cases in the framework,
and provide guidance as to **which combination of these models best fits a given software**

iteration, based upon the type of project risk.

For example, a project with low risk of not meeting user requirements but high risk of missing budget or schedule targets would essentially follow a linear Waterfall approach for a given software iteration. Conversely, if the risk factors were reversed, the Spiral methodology could yield an iterative prototyping approach.

Weaknesses

(i) Challenges to determine the exact composition of development methodologies to use for each iteration around the Spiral.

(ii) Highly customized to each project, and thus is quite complex, limiting reusability.

(iii) A skilled and experienced project manager required to determine how to apply it to any given project.

(iv) No established controls for moving from one cycle to another cycle. Without controls, each cycle may generate more work for the next cycle.

(v) No firm deadlines - cycles continue with no clear termination condition, so there is an inherent risk of not meeting budget or schedule.

5.4.4 Rapid Application Development (RAD)

Framework Type: Iterative.

Basic Principles:

- Rapid Application Development (RAD) refers to a type of software development methodology which uses **minimal planning in favor of rapid prototyping.**
- The **"planning"** of software developed using RAD is **interleaved with writing the software itself.**
- The lack of extensive pre-planning generally **allows software to be written much faster**, and makes it easier to change requirements.

(i) **Key objective** is for **fast development** and delivery of a **high quality system** at a relatively **low investment** cost,

(ii) Attempts to **reduce inherent project risk** by breaking a project into smaller segments.

(iii) **Aims to produce high quality systems** quickly, primarily through the use of iterative Prototyping (at any stage of development), active user involvement, and computerized development tools.

(iv) Key **emphasis is on fulfilling the business need** while technological or engineering excellence is of lesser importance.

(v) Active **user involvement is imperative**.

(vi) **Iteratively produces production software**, as opposed to a throwaway prototype.
(In prototype we saw that prototype can be scrapped and knowledge gained is used for developing software but yaha itna time nai ha scrap karne ka)

(ix) Produces **documentation necessary to facilitate future maintenance**.

Strengths

(i) The **operational version of an application is available much earlier** than with Waterfall, Incremental, or Spiral frameworks.

(ii) Because RAD produces systems more quickly and to a business focus, this approach tends to **produce systems at lower cost.**

(iii) Constant integration **encourage customer feedback.**

(iv) **Users are seen as gaining more of a sense of ownership of a system, while developer are seen as gaining more satisfaction** from producing successful systems quickly.

(v) **Concentrates on essential system elements** from user viewpoint.

(vi) Provides the **ability to rapidly change system design** as demanded by users.

(vii) Generally produces a dramatic **savings in time, money and human effort.**

Weaknesses

(i) More speed and lower cost may lead to a **lower overall system quality**.

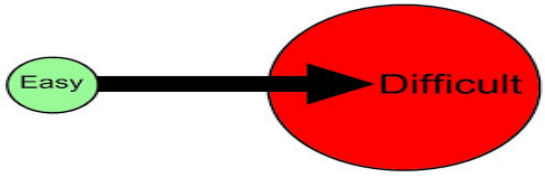
(ii) Project may **end up with more requirements** than needed (gold-plating).

(iii) Potential for **inconsistent designs** within and across systems.

(iv) Potential for **violation of programming standards** related to inconsistent naming conventions and inconsistent documentation,

(v) Formal **reviews and audits are more difficult** to implement than for a complete system.

(vi) Tendency for **difficult problems to be pushed to the future** to demonstrate early success to management.



(vii) Since some modules will be completed much earlier than others, **well –defined interfaces are required.**

5.4.6 Agile Model

Nov 13: Describe Agile Methodology of system development and describe its strengths.

- This is an **organized set of software development methodologies** based on the *iterative and incremental* development.
- This approach describes that software development is essentially a human activity and will always include variations in processes and inputs; therefore model should be flexible enough to handle the variations.
- This approach does not follow any pre defined steps.
- This methodology uses flexible planning, evolutionary development and delivery and encourages rapid and flexible response to changing requirement.

Following are considered as the key features of the agile methodologies:

- Customer satisfaction by **rapid delivery** of useful software;
- **Welcome changing requirements**, even late in development;
- Working **software is delivered frequently** (weeks rather than months);
- **Working software is the principal measure** of progress;
- Close, **daily co-operation between** business people and developers;
- **Face-to-face conversation is the best form** of communication (co-location);
- Simplicity; **Self-organizing teams**; and
- **Regular adaptation to changing circumstances**.

(a) Strengths: Some of the strengths identified by the experts and practitioners include the following:

• Agile methodology has the concept of an adaptive team, which enables to **respond to the changing requirements.**

• The team does not have to invest time and efforts and finally find that by the time they delivered the product, the requirement of the customer has changed.

• Face to face communication and continuous inputs from customer representative leaves a **little space for guesswork.**

• The **documentation is crisp** and to the point to save time.

• The **end result is generally the high quality software** in least possible time duration and

satisfied customer.

(b) Weaknesses: Some of the weaknesses identified by the experts and practitioners include the following:

- In case of some software deliverables, especially the large ones, it is **difficult to assess the efforts required at the beginning** of the software development life cycle. *(engineer ko puchonge kitne time me software denga toh vo nai bata payenga)*
- There is **lack of emphasis on necessary designing and documentation.** *(blue print banana me koi dhyan nai deta, document toh thoda bahut bhi bana lete ha)*
- Lack of documentation can result in **problem to transfer of project and knowledge**. By nature, Agile projects are extremely light on documentation because the team focuses on verbal communication with the customer rather than on documents or manuals.
- Agile **requires more re-work** and due to the lack of long-term planning and the lightweight approach to architecture, re-work is often required on Agile projects when the various components of the software are combined and forced to interact.
- The project can **easily get taken off track** if the customer representative is not clear about the final outcome.

5.5 System Development Life Cycle (SDLC)

Nov 2004: Write a short note on SDLC
Nov 2000: What is SDLC? Explain different activities coming under this cycle.
Ans: (Introductory para and 7 phases)

- The System Development Life Cycle is the **set of activities** that a system analyst has to carry out **to develop and implement an information system** under various approaches of devlopment.
- It consists of a generic sequence of steps or phases in which each phase of the SDLC uses the results of the previous one.

The advantages of SDLC are given as follows:
Code: Better planning control aur quality se har milestones pe documentation kia.

- Better planning and control by project managers;
- Compliance to prescribed standards ensuring better quality;
- The phases are important milestones and help the project manager and the user for review and signoff.
- Documentation that SDLC stresses on is an important measure of communication and control; and

From the perspective of the IS Audit, the following are the possible advantages:

Nov 10: From the perspective of IS Audit, what are the advantages of SDLC?

- The IS auditor can have clear understanding of various phases of the SDLC on the basis of the detailed documentation created during each phase of the SDLC.

- The IS Auditor on the basis of his/her examination, can state in his/her report about the compliance by the IS management of the procedures, if any, set by the management.

- The IS Auditor, if has a technical knowledge and ability of different areas of SDLC, can be a guide during the various phases of SDLC.

- The IS auditor can provide an evaluation of the methods and techniques used through the various development phases of the SDLC.

Some of the shortcomings and anticipated risks associated with the SDLC are as follows:

- The development team may find it cumbersome.

- The users may find that the end product is not visible for a long time.

- The rigidity of the approach may prolong the duration of many projects.

- IT may not be suitable for small and medium sized projects.

Brief summary cross referencing various pages of this notes.

Phase		Page No. of this notes
I.	Preliminary Investigation	21 to 27
II.	System Requirement Analysis	27 to 40
III.	System Design	41 to 47
IV.	System Development	47 to 52
V.	System Testing	53 to 64
VI.	System Implementation	64 to 68
VII.	(I) Post Implementation review	68 to 70
	(II)System Maintenance	70 to 73

(Intentionally left blank)

Prelimienary Investigation

- Determining and evaluating the strategic feasibility of the system and ensure that the solution fits the business strategy.

System Requirement Analysis

- Analyzing the typical system requirements, in view if its functionalities, deliverables etc.

System Design

- Designing the system in terms of user interface, data storage and data processing functions on the basis of the requirement phase by developing the system flowcharts, system and data flow diagrams, screens and reports.

System Devlopment

- Developing the system as per the system designed in view of its adequate implementation to lead to fulfillment of requirements to the satisfaction of all the stakeholders.

System Testing

- Requisite testing to ensure the valid and reliable implementations.

System Implementation

- Operationalization of the developed system for the acceptance by management and user before migration of the system to the live environment and data conversion from legacy system to the new system.

System Maintainance

- Continuous evaluation of the system as it functions in the live environment and its updation / maintenance.

5.5.1 Preliminary Investigation

It is predominantly **aimed to determine and analyze** the **strategic benefits in implementing the system**…
*(How to evaluate)…***through evaluation and quantification of** - productivity gains; future cost avoidance; cost savings, and Intangible benefits like improvement in morale of employees.
The deliverable of the preliminary investigation includes a report including feasibility study observations.

What are the objectives that a system analyst working in preliminary investigation stage must achieve?
Code: Request ko ache se samaj k uska size determine karne ka, time mila toh alternative approach ka feasibility aur unka cost benefit analysis kar k management ko report de do.

1. Clarify and **understand the project request**;

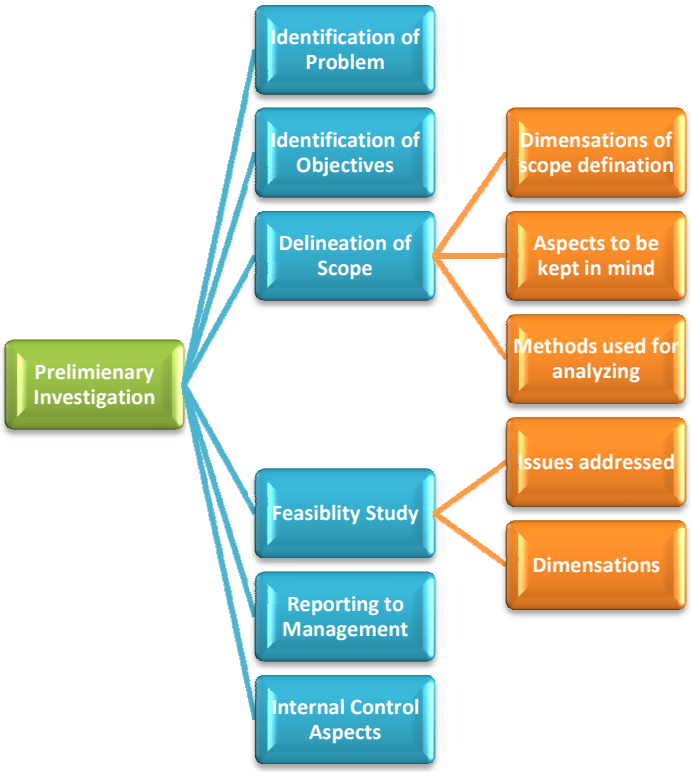
2. **Determine the size** of the project;

3. **Determine the technical and operational feasibility** of alternative approaches;

4. Assess **costs and benefits** of alternative approaches; and

5. **Report findings to the management** with recommendation outlining the acceptance or rejection of the proposal.

A preliminary investigation is normally **initiated by some sort of system request.**



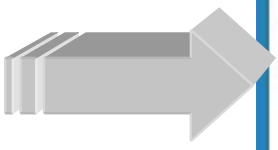
(i) Identification of Problem:

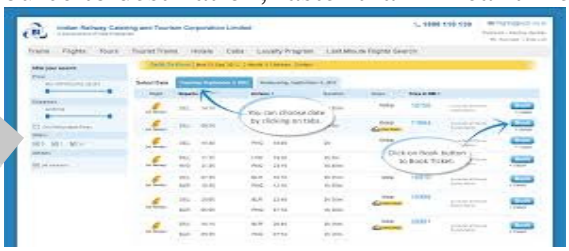
The first step in an application development is to **define the problem clearly and precisely**, which is done only after the critical study of the existing system and several rounds of discussions with the user group. Then its prevalence within the organization has to be assessed.

(ii) Identification of Objectives:

After the identification of the problem, it is easy to work out and **precisely specify the objectives of the proposed solution.**

For instance, inability to provide a convenient reservation system, for a large number of intending passengers was the problem of the Railways. So, one of the objectives was 'to introduce a system wherein intending passengers could book a ticket from source to destination, faster than in real-time'.





(iii) Delineation of Scope:

The scope of a solution **defines its typical boundaries**.

It should be clear and comprehensible to the user management stating the extent and **'what will be addressed by the solution and what will not'.**

Often, the scope becomes a contentious issue between development and user organizations.

The typical scope determination may be performed on the following dimensions:

- **Functionality Requirements:** What functionalities will be delivered through the solution?
- **Data to be Processed:** What data is required to achieve these functionalities?
- **Control Requirements:** What are the control requirements for this application?
- **Performance Requirements:** What level of response time, execution time and throughput is required?
- **Constraints:** What are the conditions the input data has to conform to? For example, what is the maximum number of characters that a name can have in a database?
- **Interfaces:** Is there any special hardware/software that the application has to interface with? For example-Payroll application may have to capture from the attendance monitoring system that the company has already installed. Then the solution developer has to understand the format of data, frequency mode of data transfer and other aspects of the software.
- **Reliability requirements:** Reliability of an application is measured by its ability to remain uncorrupted in the face of inadvertent / deliberate misuse and probability of failure-free operations. The reliability required for an application depends on its criticality and the user profile.

Moreover, while eliciting information to delineate the scope, few aspects needs to be kept in mind:

Code: Ladla Movie(Discussed in class)

➢ Different users may represent the problem and required solution in different ways. The system developer should **elicit the need from the initiator of the project** alternately called champion or executive sponsor of the project, addressing his concerns should be the basis of the scope.

➢ **While the initiator of the project may be a member of the senior management**, the actual users may be from the operating levels in an organization. An **understanding of their profile helps in designing appropriate user interface features**.

➢ While presenting the proposed solution for a problem, the **development organization has to clearly quantify the economic benefits** to the user organization. The information required has to be gathered at this stage. For example, when a system is proposed for Road tax collection, data on the extent of collection and defaults is required to quantify benefits that will result to the Transport Department.

➢ It is also necessary to **understand the impact of the solution on the organization**- its structure, roles and responsibilities. Solutions, which have a wide impact, are likely to be met with greater resistance. ERP implementation in organizations is a classic example of change management requirement. Organizations that have not been able to handle it may have a very poor ERP implementation record with disastrous consequences.

➢ **While economic benefit is a critical consideration** when deciding on a solution**, there are several other factors that have to be given weightage too.** These factors are to be considered from the perspective of the user management and resolved. For example, in a security system, how foolproof it is, may be a critical factor like the economic benefits that entail.

Two primary methods with the help of which the scope of the project can be analyzed are given as follows:

Nov 1999: Explain the two primary methods by which an analyst collects data during preliminary investigation.

- **Reviewing Internal Documents:**
  - ➢ The analysts conducting the investigation **first try to learn about the organization involved** in, or affected by, the project.
  - ➢ For example, to review an inventory system proposal, an analyst may try to know how does the inventory department operates and who are the managers and supervisors.
  - ➢ Analysts can **usually learn** these details by **examining organization charts and studying written operating procedures.**

- **Conducting Interviews:**
  - ➢ **Written documents** tell the analyst how the systems should operate, but they **may not include enough details to allow a decision to be made** about the merits of a systems proposal, **nor do they present users' views** about current operations.
  - ➢ To learn these details, analysts use interviews.
  - ➢ Interviews **allow analysts to know more about the nature of the project request** and the reasons for submitting it.

➢ **Usually**, preliminary investigation **interviews involve only management and supervisory personnel.**

(iv) Feasibility Study:

May 09: The top management of the company has decided to develop a computer information system for its operations. Is it essential to conduct the feasibility study of system before implementing it? If your answer is yes, state the reasons. Also discuss three different angles through which the feasibility study can be conducted.

After possible solution options are identified, project feasibility i.e. the likelihood that these systems will be useful for the organization is determined. A feasibility study is carried out by the system analysts, which refers to a process of evaluating alternative systems through cost/benefit analysis so that the most feasible and desirable system can be selected for development.

Which issues are typically addressed in the feasibility study?
OR
Preliminary Investigation enables requirement engineer (system analyst) to tackle the issues and feasibility study for following:

1. Determine whether the **solution** is **as per the business strategy**;
2. Determine whether the **existing system can rectify the situation** without a major modification;
3. Determine whether the **vendor product offers a solution** to the problem.
4. Define the **time frame for which the solution is required**;
5. Determine the **approximate cost** to develop the system.

The Feasibility Study of a system is evaluated under following dimensions described briefly as follows:

Code: The LOSER BF

- **Technical:** Is the technology needed available?
- **Legal:** Is the solution valid in legal terms?
- **Operational:** How will the solution work?
- **Schedule/Time:** Can the system be delivered on time?
- **Economic:** Return on Investment?
- **Resources:** Are human resources reluctant for the solution?
- **Behavioral:** Is to determine whether the inputs will be available on time to provide timely output?
- **Financial:** Is the solution viable financially?

The detailed description of each dimension is given as follows:

(a) **Technical Feasibility:**
It may try to answer, whether **implementation** of the project **viable using current technology?**
An analyst **ascertains whether the proposed system is feasible with existing or expected computer hardware and software technology.**
The technical issues usually raised during the feasibility stage of investigation include the following:

- Can the proposed application be implemented with existing technology?
- Does the necessary technology exist to do what is suggested (and can it be acquired)?
- Does the proposed equipment have the technical capacity to hold the data required to use the new system?
- Can the system be expanded if developed?
- Will the proposed system provide adequate responses to inquiries, regardless of the number or location of users?
- Are there technical guarantees of accuracy, reliability, ease of access, and data security?

(b) Financial Feasibility:
The solution proposed may be prohibitively costly for the user organization. For example, Monitoring the stock through VSAT network connecting multiple locations may be acceptable for an organization with high turnover. But this may not be a viable solution for smaller ones.

(c) **Economic Feasibility:**
It includes an evaluation of all the incremental costs and benefits expected if the proposed system is implemented. After problems or opportunities are identified, the analysts must determine the scale of response needed to meet the user's requests for a new system as well as the approximate amount of time and money that will be required in the effort.
The financial and economic questions raised by analysts during the preliminary investigation are for the purpose of estimating the following:

- The cost of conducting a full systems investigation;
- The cost of hardware and software for the class of applications being considered;
- The benefits in the form of reduced costs or fewer costly errors; and
- The cost if nothing changes (i.e. the proposed system is not developed).

(d) **Schedule or Time Feasibility:**
Schedule feasibility involves the design team's estimating how long it will take a new or revised system to become operational and communicating this information to the steering committee. For

example, if a design team projects that it will take 16 months for a particular system design to become fully functional, the steering committee may reject the proposal in favor of a simpler alternative that the company can implement in a shorter time frame.

(e) **Resources Feasibility:**
This focuses on human resources. Implementing sophisticated software solutions becomes difficult at specific locations because of the reluctance of skilled personnel to move to such locations.

(f) **Operational Feasibility:**
It is concerned with **ascertaining the views of workers, employees, customers and suppliers** about the use of computer facility. A system can be highly feasible in all respects except the operational and fails miserably because of human problems.
Some of the questions, which help in testing the operational feasibility of a project, may include the following:
Code: New course by ICAI

- Is there sufficient support for the system from management*(CA)* and from users*(Students)*? *(naye course ko CA aur student support kar rahe ha ya nai?)*
- Are current business methods acceptable to users? *(Students ko current methods acceptable ha ya nai)*
- Have the users been involved in planning and development of the project? *(course banana me students ka feedback lia ya nia)*
- Will the proposed system cause harm? Will it produce poorer results in any respect or area? *(kya naya course kuch harm karenga by getting poor results in particular subject)*
- Will individual performance be poorer after implementation than before? *(kya students ka performance poor honga compared to before)*

(g) Behavioral Feasibility:
In this feasibility, the issues relating to system environment, primarily; collection of input is analyzed to determine whether required input will be available on time for processing, specially when inputs are provided by external entities.*(say Financial Institution)*

(h) **Legal Feasibility:**
Legal feasibility is largely concerned with whether there will be any conflict between a newly proposed system and the organization's legal obligations. Any system, which is liable to violate the local legal requirements, should also be rejected. For example, a revised system should comply with all applicable statutes about financial and statuary reporting requirements, as well as the company's contractual obligations.

(iv) Reporting Results to Management:
After the analyst articulates the problem, defines the same along with its scope, s/he **provides one or more solution alternatives and estimates the cost and benefits of each alternative and reports**

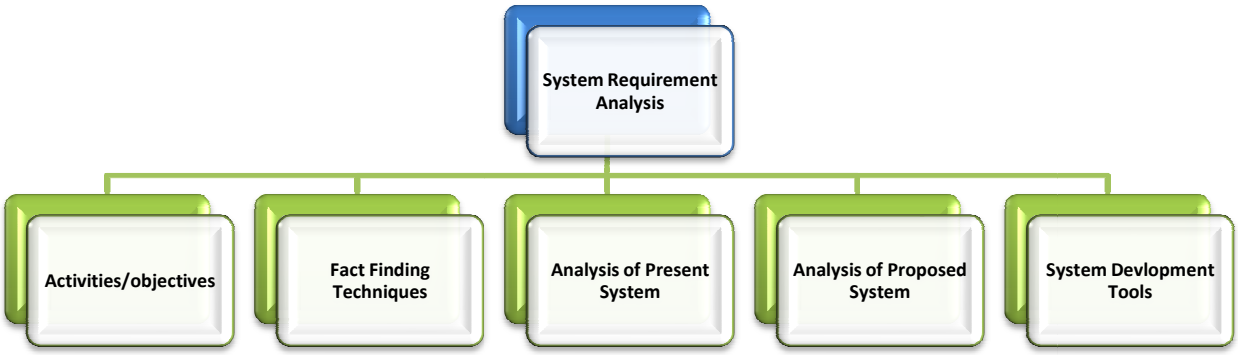
these results to the management.
The report should be accompanied by a short covering letter of intent that summarizes the results and makes the recommendation regarding further procedures.
From the analyst's report, management should determine what to do next.

(vi) Internal Control Aspects:
The key internal control aspects in this phase are as follows:

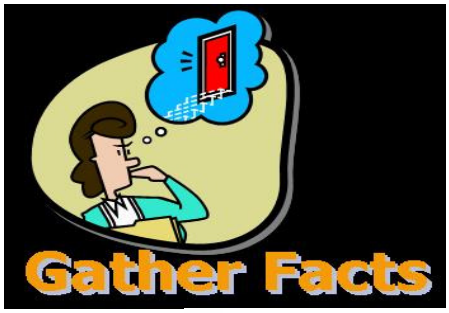
- Whether problem definition is proper? *(Stage1)*
- Whether all feasibility studies have been properly done? *(Stage 3)*
- Whether results of feasibility studies have been documented? *(Stage 3 ka documentation)*
- Whether management report submitted reflects the outcome of feasibility studies done? *(Stage 4)*

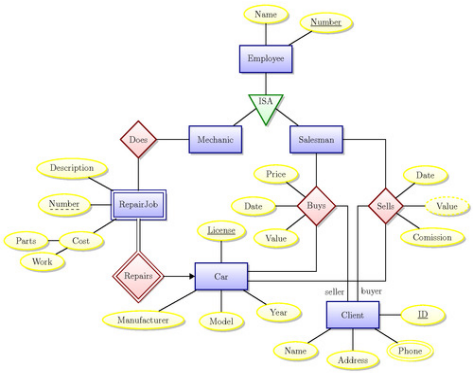
5.5.2 System Requirements Analysis

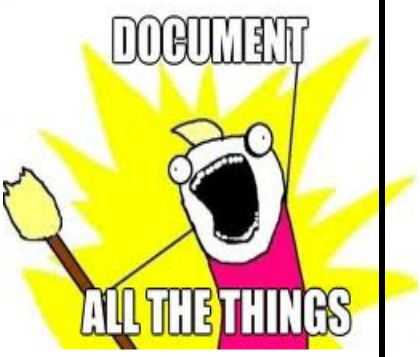
May 2002: What do you mean by 'requirement analysis'?
This phase includes a thorough and detailed understanding of the current system, identifies the areas that need modification to solve the problem, the determination of user/managerial requirements and to have fair idea about various systems development tools.



May 2013: List the activities to be performed during System Requirement Analysis.
The following objectives are performed in this phase in order to generate the deliverable, Systems Requirements Specification (SRS):







- To identify and consult the stake owners to **determine their expectations** and resolve their conflicts;

- To analyze requirements to detect and **correct conflicts and determine priorities**; *(expectation me se conflict nikalo aur bache hue me se priority set karo ko konsi expectation pehle fulfill karni ha)*

- To **gather data or find facts** using tools like - interviewing, research/document collection, questionnaires, observation;

- To verify that the **requirements are complete, consistent, unambiguous, verifiable**, modifiable, testable and traceable;

- To **model activities** such as developing models to document Data Flow Diagrams, E-R Diagrams; and

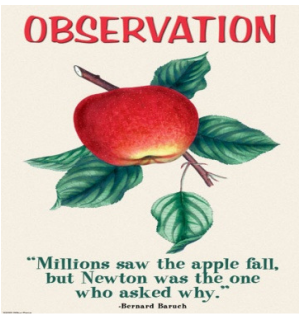
- To **document activities** such as interview, questionnaires, reports etc. and development of a system (data) dictionary to document the modeling activities.

(i) Fact Finding:

- Every system is built to meet some set of needs, for example, the need of the organization for lower operational costs, better information for managers, smooth operations for users or better levels of services to customers.
- To assess these needs, the analysts often interact extensively with people, who will be benefited from the system in order to determine 'what are their actual requirements'.
- Various fact-finding techniques/tools are used by the system analyst for determining these needs/requirements are briefly discussed below:

Nov 2005: What are the fact finding techniques used by system analyst?





- **Documents:**
  - ✓ Document means manuals, input forms, output forms, diagrams of how the current system works, organization charts showing hierarchy of users and manager responsibilities, job descriptions for the people, who work with the current system, procedure manuals, program codes for the applications associated with the current system, etc.
  - ✓ Documents are a **very good source of information** about user needs and the current system.

- **Questionnaires:**
  - ✓ Users and managers are asked to complete questionnaire about the information systems when the traditional system development approach is chosen.
  - ✓ The main strength of questionnaires is that a **large amount of data can be collected through a variety of users quickly**.
  - ✓ Also, if the questionnaire is skillfully drafted, **responses can be analyzed rapidly with the help of a computer.**

- **Interviews:**
  - ✓ Users and managers may also be interviewed to extract information in depth.
  - ✓ The data gathered through interviews often **provide** system developers with a larger **picture of the problems and opportunities.**
  - ✓ Interviews also give analyst the opportunity to observe and **record first-hand user reaction** and to probe for further information.

- **Observation:**
  - ✓ In general and particularly **in prototyping approaches, observation plays a central role** in requirement analysis.
  - ✓ Only by observing how users react to prototypes of a new system, the system can be successfully developed.

(ii) Analysis of the Present System:

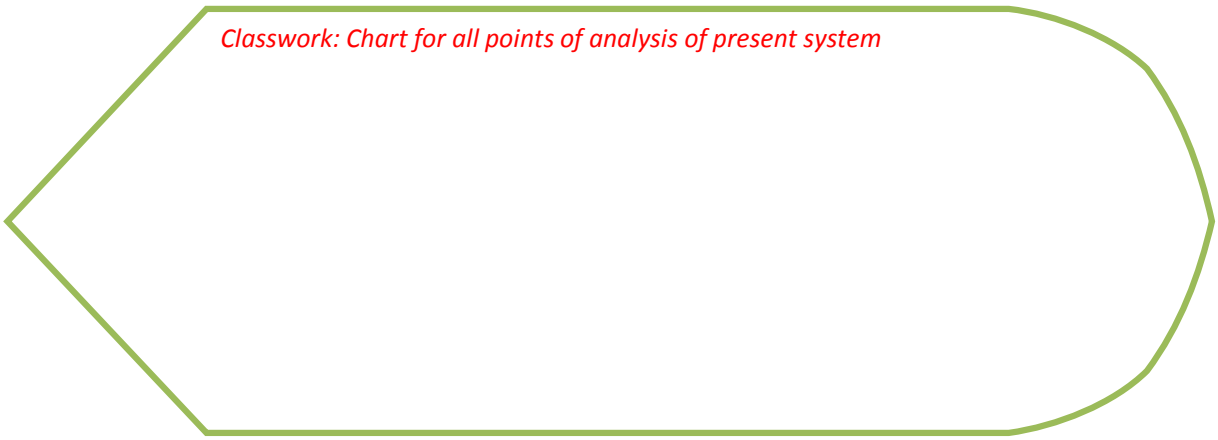
May 1999/May 05/ May 08: Discuss in detail, how the analysis of present system is made by the system analyst.

May 2011: Discuss the various functional areas to be studied by system analyst for a detailed investigation of present system.

Detailed investigation of the present system involves **collecting, organizing and evaluating facts about the system and the environment** in which it operates.

There should be **enough information assembled** so that a qualified person **can understand the present system without visiting** any of the operating departments.

The following areas should be studied in depth:



- **Reviewing Historical Aspects:**
  - A brief history of the organization is a **logical starting point for an analysis** of the present system. The historical facts **enable to identify the major turning points and milestones** that have influenced its growth.
  - A **review of annual reports and organization charts** can identify the growth of management levels as well as the development of various functional areas and departments.
  - The system analyst should ***investigate 'what system changes have occurred in the past including operations' that have been successful or unsuccessful*** with computer equipment and techniques.

- **Analyzing Inputs:**
  - A detailed analysis of present inputs is **important since they are basic to the manipulation of data.** Source documents are used to capture the originating data for any type of system.
  - The system **analyst should be aware of various sources** from where the data are initially captured, *keeping in view the fact that outputs for one area may serve as an input for another* area.
  - The system analyst must **understand the nature of each form**, 'what is contained in it', 'who prepared it', 'from where the form is initiated', 'where it is completed', the distribution of the form and other similar considerations.
  - If the analyst investigates these questions thoroughly, s/he will be **able to determine how these inputs fit into the framework of the present system.**

- **Reviewing Data Files:**
  - The analyst should **investigate the data files** maintained by each department, **noting their number and size,** where they are located, who uses them and the number of times per given time interval, these are used.
  - **Information on common data files and their size will be an important factor**, which will influence the new information system.
  - The system analyst should also **review all on-line***(file on integrated database)* **and off-line files**, which are maintained in the organization ***as it will reveal information about data that are***

not contained in any outputs.

- The **related cost of retrieving and processing** the data is another important factor that should be considered by the systems analyst.

• **Reviewing Methods, Procedures and Data Communications:**

- A method is defined as a way of doing something; a procedure is a series of logical steps by which a job is accomplished.
- A procedure review is an **intensive survey of the methods** by which each job is accomplished, the equipment utilized and the actual location of the operations.
- Its basic objective is to **eliminate unnecessary tasks or to perceive improvement opportunities** in the present information system.

- A system analyst also needs to **review and understand the present data communications** used by the organization.
- S/he must **review the types of data communication equipment** including data interface, data links, modems, dial-up and leased lines and multiplexers.

• **Analyzing Outputs:**

- The outputs or reports should be **scrutinized carefully** by the system analysts **in order to determine 'how well they will meet the organization's needs**.
- The analysts must **understand what information is needed and why, who needs it** and when and where it is needed.
- Often, many reports are a **carry-over from earlier days and have little relevance to current operations**. Attempts should be made to eliminate all such reports in the new system.

• **Reviewing Internal Controls:**

- A detailed investigation of the present information system is not complete until internal control mechanism is reviewed.
- **Locating the control points helps** the analyst to **visualize the essential parts** and framework of a system.
- An examination of the present system of internal controls may **indicate weaknesses that should be removed** in the new system.

• **Modeling the Existing System:**

- **As the logic** of inputs, methods, procedures, data files, data communications, reports, internal controls and other important items **are reviewed and analyzed in a top down manner**; ***the processes must be properly documented.***
- The flow charting and diagramming of present information **not only organizes the facts, but also helps to disclose gaps and duplication in the data gathered**.
- It allows a **thorough comprehension of the numerous details** and related problems in the present operation.

• **Undertaking Overall Analysis of the Existing system:**

- Based upon the aforesaid investigation of the present information system, the final phase of the detailed investigation includes
- The **analysis of the present work volume**;

- The current **personnel requirements**;
- The present **costs-benefits** of each of these must be investigated thoroughly.

(iii) System Analysis of Proposed Systems:

- After a thorough analysis of each functional area of the present information system, the **proposed system specifications must be clearly defined, which is determined from the desired objectives,** set in the preliminary investigation stage.
- Likewise, consideration should be given to the strengths and short comings of the present system.

- **The required systems specifications should be in conformity with the project's objectives articulated and in accordance with the following:**

  - **Outputs** are produced with great emphasis on timely managerial reports that utilize the management by exception' principle. *(output ka focus timely managerial report mile uspe ho aur managerial report management by exception principle use karta ha)*

  - **Databases** to be maintained with desired capabilities such as online processing.

  - **Input** data is prepared directly from original source documents for processing by the computer system.

  - **Methods and procedures** that show the relationship of inputs and outputs to the database, utilize data communications as, when and where deemed appropriate.

  - Work volumes and timings are carefully considered for present and future periods including peak periods.

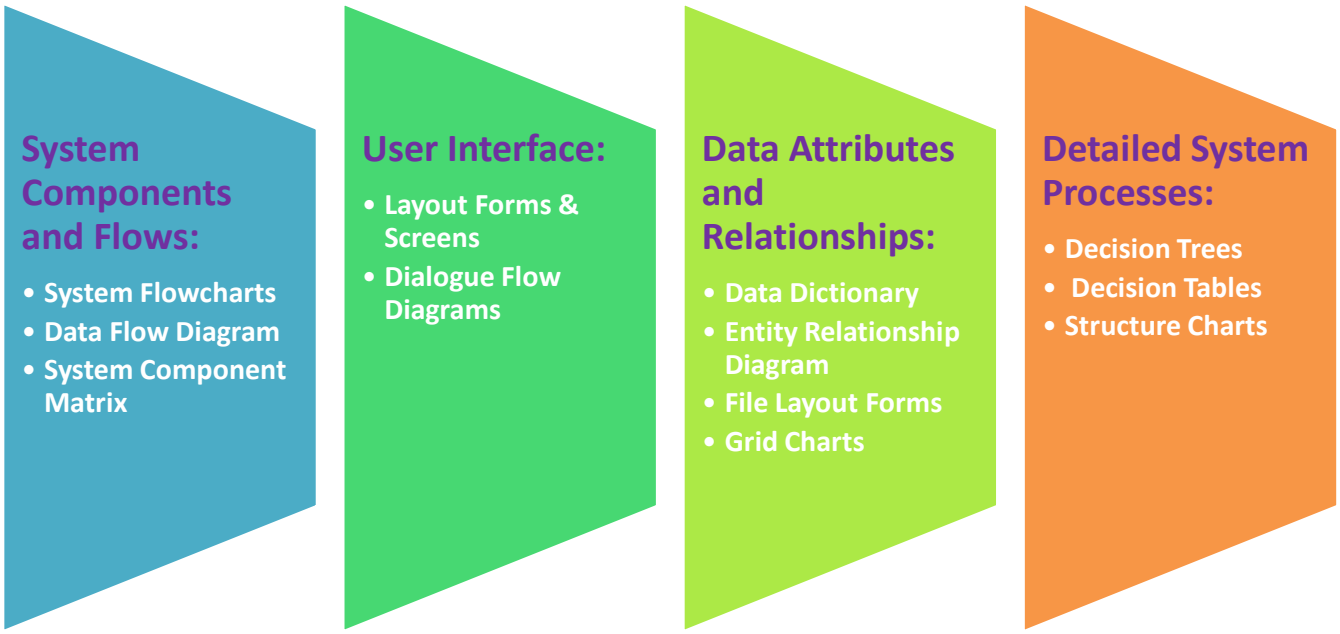
The starting point for compiling these specifications is output.*(Eg: based on tax, determine the income)*
After outputs have been determined, it is possible to infer what inputs, database, methods, procedures and data communications must be employed.

(iv) System Development Tools: Many tools and techniques have been developed to improve current information systems and to develop new ones.

Nov 2001: "In order to improve the current information systems and to develop the new ones, several system development tools have been developed". What are this system development tools? Explian them briefly.

May 2003: Describe any four categories of the major tools that are used for system development.

The major tools used for system development specification or representations can be classified into four categories based on the systems features. These are described briefly as follows:



(v) System Components and Flows:

➢ These tools help the system analysts to *document the data flow* among the major resources and activities of an information system.

- ➢ ***System flow charts*** are typically **used to show the flow of data** media as they are processed by the hardware devices and manual activities.
- ➢ A ***data flow diagram*** **uses a few simple symbols to illustrate the flow of data** among external entities (such as people or organizations etc.), processing activities and data storage elements.
- ➢ A ***system component matrix*** **provides a matrix framework to document** the resources used, the activities performed and the information produced by an information system.

• User Interface:

➢ *Designing the interface* between end users and the computer system is a major consideration of a system analyst while designing the new system.

- ➢ ***Layout forms and screens*** are used **to construct the formats and contents of input/output media** and methods.
- ➢ ***Dialogue flow diagrams*** **analyze the flow of dialogue** between computers and people.

• Data Attributes and Relationships:

➢ The *data resources in information system are defined, catalogued and designed* by this category of tools.

- ➢ A ***Data Dictionary*** catalogs the **description of the attributes (characteristics) of all data elements and their relationships** to each other as well as to external systems.
- ➢ ***Entity-relationship diagrams*** are **used to document the number and type of relationship among the entities** in a system.
- ➢ ***File layout forms*** **document the type, size and names of the data elements** in a system.
- ➢ ***Grid charts*** help in **identifying the use of each type of data element** in input/output or storage media of a system.

- **Detailed System Processes:**

➢ These tools are used to *help the programmer to develop detailed procedures and processes* required in the design of a computer program.

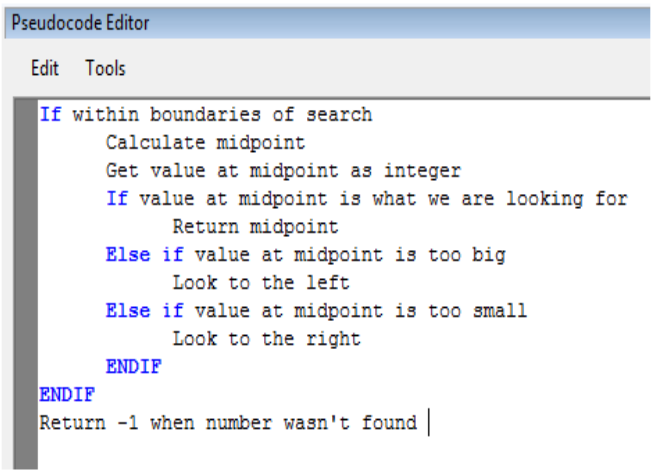
- ➢ ***Decision trees and decision tables*** **use a network or tabular form to document the complex conditional logic** involved in choosing among the information processing alternatives in a system.
- ➢ ***Structure charts*** **document the purpose, structure and hierarchical relationships** of the modules in a program.

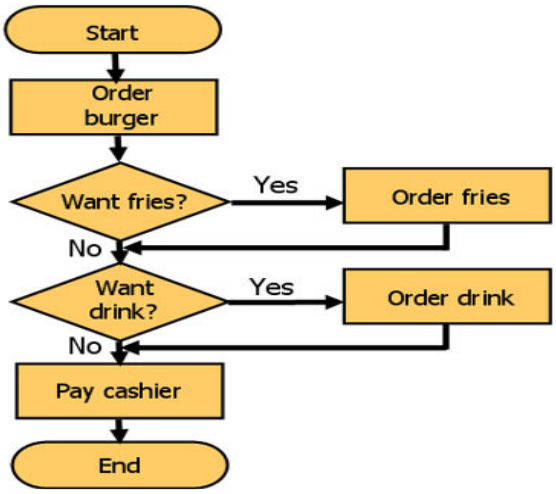
It is clear from the foregoing description that a number of CASE tools are in use for typical representation, specifications and system modeling. A set of the prominent tools have been already mentioned categorically and some are being described in detail as follows:

May 2004/Nov 2007: Describe any five system development tools.

(a) **Structured English:**

- ➢ Structured English, also known as **Program Design Language (PDL),** is **the use of the English language** with the syntax of structured programming.
- ➢ Thus, Structured English **aims at getting the benefits of both the programming logic and natural language.**
- ➢ Program logic that helps to attain precision and natural language that helps in getting the convenience of spoken languages.





(b) **Flowcharts:**

- ➢ Flowcharting is a **pictorial representation technique** that can be used by analysts to represent the inputs, outputs and processes of a business process.
- ➢ It is a **common type of chart that represents** an algorithm or **process showing the steps as boxes** of various kinds, **and their order by connecting these with arrows**.
- ➢ Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

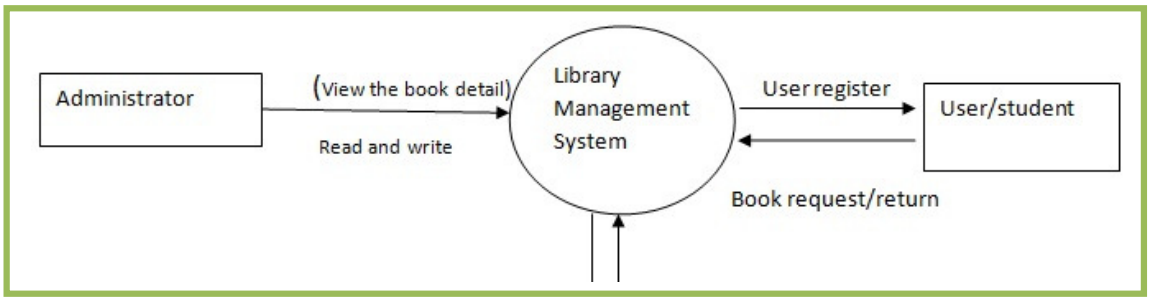
(c) **Data Flow Diagrams:**

Nov 1999: What is data flow diagram?

A Data Flow Diagram *uses few simple symbols* **to illustrate the flow of data** among external entities

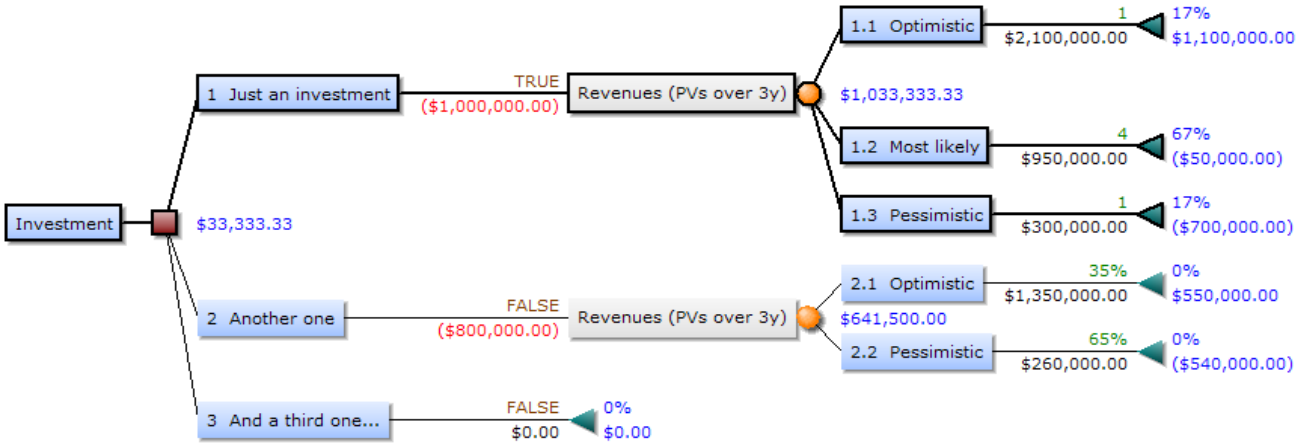
(such as people or organizations, etc.), processing activities and data storage elements.
A DFD is composed of four basic elements: Data Sources and Destinations, Data Flows, Transformation processes, and Data stores shown in table below.
These four symbols are combined to show how data are processed.

Symbol	Name	Explanation
	Data Sources and destinations	The people and organizations that send data to and receive data from the system are represented by square boxes called Data destinations or Data Sinks.
	Data flows Transformation process	The flow of data into or out of a process is represented by curved or straight lines with arrows. The processes that transform data from inputs to outputs are represented by circles, often referred to as bubbles.
	Data stores	The storage of data is represented by two horizontal lines.



(d) Decision Tree: *(Example: Baseyan Decision tree in Capital Budgeting)*

- A Decision Tree or tree diagram is a support tool that **uses a tree-like graph** of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.
- Decision tree is commonly **used in operations research**, specifically in decision analysis, to help identify a strategy most likely to reach a goal and to calculate conditional probabilities.



(d) **Decision Table:**

- A Decision Table is a table, which may **accompany a flowchart, defining the possible contingencies** that may be considered within the program and the appropriate course of action for each contingency.
- Decision tables are necessitated by the fact that branches of the flowchart multiply at each diamond (comparison symbol) and may easily run into scores and even hundreds.
- If, therefore, the programmer attempts to draw a flowchart directly, s/he is liable to miss some of the branches.

- **The four parts of the decision table are given as follows:**

  - **Condition Stub** – This comprehensively lists the comparisons or conditions.
  - **Action Stub –** This comprehensively lists the actions to be taken along various program branches.
  - **Condition entries –** This list in its various columns the possible permutations of answer to the questions in the conditions stub) and
  - **Action entries –** This lists in its columns corresponding to the condition entries the actions contingent upon the set of answers to questions of that column.

For better understanding of decision table an example is produced below. Students may note that below example is not in CA Final curriculum.

Example: No charges are reimbursed to the patient until the deductible has been met. After the deductible has been met, reimburse 50% for Doctor's Office visits or 80% for Hospital visits. There will be 4 rules:
(i) The first condition (Is the deductible met?) has two possible outcomes – yes or no.
(ii) The second condition (type of visit) has two possible outcomes - Doctor's office visit (D) or Hospital visit (H). Two times two is four.

Solution:

Conditions	**1**	**2**	**3**	**4**
1. Deductible met?	Y	Y	N	N
2. Type of visit	D	H	D	H
Actions				
1. Reimburse 50%	X			
2. Reimburse 80%		X		
3. No reimbursement			X	X

(e) **CASE Tools:**

- The **data flow diagram and system flow charts** that users review are commonly **generated by** systems developers **using** the on-screen drawing modules found in **CASE** (Computer-Aided-Software Engineering) software packages.
- **CASE refers to the automation of anything that humans do to develop systems** and support virtually all phases of traditional system development process.
- For example, these **packages can be used to create** complete and internally consistent **requirements specifications** with graphic generators and specifications languages.

(g) **System Components Matrix:**

- A System Component Matrix **provides a matrix framework to document** the resources used, the activities performed and the information produced by an information system.

- The activities are arranged in form of inputs, processing, outputs, storage and controls of information system and resources are represented as hardware, software users and data.
- Table below illustrates the use of a system component matrix to document the basic components of a sales processing and analysis system in an organization.

Information system Activities	Hardware Resources		Software Resources		People Resources		Data Resources	Information Products
	Machines	Media	Programs	Procedures	Specialists	Users		
Input	POS Terminals	Bar tags, Mag Stripe Cards	Data Entry program	Data Entry procedures		Sales clerks customers	Customer data, product data	Data entry displays
Processing	Mainframe Computer		Sales processing program, sales analysis program	Sales transaction procedures	Computer operators	Sales clerks managers	Customer inventory and sales databases	Processing status displays
Output	POS Terminals, Management Workstations	Paper reports and receipts	Report generator program, Graphic program	Output use and distribution procedures		Sales clerks managers, customers		Sales receipts, sales analysis reports and displays
Storage	Magnetic Disk Drives	Magnetic disk packs	Database management system program		Computer operators		Customer, inventory and sales databases	
Control	POS terminals, Management workstations	Paper documents and control reports	Performance monitor program, security monitor program	Correction procedures	Computer operators control clerks	Sales clerks managers customers	Customer, inventory and sales database	Data entry display, sales receipts, Error display and signals

(h) **(Nov 02/May 05/May 07/May 12) Data Dictionary:**

Meaning:

A data dictionary is a computer file which **contains descriptive information about the data items** in the files of a business information system.

Thus, a data dictionary is a computer file about data.

Content:

Each computer record of a data dictionary **contains information about a single data item** used in a business information system.

This information may include –

The identity of the source document(s) used to create the data item;

The names of the computer files that store the data item;

The names of the computer programs that modify the data item;

The identity of the computer programs or individuals permitted to access the data item for the purpose of file maintenance, upkeep, or inquiry; the identity of the computer programs or individuals not permitted to access the data item etc.

Additions:
As new data fields are added to the record structure of a **business file**, **information** about each new data item **is used to create a new record in the data dictionary**.
Similarly, **when new computer programs are created** that access data items in existing files, the **data dictionary is updated** to indicate the data items the new programs access.

Deletions:
Finally, **when data fields are deleted** from the structure of file records, their **corresponding records in the data dictionary are dropped.**

Usage:
Accountants and auditors can also make good use of a data dictionary.
For example, a data dictionary can help to **establish an audit trail** because it can identify the input sources of data items, the computer programs that modify particular data items, and the managerial reports on which the data items are output.
When an accountant participates in the design of a new system, a data dictionary can also be used to **plan the flow of transaction data through the system**.

(v) Systems Specification:
Nov 11: At the end of the analysis phase, the system analyst prepares a document called "System Requirement Specifications". What are the content of SRS.
A well documented SRS may normally contains the following sections:

- **Introduction:** This section describes the scope of the system to be developed, its goals and objectives etc.
- **Information Description***(Inputs/Outputs)***:** This describes the problem as well as information content required with flow charts and structure. It also describes hardware, software, human interfaces for external system elements and internal software functions. *(problem kya ha, uske lie information kya chaiye aur interfaces kya honge)*
- **Functional Description***(Process/Database)***:** This section describes about all the functions to be performed in a diagrammatic manner, Processing narrative for each function, Interplay among functions, Design constraints. *(Eg: Kaizen Costing)*
- **Behavioral Description:** This describes activities needed for connecting to external system and internal controls needed for security of system.
- **Validation Criteria:** Describes classes of tests to be performed to validate functions, performance and constraints.
- **Appendices:** Data flow/Object Diagrams; Tabular Data; Detailed description of algorithms charts, graphs and other such material.
- **SRS Review:** The development team **makes a presentation and then hands over the SRS document to be reviewed by the user** or customer.

The review reflects the development team's understanding of the existing processes.
Only, after ensuring that the document represents existing processes accurately, the user should sign the document.

(vi) Roles Involved in SDLC:

A variety of tasks during the SDLC are performed by special teams/committees/individuals based on requisite expertise as well as skills. Some of the generic roles are described as follows:

(a) **Steering Committee:** *(steering- name se hi samaj raha ha direct karenga)*

It is a special high power **committee of experts** to accord approvals for go-ahead and implementations. Some of the functions of Steering Committee are given as follows:

- To **provide overall directions** and **ensures appropriate representation** of affected parties;
- To **undertake corrective actions** like rescheduling, re-staffing, change in the project objectives and need for redesigning.

(b) **(Nov 11)Project Manager:** *(Eg: Boss of a CA firm)*

A project manager is normally **responsible for more than one project** and **liaisonsing with the client** or the affected functions.

(c) S/he is **responsible for delivery of the project deliverables within the time/budget***(responsible for giving audit report in time)* and periodically **reviews the progress of the project** with the project leader and his/her team. *(Partner conducts regular review of audit manager & audit article)*

(d) **Project Leader:** *(Eg: Audit Manager who have been allotted particular audit)*

The project leader is **dedicated to a project**, who has to ensure its completion and fulfillment of objectives.

S/he **reviews the project status more frequently than a Project Manager** and the entire project team reports to him/her.

(e) (Nov 11)Systems Analyst / Business Analyst:

The systems analysts' main responsibility is to **conduct interviews** with users and understand their requirements.

S/he is a **link between the users and the designers/programmers**, who convert the users' requirements in the system requirements and plays a pivotal role in the Requirements analysis and Design phase.

(f) **Module Leader/Team Leader:** *(Eg: Article given specific area in audit)*

A project is divided into several manageable modules, and the **development responsibility for each module is assigned to Module Leaders**.

For example, while developing a financial accounting application – Treasury, Accounts payable, Accounts receivable can be identified as separate modules and can be assigned to different module leaders.

Module leaders are **responsible for the delivery of tested modules** within the stipulated time and cost.

(g) Programmer/Developers:

Programmers are a mason of the software industry, who **converts design into programs** by coding using programming language.

Apart from developing the application in a programming language, they also **test the program for debugging activity** to assure correctness and reliability

(h) (Nov 11)Database Administrator:

The **data** in a database environment has to be **maintained by a specialist** in database administration so as to support the application program.

The DBA **handles multiple projects; ensures the integrity and security of information stored** in the database and also helps the application development team in database performance issues. **Inclusion of new data elements has to be done only with the approval of** the database administrator.

(h) Quality Assurance:

This team sets the **standards for development, and checks compliance with these standards** by project teams on a periodic basis.

(i) Testers:

Testers are a **junior level quality assurance personnel** attached to a project, **who test programs and subprograms** as per the plan given by the module / project leaders and prepare test reports.

(j) Domain Specialist: *(eg: Actuary is a domain specialist if you are making an application in insurance field)*

Whenever a project team has to develop an application in a field that's new to them, they take the help of a domain specialist.

For example, if a team undertakes application development in Insurance, about which they have little knowledge, they may seek the assistance of an Insurance expert at different stages.

This makes it easier to anticipate or interpret user needs.

A domain specialist need not have knowledge of software systems.

(k) (Nov 11)IS Auditor:

As a member of the team, IS Auditor **ensures that** the application **development also focuses on the control perspective.**

S/he should be involved at the Design Phase and the final Testing Phase to **ensure the existence and the operations of the Controls** in the new software.