SYSTEM DEVELOPMENT LIFE CYCLE

Chapter 2

System Development Life Cycle Methodology

Contains the system development processes in detail.

CA AKHIL MITTAL

CHAPTER 2

SYSTEM DEVELOPMENT LIFE CYCLE METHODOLOGY

A. SYSTEM DEVELOPMENT PROCESS:

System development refers to the **process of examining a business situation with the intent of improving** through the better procedures and methods. It has 2 major components:

I. **SYSTEM ANALYSIS:**
It is the process of gathering and interpreting facts, diagnosing the problems & recommends the improvement to the system.

II. **SYSTEM DESIGN:**
It is the process of planning a new business system.

B. SYSTEM DEVELOPMENT PROCESS:

Reasons that most of the organization fails to achieve their systems development objectives are:

CODE TO REMEMBER: **T-SUPPORT**

1. **T**echnologies (new):
   If an organization tries to create a competitive advantage then it has to keep it update with the changing environment by using the new technologies but **the personnel in the organization may not be able to make the best use of the technologies for the benefit of the organization.**

2. **S**upport lacking from management:
   The system developers watch closely the management action so as to distinguish the projects that are important and which are not.

3. **U**ser needs changing:
   User requirement for the information technology is changing day by day. Since these changes are frequent then it is difficult for the developer to implement these changes.

4. **P**roject standardization absence:
   Some organization doesn't follow the project management & system development methodologies as such the projects doesn't complete in time.

5. **P**articipation is less:
   User must participate in the development of system by intimating their requirement. As such there is less resistance to changes by the users.

6. **O**ver/under trained development staff:
   In certain cases, system developers lack **sufficient knowledge & education background.** As such the development of system may not be successful.

7. **R**esistance to changes:
   Some people in the organization have a tendency that they resist to changes. As such if the personnel interest gets above the organization's interest, then it is quite possible that the system will fail.

8. **T**esting & training inadequacy:
   New system must be tested first before implementing it in the organization. Users must be trained so that they can effectively use the new system.

C. SYSTEM DEVELOPMENT METHODOLOGY:

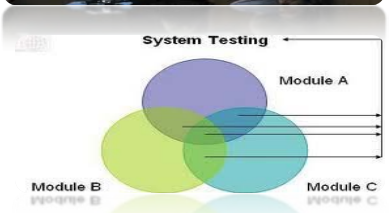
It is a **formularized, standardized, documented set of activities** used to manage **a system development project.** The methodology is characterized by the following:

1. *Project is divided in to a number of identifiable processes & each has a starting & ending point.*

2. *Specific reports & other documentation must be produced periodically to make development personnel accountable.*

3. Users including auditor, managers are required to participate in the project.

4. The system must be tested thoroughly prior to implementation.



5. Training plan for those who will operate & use the system.

6. A post implementation review of all developed systems.

D. APPROACHES TO SYSTEM DEVELOPMENT:

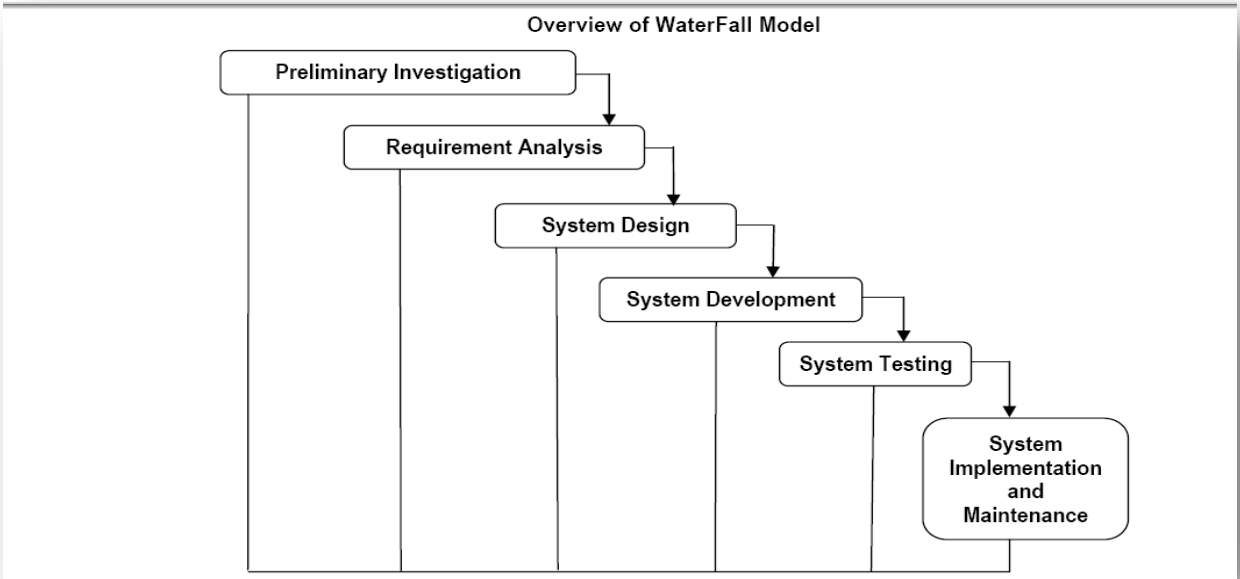
CODE TO REMEMBER: **W-P.A.I.R.**

1. **W**aterfall: **Linear Framework type**
2. **P**rototyping: **Iterative framework type.**
3. **A**gile methodologies
4. **I**ncremental: **Combination of linear & iterative type.**
5. **R**apid Application Development: **Iterative framework type.**
6. **S**piral : **Combination linear & iterative framework Type**

NOW EXPLAINING EACH MODEL IN DETAIL:

A. Traditional/ waterfall/ sequential approach:

- It is a traditional development approach.
- This phase includes **requirement analysis, specification & design req.**
- In this approach, an activity is undertaken only when the prior steps are fully completed.



Basic principles	Strengths
CPE	**CPQ**
1. Control is maintained over the project life through the use of extensive written documentation.	1. **Conserve resources.**
2. Phases are formed of the projects i.e. project is divided into sequential phases.	2.**P**rogress of the system development measurable.
3. Emphasis is on planning, time schedules, budgets etc	3. Quality is ensured since there is orderly sequence of development steps.

WEAKNESSES OF THE TRADITIONAL APPROACH

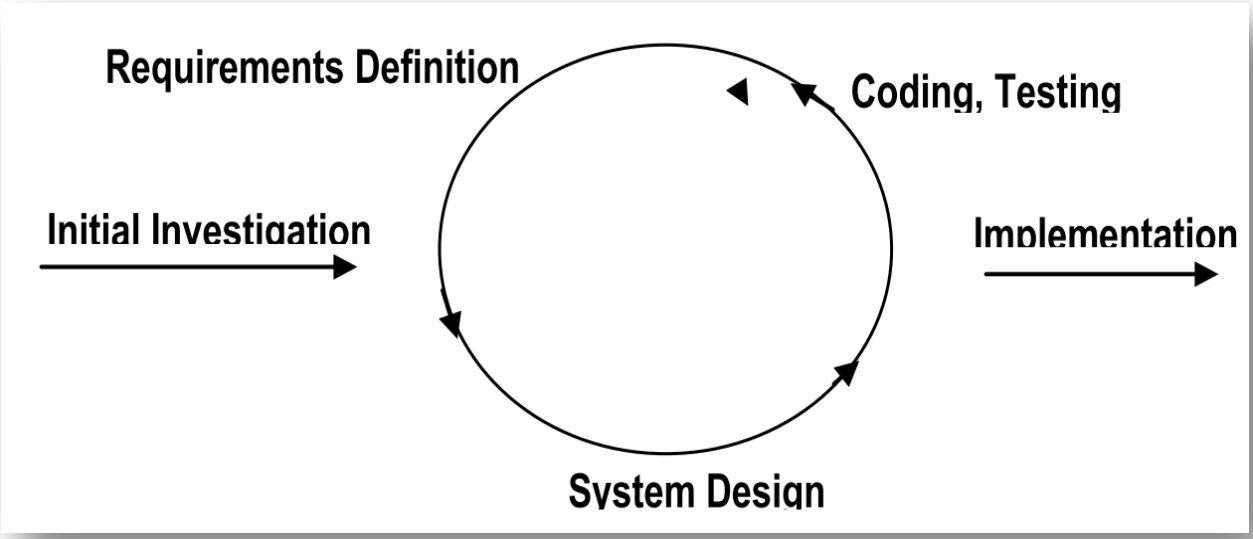
CODE TO REMEMBER: **W.I.R.E.D.**

1. **W**ritten specifications are often difficult to read & to be understood by the users.
2. **I**nflexible, slow, costly due to excessive tight controls.
3. **R**esistance to changes. It means if any change to be done at the end of the system lifecycle is discouraging & cumbersome.
4. **E**arly identification of the system specification requirement is the crux of this approach. But users are not able to point out their requirements in early phase.
5. **D**ocumentation is excess in this approach is difficult & also to update the same.

B. Prototyping Model:

- The objective of this approach is **to develop small/pilot version** called a **prototype.**
- It is a usable system that is built quickly & at lesser cost.
- One major benefit is that if the prototype works as per requirement then it is turned into final system otherwise it is scrapped.

Basic principles	
1. Identifying system Requirement:	In prototype approach, developing team needs only fundamental system requirements to build initial prototype.
2. Develop the initial prototype:	Here the designer create initial base for the system. The main emphasis is on to ease the interaction between users & system by using display screen, menus, input prompts & source documents.



3.Test & Revise:	Designer first demonstrate model to the users & allow them to experiment. After getting feedback of the users designer resubmits the modified model.
4. Obtain User Approval:	Users formally approve the final version of the prototype.

STRENGHTS OF PROTOTYPE MODEL	
Code to remember : F.L.U.T.E.	**Code to remember : O-K.I.D.S.**
1.Flexible designs & innovation.	**1. O**bjective which are unclear, experimenting, investigation issue s are resolved
2.Liaisoning between users & system & communication among project stakeholders	**2. K**nowledge is gained in an early iteration as later iterations are developed.
3.User oriented system since there is high involvement of the users.	**3. I**mplementation of the incomplete application applications.
4.Time involved in developing the prototype is rather less.	**4. D**ifficult or confusing functions are easy to identify.
5.Errors here are detected early since prototype is made with effective interaction with the users.	**5. S**pecifications generation for production application.

WEAKNESSESS OF PROTOTYPE MODEL
Code to Remember: R.B.C.-L.I.T.E.
[means if **RBC** is less(**lite**), then weakness]
1. Requirement changes: Requirements of the user may change frequently. As such it is difficult to implement the changes frequently in the prototype model.
2. Behavioral problems: It means dissatisfaction among the users of the system if the system is unable to meet all user demands. Also dissatisfaction arises when there is too much interaction for the prototypes.
3. Control not strict: Approval process and control are not strict.
4. Lack of proper testing & documentation: Interactive processes of prototype model to be experimented extensively. As such the developer got less time for testing & documenting the processes of final approved information system. Inadequate testing may led to **error prone** system.
5. Identification of non functional elements difficult to document.
6. Time devotion less: Prototyping can be success if the users are willing to devote their time in experimenting with the prototype. However it might be possible that users are not willing to spend time as required.

C. Agile Methodologies:

- It is group of software development methodologies based on **iterative & incremental development,**
- Where solutions are obtained through collaboration between **self organizing & in teams.**
- It promotes the adaptive planning, development & delivery of software.

PRINCIPLES OF AGILE METHODOLOGIES

Code to Remember: D.O.S.- C.A.S.E.

1. **Development**:
   Development is maintained at a constant pace.

2. **Organized Team:**
   One of the major benefits of the agile methodologies is that it has self organized team which can tackle all the problems.

3. **Simplicity**

4. **Cooperation:**
   There is close & daily cooperation between business people & the developers.

5. **Adaptive to changes:**
   Regular adaptive to changing circumstances.

6. **Satisfaction to customer:**
   Customer satisfaction by rapid delivery of useful software.

7. **Excellence:**
   There is continuous attention to the technical excellence & good design.

STRENGHTS OF AGILE METHODOLOGIES

Code to Remember: Q-C.A.D.

1. **Quality software** :
   One of the benefits of this methodology is the delivery of **high quality software** in least possible time & satisfied customer.

2. **Communication is effective:**
   Face to Face communication & continuous inputs from the users leaves no probability of any work based on any guesses.

3. **Adaptive team:**
   Agile methodology has the concept of the adaptive team, which responds quickly to the changing requirements.

4. **Documentation:**
   Here the document is crisp(less) & to the point hence saving time.

WEAKNESSES OF AGILE METHODOLOGIES

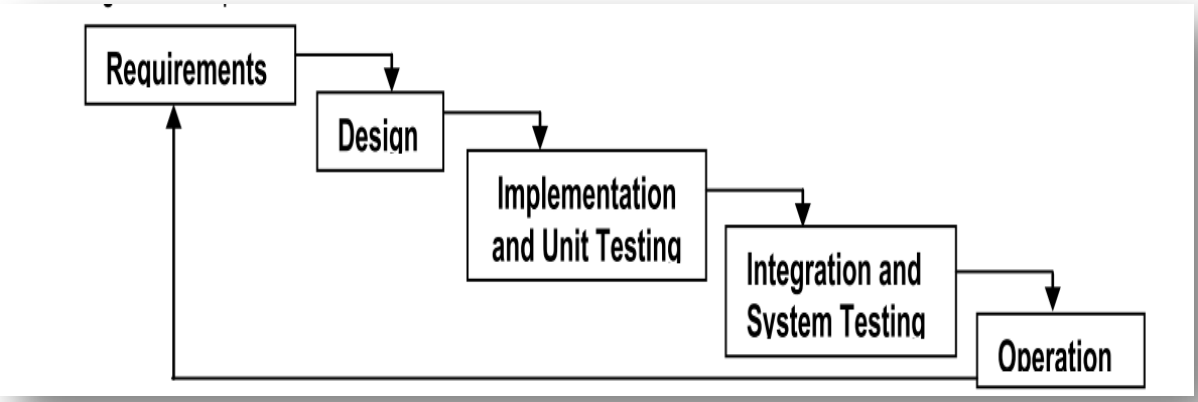
Code to Remember: V.A.R.I.E.D.

1. **Verbal communication:**
   Under agile methodology, more emphasis is on verbal communication with customer rather than documenting or preparing manuals.

2. Accessing incapability: In case of software which is rather large in size, it is difficult to access the **efforts** required in the beginning of software development life cycle.
3. Rework required: Agile require rework. Because of lack of long term planning, rework is often required to integrate the agile project with other components of the software.
4. Integration difficulty: Agile lacks attention to outside integration. While developing the agile methodologies the developer don't give time to integration points with other systems in advance.
5. Experienced programmers: Only senior programmers are capable of taking any decision during the development process.
6. Designing & documentation: There is less emphasis on the designing & documentation.

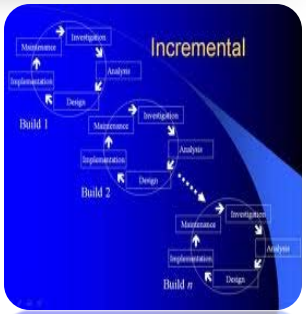
D. Incremental Model:

- Here the model is designed, implemented and tested incrementally **(little is added each time)** until the product is finished.
- **INCREMENTAL MODEL = WATERFALL + PROTOTYPE MODEL.**



F

Following procedure is adopted under this approach:



!! A product is decomposed into number of components.

!! Each component is delivered to clients when it is complete.

As I already mentioned that it is a mixture of both waterfall & prototype, I hereby explaining the reason for the same:

A series of mini **waterfalls** are performed, whereby all parts of the waterfall development model are completed for a small part of the system.

OR

Overall requirements are to be defined before .proceeding to next individual increment model.

OR

The initial concepts, requirement analysis & design architecture is defined under **waterfall model** and followed by iterative (repetitive) **prototyping.**

STRENGHTS OF INCREMENTAL MODEL
Code to Remember: D.M.-KFC
1. Delivery of system : One of the benefits of this methodology is the delivery of system to its customer as and when the component of the system is ready.
2. Monitoring the changes: Gradual implementation of the components of the system
3. Knowledge : Potential exists for the exploitation or using the knowledge gained in early incremental phases as later incremental phases are developed.
4. Flexibility: This type of model are indeed flexible & less costly if requirement or scope changes.
5. Control Maintained: Moderate control is maintained over the project life by having proper & written documentation. It's just like blue print of project on which basis project deviation, if any, is measured & controlled.

WEAKNESSES OF INCREMENTAL MODEL
Code to Remember: O-R.A.W.
1. Overall consideration lacks: There is usually lack of consideration of the business problems & technical requirements when the concentration is on the smaller parts/components of the system.
2. Rigid: Each phase of an iteration is RIGID & do not overlap each other.
3. Architectural issues: Problems may arise relating to architectural since at the time of developing the system not all requirements are known.
4. Well defined interface required: Since some modules are completed much earlier than others as such well defined interface required.

E. RAPID APPLICATION DEVELOPMENT (RAD):

- It is type of software development methodology which uses **minimal planning** in favor of **rapid prototyping.**
- Since the element of planning is minimal in this approach, **the softwares are developed at fast rate & are adaptable.**

PRINCIPLES OF RAD
Code to Remember: F.U.N-Q-D.R.P
1. Fast development of system: Since we have already discussed that the element of planning is minimal, as such this approach is able to produce the high quality system in a quick time at a very low investment cost.
2. User involvement: Active user involvement is **imperative (necessary)**.
3. Needs of business: In this approach key emphasis is on the **business needs** rather than on the technical or engineering issues.
4. Quality system: Aims to produce quality system quickly, through the use of iterative prototyping & various development tools i.e. **DBMS, Graphical user interface, Codes etc.**

5. Documentation:
Produces **documentation** necessary to facilitate future **development & maintenance.**

6. Risk reduces:
Aims to reduce **inherent project risk** by breaking the project into smaller **segments** & providing **ease** in changing any requirement during development phase.

7. Prioritization criteria :
Project control involves **prioritizing development** & defining deadlines for the projects undertaken.

STRENGHTS OF AGILE METHODOLOGIES

Code to Remember: O-I.C.-SRS

OH!!! I C(SEE) SRS cinema

1. Operational version of application:
Under the RAD approach, operational version of an application is available much faster & earlier than other approaches.

2. Initial review possible:
Quick initial reviews are possible.

3. Cost:
RAD produces system more quickly, as such this approach tends to produce systems at **low cost.**

4. Stakeholder involvement:
Since the user's involvement is much more under this approach, as such there is indeed great level of commitment & involvement. Because if such involvement **users** feel sense of **ownership** & developers feel **satisfied** by developing user based system.

5. Rapid changes possible:
Provides the ability to rapidly changes the system design as required by users.

6. Saving:
RAD helps in saving time, money and human efforts.

WEAKNESSES OF RAD

Code to Remember: Q-MR-VSD

Kyu(Q) – MR -- VaSuDev

1. Quality compromise:
More speed & lower cost may lead to **lower overall system quality.**

2. Misalignment: Since system is developed in a quick time as such it might be possible that sufficient information is not there with the developers. AS such there may be possibility that **developed system may get MISALIGNED** i.e. from the standard path.
3. Reuse difficulty: Modules which are constructed in a quick time lacks usability in future years.
4. Violation of standards: There may be chances where violation of programming standards is there. It involves **inconsistent documentation etc.**
5. Requirement issues: Since in RAD it is possible to incorporate changes rapidly as required by users, as such it might be possible that **project may end with high requirements** than needed.
6. Design inconsistency: Potential for design inconsistency within and across the system.

F. Spiral Model:

PRINCIPLES OF SPIRAL MODEL
Code to Remember: D.O.S.- C.A.S.E.
It is a software development process combining elements of both design & prototyping in stage. It is also known as: ❖ **Spiral Lifecycle.** ❖ **System Development Method.** **SPIRAL MODEL = WATERFALL + PROTOTYPE MODEL.** Spiral Model is generally used for **large, expensive & complicated projects.**

Following procedure is followed:

New system requirements are defined

be defined in detail.

It involves interviewing number of users.

It represents both external / internal users.

Preliminary Design is created for new system

system. Here:

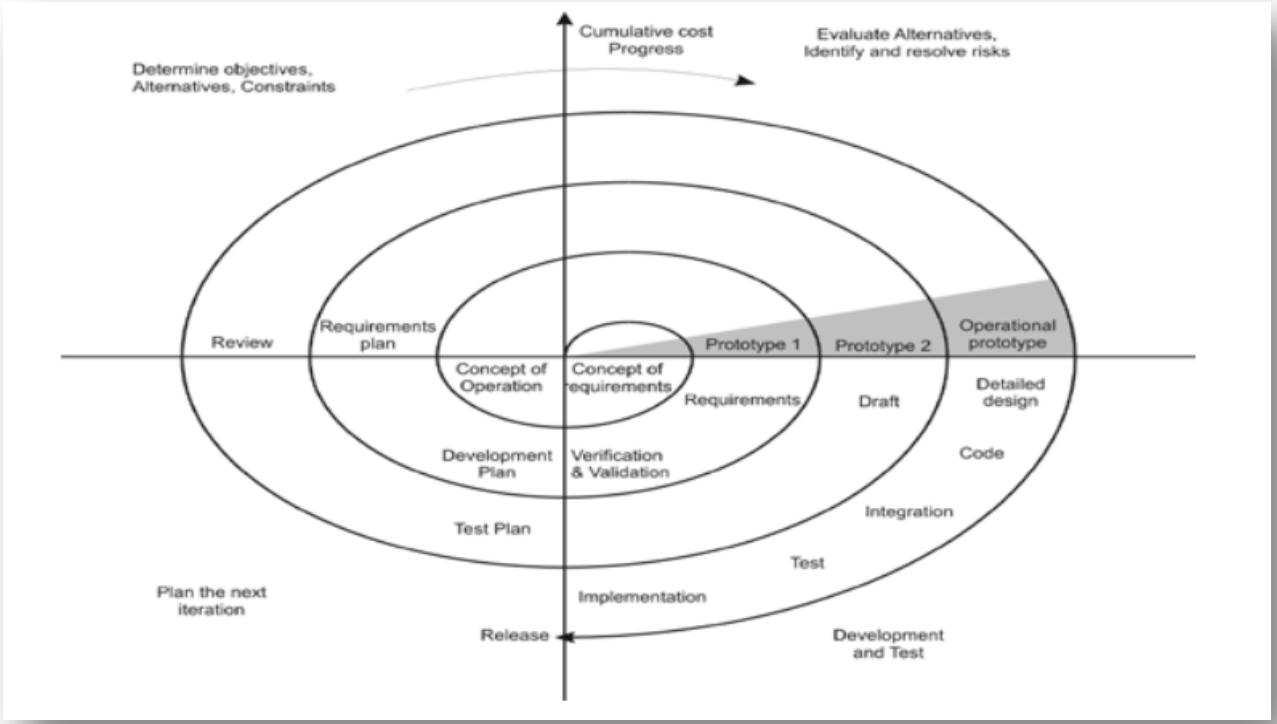
1. All possible alternatives are analysed & strategies are decided thereon.
2. It includes identify & resolving the possible risks in the project development.

Prototype is constructed for the new system

system. Also **SECOND** prototype is constructed by a defined procedure:

1. **Doing SWOT** of first prototype.
2. Define requirement of second prototype.

Plan, design & construct the second prototype.



STRENGHTS OF SPIRAL MODEL

Code to Remember: R.B.I.

6. **Risk avoidance** :
   Enhance risk avoidance.

7. **Best Methodology selecting tool:**
   Useful in selecting the best methodology to follow for development of given software.

8. **Incorporation of other methodologies :**
   Can incorporate **waterfall, prototype & incremental methodologies**. It provides guidance in knowing the best combination of the system models & software.

WEAKNESSES OF SPIRAL MODEL

Code to Remember: L.E.T.S-C

1. **Limiting reusability:**
   Such model is highly customized to project. Thus it is complex & limits reusability.

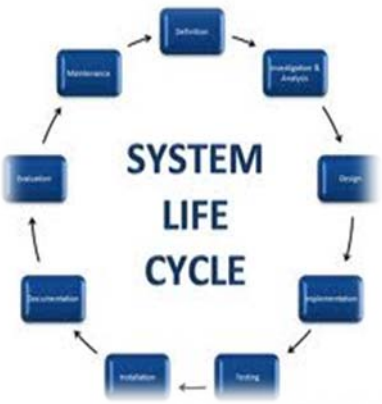
2. **Exact composition not measurable:**
   It is difficult to determine the exact composition of development methodologies to use for each cycle.

3. **Termination point absence:**
   There is no clear termination point or deadline between the cycles. One cycle continues after other hence **an inherent risk of not meeting budget or schedule.**

4. **Skilled manager required:**
   A skilled & experienced **project manager** required to determine how to apply this model in a given practical business situation.

5. **Control absence:**
   There is lack of established control from moving one cycle to another cycle.

E. SYSTEM DEVELOPMENT LIFE CYCLE (SDLC):



1. SDLC provides **system designers & developers** to follow a sequence of activities.
2. SDLC consists of various phases **where 1 phase uses result** of previous phase.
3. SDLC is document driven i.e. **at crucial stages during the** process, **documentation is produced.**

ADVANTAGES OF SDLC SYSTEM

1. Better planning & control by the project manager.
2. Compliance to prescribed standards.
3. Documentation is the imperative part of SDLC.
4. Phases form important part of the SDLC and enable users & project manager to review.

ADVANTAGES OF SDLC SYSTEM i.r.o. IS AUDIT

A. Since there is **documentation** at every phase of the SDLC, as such **IS auditor** can have clear understanding of various phases of the SDLC.

B. **IS auditor** can state in the **report the compliances** by IS management of standards set by the organization.

C. IS auditor if has **technical knowledge,** can be guide during SDLC phases.

D. IS auditor can provide **techniques** used through various development phases of the SDLC.

PHASE NO.	PHASE NAME	NATURE OF ACTIVITY
1.	Preliminary Investigation	Determining and evaluating the strategic benefits of the system and ensure that the solution fits the business strategy. Includes cost-benefit analysis of the proposed system.
2.	Systems Requirements Analysis	Analyzing the type of the system on the basis of the users requirements.
3.	Systems 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.
4.	Systems Development/Programming	Programming the system as designed and conduct the continuous testing and debugging.
5.	Systems Testing	Various kinds of testing is conducted before the developed system is implemented. This includes Unit Testing, Integration Testing and System Testing etc.
6.	Systems Implementation	Final Testing and quality of controls audit, acceptance by management and user before migration of the system to the live environment and data conversion from legacy system to the new system.
7.	Post Implementation Review and Maintenance	Continuous evaluation of the system as it functions in the live environment and its updation. Maintenance includes continuous evaluation of the system as it functions in the live environment and its updation.

Now explaining each stages of SDLC in detail:

Now I am explaining a way to remember these phases in chronological order in Hindi

MAINE FIRST OF ALL ANALYZE KIYA: BENEFITS OF SYSTEM

REQUIREMENT PATA KI OF USERS: USER REQUIREMENT

DESIGN KIYA SYSTEM USER KE HISAB SE: DESIGNING THE SYSTEM

PROGRAMMING KI BY USE OF CODES: PROGRAMMING THE SYSTEM

SYSTEM KO TEST KIYA AFTER MAKING: VARIOUS KINDS OF TEST DONE

SYSTEM IMPLEMENT KIYA TEST KARKE: FINAL TESTING & MGMT APPROVA

IMPLEMENT KIYA OR MONITOR KARTE RAHE: POST IMPLEMENTATION REVIEW

PRELIMINARY INVESTIGATION

PHASE I

Objective: **To determine & analyze the strategic benefits in implementing the system.**

STEPS INVOLVED IN PRELIMINIARY INVETSIGATION:

CODE TO REMEMEBER: P.O.S.-F.M.

(I) Identification of Problems.

- First step is to define the problems clearly & precisely.
- Problems that affect organization largely are **likely to receive** immediate **management attention.**
- The analyst working on the **preliminary investigation** must accomplish the following objectives:

CODE TO REMEMBER: U.S.-F.C.A.

1. **Understanding Project request.**
2. **Size of the project is determined.**
3. **Feasibility study made.**
4. **Cost & benefit analysis.**
5. **Acquaintance with the management & reporting the findings & giving recommendations.**

(II) Identification of Objectives.

- After the problem is identified, then it is easy to work out the **objectives** of the proposed solution.
- For instance, **more time involved in doing reservation in railway,** as such the objective-"*To introduce a system whereby passengers could buy the ticket online faster & in real time*"

(III) Delineation(Sketching outline) of scope: [FDR CIR]

- When a solution is defined, scope of work gets defined.
- While stating the scope, following things to be kept in mind:

1. **F**unctional requirement: Functionalities to be delivered by solution obtained.
2. **D**ata: What data required to achieve these functionalities.
3. **R**equirement of control: What control requirements for the given application.
4. **C**onstraints: What are constraints? **For example** number of character in database.
5. **I**nterfaces: Is any special interface software/hardware needed to enhance interface.
6. **R**eliability: Application reliability is measured if it remain uncorrupted in case of malicious use of the system.

- There are primarily 2 methods by which a project can be analyzed:

1. Reviewing internal documents:

The analyst will first of all analyze the working of the organisation. This can be done when analyst **examines organisation charts & written operating procedures.**

2. Conducting Interviews:

Written document tells analyst what to do. Also there is no **users view** present in the documents. So **analyst uses INTERVIEWS.**

(IV) Feasibility Study:

TYPES OF FEASIBILTY STUDY
Code to Remember: B.E.S.T. – F.L.O.R.
1. Behavioral : Is solution going to bring any adverse effect on quality of work life.
2. Economical: Return on investment.
3. Schedule/ time: Can be system delivered on time.

4. Technical : Is the technology needed by the organization is **available or not?**
5. Financial: Is the solution viable financially?
6. Legal: Is the solution valid in **legal terms?**
7. Operational: How will the solution work?
8. Resources: Are human resources reluctant for the solution?

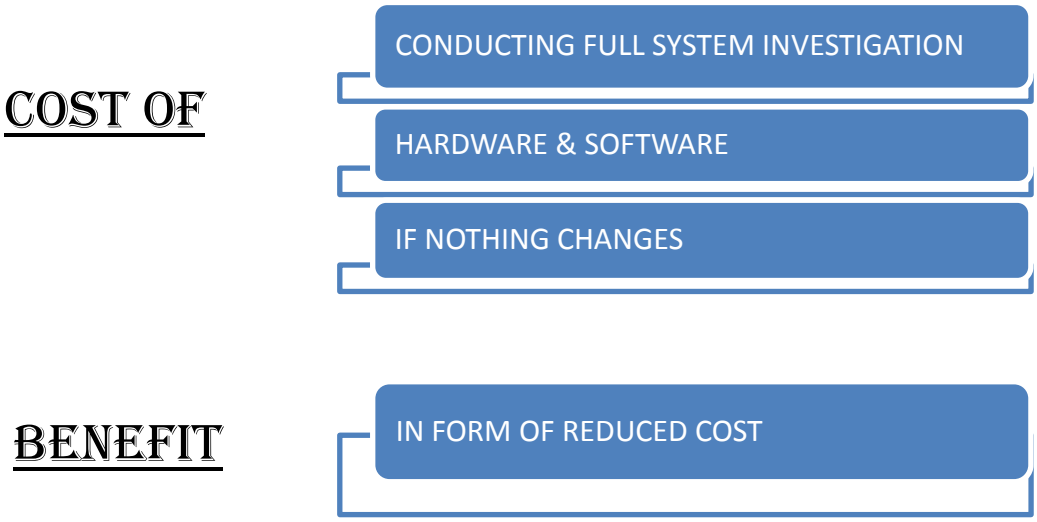
Now I am hereby explaining each in detail:

A. BEHAVIORAL FEASIBILITY:

- ❖ It is system designed to process data & produces the desired output.
- ❖ If the data is not easily available or collectable, system may not succeed.

B. ECONOMICAL FEASIBILITY:

- ❖ Includes evaluation of incremental cost & benefit expected if the proposed system is implemented.
- ❖ Following questions may be raised during economical feasibility:



❖ Types of system cost:

DEVELOPMENT COST

Includes salaries of system analyst, cost of preparing data files, cost of testing etc.

OPERATING COST

Includes the hardware/ software rental/depreciation charges, salaries of various computer personnel, etc

INTANGIBLE COST

Are the cost that can't be easily measured. E.g.: **New system may indirectly lower the employees morale**

C. SCHEDULE FEASIBILITY:

- ❖ It means estimated time to be taken by the system designer to make the system operative.
- ❖ After reviewing this time issue, management can accept or reject the proposal.

D. TECHNICAL FEASIBILITY:

- ❖ Concerned with issues relating to **hardware & software.**
- ❖ Following issues may erupt in the technical feasibility phase:
  - **Is technology exist or available as suggested?**
  - **Is NEW SYSTEM is capable to hold voluminous data?**
  - **Will new system successfully responds to queries, enquires etc.**

E. FINANCIAL FEASIBILITY:

- It might be possible that the proposed system may prove costly to the organization.
- This may be checked if **proper financial feasibility study is taken up.**

F. LEGAL FEASIBILITY:

- It is concerned with evaluating conflict between **new proposed system & organizational legal obligation.**
- If there is violation of laws due to **new proposed system,** then it should be rejected.

G. OPERATIONAL FEASIBILITY:

- It is concerned with ascertaining the view of workers, employees & suppliers about the use of **computer facility.**

H. RESOURCES FEASIBILITY:

- This focuses on human resources.
- If we implement sophisticated system in a rural area then obviously there will be lack of **trained personnel.**

(V) Reporting to Management:

- After analyst identify the problem & understood the scope,
- Next step is to **provide one or more solution alternatives,**
- Cost & Benefit analysis& **reporting the same to management.**
- REPORT+ COVER LETTER+RECOMMENDATIONS.
- It is from the report management **decide what to do or not.**

System Requirement Analysis

Phase II

Objective: To have thorough & detailed understanding of current system. Following activities are performed under this phase:

- ☐ Identify areas that need notification.
- ☐ Consult the stakeholders to determine their expectation.
- ☐ **To gather data, facts etc.** through fact finding techniques.
- ☐ To document all the procedure adopted to gather data.
- ☐ To verify that requirements are complete, verifiable, and testable.

Code to remember: T.C.P.S.

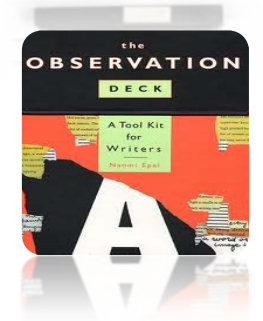
The CP's [The Connaught Place's]

1. Techniques of fact finding:

Code to Remember : D.O.-.I.Q.

A. Document:

- **Includes manual, input forms, diagrams, organizational chart.**
- **Documents are very good source of information about user needs & current system.**



B. Observation:

- **Observation is an important tool in knowing the facts.**
- **For Instance, an analyst watches the reaction of users using new system & accordingly develop requires system application.**

C. Interview:

- **Users can be interviewed in order extract relevant information.**
- **Through interview system analyst can came to know about the problems and opportunities.**

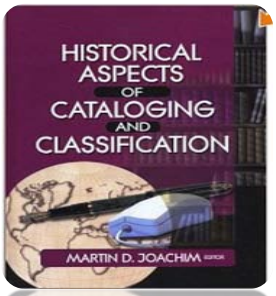
D. Questionnaire:

- Users & managers are asked to complete questionnaire,
- About the information system.
- Large amount of data can be collected through this process from different users.

2. Current system Analysis:

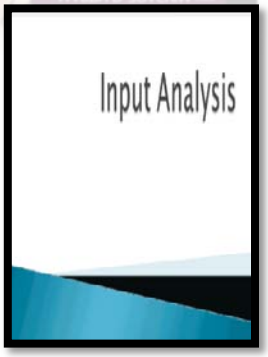
Code to Remember: H.I.D.-M-O.I.M.A.
HIDe Mom Ohh.....!!! IMA

A. Historical aspect review:



- Review of the organization is the starting point for analysis of current (present) system.
- Review of annual reports/Org. charts etc.

B. Input analysis:



- Detailed analysis is imperative for manipulating data.
- System analyst must analyze each source of data.
- System analyst must thorough investigate in order to determine if DATA can fit in present framework.

C. Data files maintained reviewed :

- Analyst should review files maintained by each department.
- He should review both on-line & offline files maintained in the organization.
- Cost of retrieving such data must be considered.

D.Method, procedure & data communication review:

-

METHODS: It is defined as a way of doing something.

PROCEDURE: A procedure review is **intensive survey of the methods by which each job is accomplished.**

DATA COMMUNICATION: System Analyst must review the types of data communication equipments which include **data interface, modems, dial up, leased lines etc.**

E. Output analysis:

- **The output should be scrutinized carefully,**
- **In order to determine how company needs fulfilled.**
- **Analyst must know that which information is needed & who will use.**
- **Irrelevant information must be ignored.**

F.Internal control review:

- **For complete present system analysis. It is imperative that internal controls are reviewed.**
- **Locating the control points helps analyst to know the essential framework of the system.**
- **Through this review weaknesses in the present system can be known & are rectified accordingly.**

G. **M**odel the existing & logical system:

- **As inputs, methods, procedures, reports, internal control etc are reviewed,**
- **Then the process must be documented.**
- **FLOW CHART & DIAGRAMATIC models may be used to present the information.**
- **Helps in disclosing gaps & duplication in the data gathered.**

H. **A**nalysis of overall present system:

- **Final phase of detailed investigation includes:**
  - -- Analysis of present work Volume.
  - -- Analysis of current personnel requirement.
  - -- Present cost & benefits analysis.

3. Proposed system Analysis:

- The proposed system specification must be properly defined.
- The required system specifications should be in conformity with the project's objective:
  - **Output produced in time & timely managerial report.**
  - **Database to be maintained with online processing capabilities.**
  - **Input data prepared directly from the source documents.**
  - **Methods that depicts the relationship between the input & outputs to the database.**
  - **Work volumes to be carefully considered.**

4. System Development Tools:

There are various development tools available for improving the current system & developing the new systems. Such tools help users & developers in:

CODE TO REMEMBER: C.A.P.

A. CONCEPTUALISE, clarify, document & communicate the activities.

B. ANALYSE present business operations, management decision making & information activities of the organization.

C. PROPOSE and design new/improved systems to solve the business problems.

Following are the system development tools:

1. SYSTEM COMPONENT AND FLOWS:

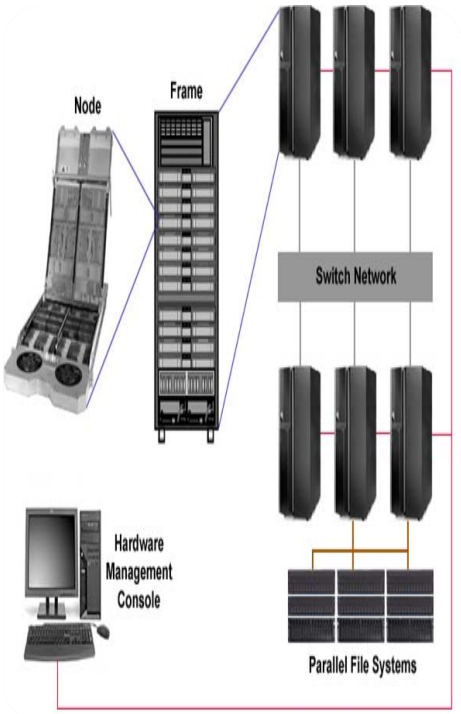
This tool help analyst in following ways:

SYSTEM FLOW CHART

1. **Document the data flow among major resources & activities of Information system.**
2. **System flowcharts* show** the flow of data media as **processed by hardware & manual devices.**

DATA FLOW DIAGRAM

1. **Data flow diagram*** uses a few **simple symbols** to show the flow of data among external entities.



2. USER INTERFACE:

USER INTERFACE

1. It is a major consideration of system analyst.
2. It is through proper interface system by which **users can interact with the system.**
3. LAYOUT & FORMS **are used to construct the format of both** INPUT /OUTPUT **media.**

DIALOGUE FLOW DIAGRAM

1. It analysis the flow of dialogue between **computers & people.**
2. They document **flows among different display screen** generated by end users.

3. DATA ATTRIBUTES & RELATIONSHIP:

Here data resources are defined & designed by this tool:

DATA DICTIONARY

It catalogs the description of the characteristic of **all data element &** their relationship to each other.

ENTITY-RELATIONSHIP DIAGRAM

It is used to **document the number & type** of relationship among the entities of the system

FILE LAYOUT

Forms document the **type, size & names** of the data element.

4. DETAILED SYSTEM PROCESS:

- Tool used to help the programmer develop,
- **DEATILED procedures & processes,**
- **Required in the computer program design.**

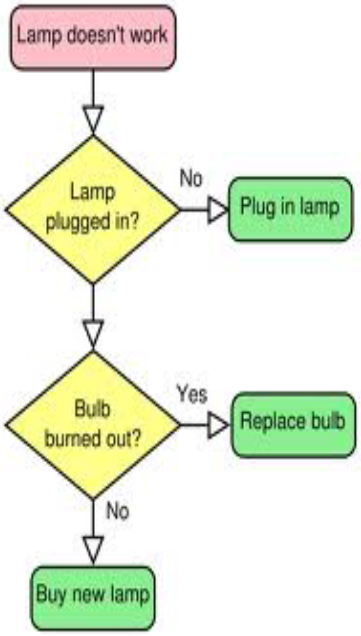
Here are some tools in details:

A. STRUCTURED ENGLISH:

- Also known as **Program Design Language (PDL)** or **Pseudo Code**.
- **STRUCTURED ENGLISH is use of English language with syntax of structured programming**.
- It helps in getting benefit of both **programming logic & natural language**.

B. FLOWCHARTS:

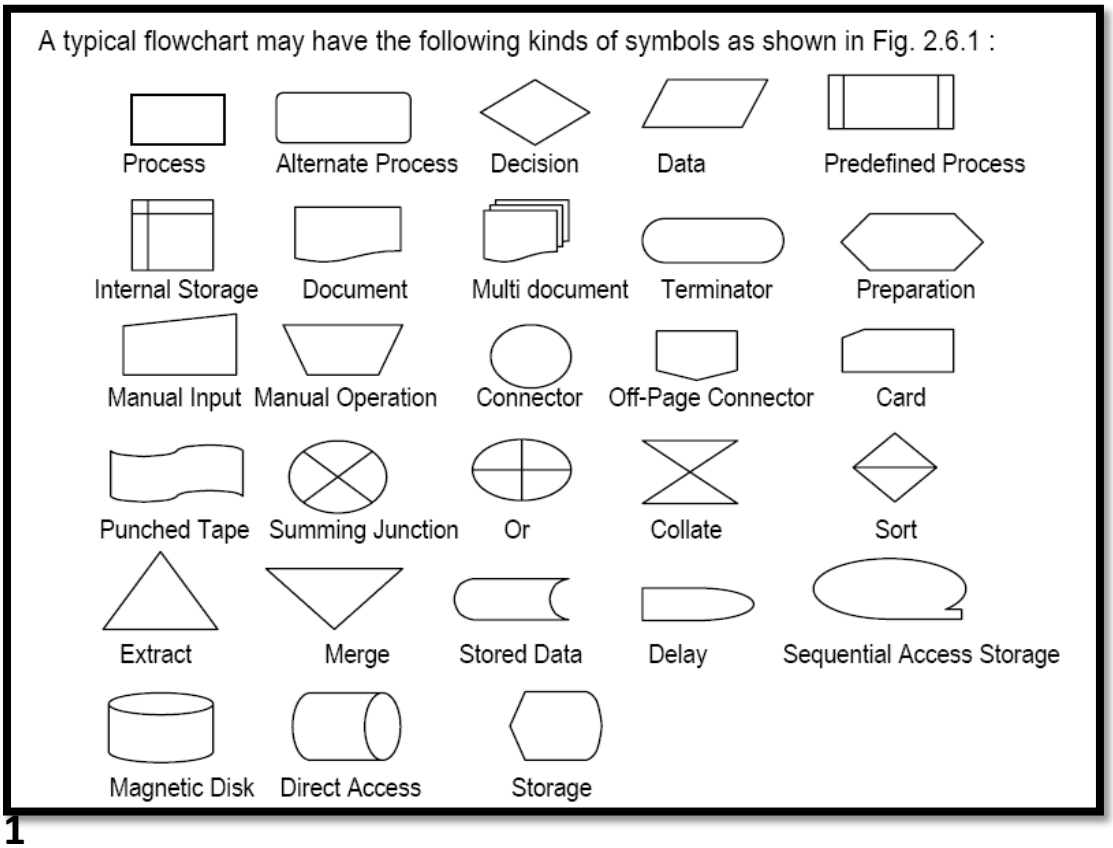
- It is graphic technique that is used to represent **input, output & processes** of a business in a **pictorial form**.



☺ **Common type of chart, representing an algorithm or process,**

☺ **Showing the steps as boxes & their order by connecting with the arrows.**

☺ **Flowcharts are used** in **analysing, designing, documenting** process or program in various fields**.**



1

B.1. Types of flowchart:

CODE TO REMEMBER: D.-D.S.P.

1. **Document Flowchart:** Shows a document flow through system.
2. **Data Flowchart:** Shows data flows in the system.
3. **System Flowchart:** Showing controls at physical/ resource level.
4. **Program Flowchart:** Showing controls in a program within a system.

B.2. Benefits of flowchart:

CODE TO REMEMBER: C^2D^2 - AP

CD OF AP (ASSISTANT POLICE)

1. **Communication:** Flowcharts are better way to communicate system logics.
2. **Documentation:** Program flowcharts serves as a good program documentation
3. **Coding:** Flowcharts act as a guide during system & development analysis.
4. **Debugging:** Flowcharts help in debugging processes.
5. **Analysis:** With the flowchart, problem can be analyzed more effectively.
6. **Program maintenance:** Flowcharts help the programmer to put more efforts.

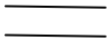
B.3. Limitations of flowchart:

CODE TO REMEMBER: C.A.R.

1. **Complex Logic:** Often program logics are quite complicated so as flowcharts.
2. **Alteration/Modifications:** If alteration require, re-drawing of flowchart required
3. **Reproduction:** As flowcharts symbols can't be typed, reproduction of flowchart becomes a problem

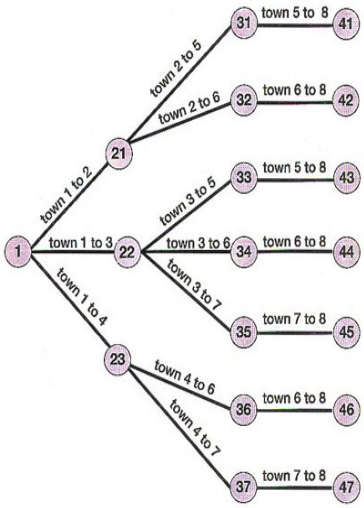
C. DATA FLOW DIAGRAM:

- It uses simple symbols to show the flow of data,
- Among external entities **(both people, organization)**.
- **4 components of the DFD:**

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	The flow of data into or out of a process is represented by curved or straight lines with arrows.
	Transformation process	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.

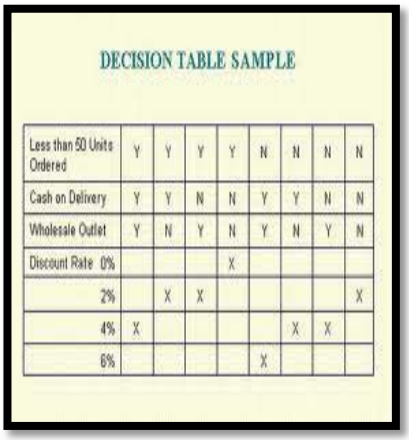
D. DECISION TREE:

- It is a supportive tool that **uses TREE-LIKE** graph/model of design.
- It is commonly used in **operational research**(decision analysis),
- To identify strategies **most likely to reach the goal**.
- Refer ICAI module for Examples.



E. DECISION TABLE:

- Accompanies the flowchart.
- Defining the possible contingencies that to be considered within the program
- 4 parts of decision table:
  -- **Condition hub** -- **Action Hub**
  -- **Condition Entries** -- **Action Entry**


DECISION TABLE SAMPLE

Less than 50 Units Ordered	Y	Y	Y	Y	N	N	N	N
Cash on Delivery	Y	Y	N	N	Y	Y	N	N
Wholesale Outlet	Y	N	Y	N	Y	N	Y	N
Discount Rate 0%				X				
2%		X	X					X
4%	X					X	X	
6%					X			

F. CASE TOOLS:

- **DFD & System flowcharts** are generated by system developers,
- Using ON-SCREEN **drawing modules found in CASE**
- **CASE** = **C**omputer **A**ided **S**oftware **E**ngineering
- CASE refers to **automation of anything.**


Development of CASE(Computer Aided Software Engineering) Tools

• CASE tools for SA/SD[21](e.g. StP)

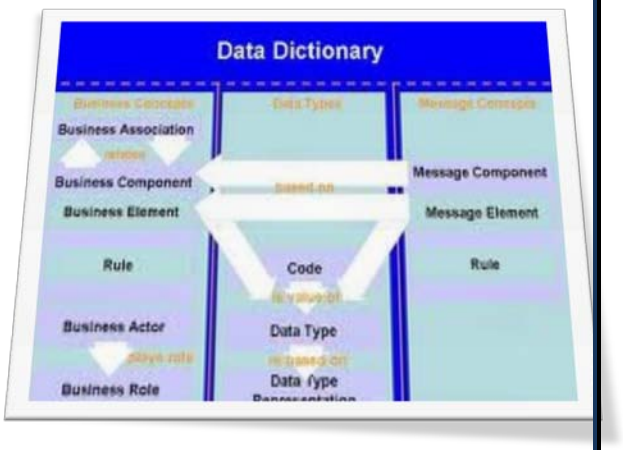
[21] Meilir Page-Jones: **The Practical Guide to Structured System Design**, Prentice-Hall Englewood, N.J., (1988).

G. SYSTEM COMPONENT MATRIX:

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 custom-mers	Customer data, product data	Data entry displays
Processing	Mainframe Computer		Sales processing program, sales analysis program	Sales transaction procedures	Computer operators	Sales clerks mana-gers	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 mana-gers, custom-mers		Sales receipts, sales analysis reports and displays
Storage	Magnetic Disk Drives	Magnetic disk	Database manageme		Computer operators		Customer, inventory	
		packs	nt system program				and sales databases	
Control	POS terminals, Management workstations	Paper documen ts and control reports	Performanc e monitor program, security monitor program	Correction procedures	Computer operators control clerks	Sales clerks mana-gers custom-mers	Customer, inventory and sales database	Data entry display, sales receipts, Error display and signals

- Provides MATRIX framework to document the resources used,
- Activities performed, information produced by information system.
- Highlights the basic activities of inputs, processing, output, storage & control, hardware, software that transform data into **useful information.**

H. DATA DICTIONARY:



- DATA DICTIONARY is a computer file that contains **descriptive information** about data items in the files of a business information system.

- **DATA dictionary contains various types of information:**

  -- ***Identity of source documents.***

  --***Names of computer files that store data.***

  --***Identity of programs or systems permitted or not permitted to be accessed.***

- There is deletion/ addition in the data dictionary on a regular basis. Each information has corresponding data field. In case a data field is deleted, their corresponding record in data dictionary is **DROPPED.**

Name of data field	File in which stored	Source document	Size in bytes	Type
Inventory quantity on hand	Inventory master file	Form number ABC 123	4	Numeric

← Data dictionary

Usefulness of the **data dictionary for:**

AUDITOR

- **Helps in establishing audit trail as input source of data is identifiable.**

ACCOUNTANT

- **It is helpful for the accountant in the sense that it can be used to plan the flow of transaction data through the system.**

I. LAYOUT FORMS & SCREENGENERATOR, MENU GENERATOR, REPORT GENERATOR, CODE GENERATOR ETC.

A. LAYOUT FORM & SCREEN GENERATOR:

- They are for printed report used to format the desired layouts.
- Example for such layout:

Customer Order Report
Date MM/DD/YY
Order Number 9999
Customer Name XXXXXXXXXXXXXXXXXXXXX

Catalog Number	Available	Location	Cost	Stock Level
XXXXXXXXXXXX	X	XXXXXX	999.99	99999
XXXXXXXXXXXX	X	XXXXXX	999.99	99999
XXXXXXXXXXXX	X	XXXXXX	999.99	99999
XXXXXXXXXXXX	X	XXXXXX	999.99	99999
XXXXXXXXXXXX	X	XXXXXX	999.99	99999
XXXXXXXXXXXX	X	XXXXXX	999.99	99999
	3 Exit	1.8 Column	8 Repeat	10 Field

B. MENU GENERATOR:

- It outlines the function which is to be accomplished by system.
- Menu may be linked to other submenus.

C. REPORT GENERATOR:

- It has the capacity to perform similar functions as in screen generator.
- It also indicates **totals, sequencing & control breaks etc.**

D. **CODE GENERATOR:**

- It allows the analyst to generate modular units of source code,
- From the high level specifications provided by system analyst.

SRS (SYSTEM REQUIREMENT SPECIFICATION)

1. It is prepared at the end of analysis phase.

2. SRS contains the following:

- ***Introduction***: *Goals/objectives of software.*
- ***Information Description***: *Problem, information content, flow structure, hardware, software etc.*
- ***Functional Description***: *Diagrammatic representation of functions, design constraint etc.*
- ***Behavioral Description***: *Response to external events & internal controls.*
- ***Validation Criteria***: *Classes of tests to be performed to validate functions, performance & constraints.*
- ***Appendix***: *Data flow/ object diagram; tabular data etc.*

ROLES INVOLVED IN THE SDLC

(i) STEERING COMMITTEE:

--Provides overall direction ensuring appropriate representation of affected parties.

-- Responsible for **all cost & timetables.**

-- Conducting review of progress of project.

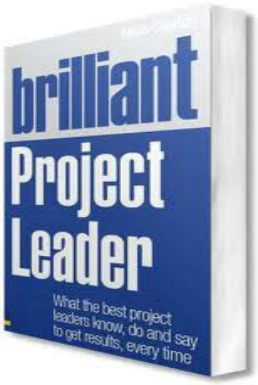
(ii) PROJECT MANAGER:

-- He is responsible for one or more projects,

-- Liaison with the clients.

-- Delivery of the projects within **time & budget.**

-- Review the progress of the project



-- **Project leader is dedicated to a project,**

-- **Ensuring its timely completion & objective fulfilment.**

-- **He reviews the project position more frequently.**

-- **Entire team reports to him.**

(iii) SYSTEM/BUSINESS ANALYST:

-- **System analyst responsibility is to conduct interview with the users & understand their requirement.**

-- **Link between UERS & PROGRAMMERS,**

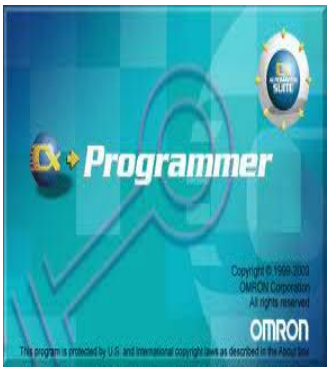
-- **Who converts user requirement in system requirement.**

(iv) MODULE/TEAM LEADER:

-- **Each project is divided into various modules & each module is under module leader.**

-- **Module leaders are responsible for delivery of tested modules within given time & money resources.**

(v) PROGRAMMER/ CODER/ DEVELOPER:



-- **Programmers are mason(builder) of software industry**

-- **who converts DESIGN into PROGRAMS by CODING**

-- **Using PROGRAMMING LANGAUGE.**

-- **He also TEST THE PROGRAM for DEBUGGING**

(vi) DATABASE ADMINISTRATOR(DBA):

-- Data in database has to be maintained by the specialist.

-- DBA handles multiple projects; ensure INTEGRITY & SECURITY of information stored in database.

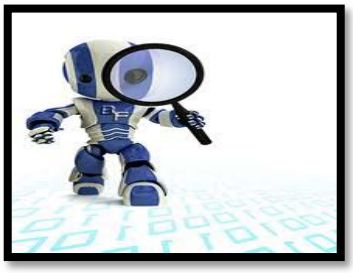
-- Helps the application development team in database performance issues.

(vii) QUALITY ASSURANCE:

-- This team sets STANDARD for development.

-- Checks COMPLIANCE with these standards by project team on PERIODIC BASIS.

(viii) TESTER:



-- Junior level quality assurance personnel

-- attached to a project who TEST PROGRAMS

-- as per plan given by the project leaders.

(ix) DOMAIN SPECIALIST:

-- Domain expert is needed where a project team has to develop an application in a new field.

-- Help of domain expert ensures that it is easy to interpret/anticipate user needs.

(x) IS AUDITOR:

-- **IS auditor ensures that application development also focuses on CONTROL PERSPECTIVE.**

-- **He should be involved in DESIGN & TESTING PHASE to ensure existence & operations of the controls in the new software.**

SYSTEM DESIGN

PHASE III

Next phase is the **SYSTEM DESIGN.**

Objective: It describes the part of the system & their interaction. It sets out **how the system shall be implemented using the** chosen **hardware, software** & network facilities, **specifies the program &** database specifications.

Activities: It includes the following activities:

-- **Describing inputs & outputs** -- **Determining processing steps**

-- **Computation rules for new solution** -- **Determining data files**

-- **Preparing program specifications** -- **Internal/external controls.**

Once the detailed design is completed, the design is distributed to the system developers for coding. The design phase includes following **STEPS:**

CODE TO REMEMBER: **A.D.D. - U.P.A.**

ADDRESS OF UPA GOVERNMENT

1. **A**rchitectural Design
2. **D**ata Design
3. **D**atabase Design
4. **U**ser-interface Design
5. **P**hysical Design
6. **A**cquisition & design of hardware/system software platform

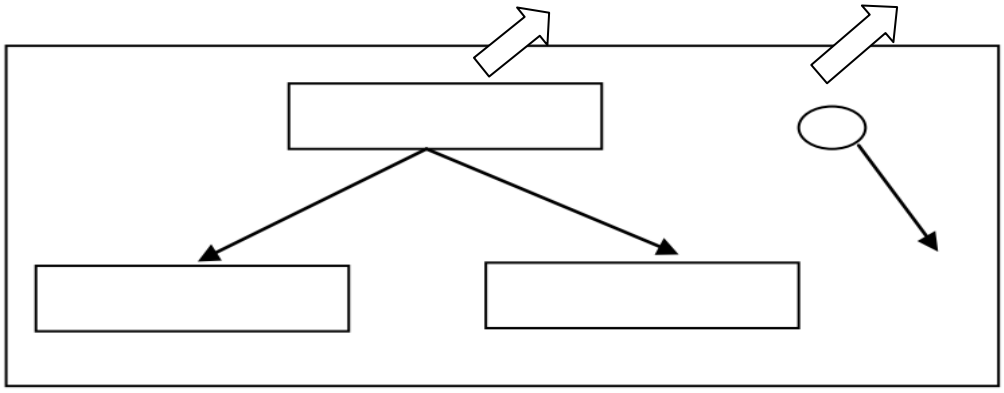
Now explaining each step in detail:

1. Architectural Design

- Deals with the organization of applications in terms of **hierarchy** of **modules & sub-modules.**

- Architectural design is made with the help of the tool called **FUNCTIONAL DECOMPOSITION.**

- It has 3 elements: **Modules** **Couple**



MODULE: Represented by a box & connection between them by arrows.

COUPLE: It is data element that moves from 1 module to another & is shown by an arrow with the circular tail.

2. Data Design

- It is important step in the design of the new system.
- Inputs required that are required:
  - **-- Existing Data/ information Flow**
  - **-- Problems with the present system**
  - **-- Objective of the new system**

3. Database Design

- It involves determining the **scope ranging LOCAL to GLOBAL structure.**

- **Where the interdependence is more, there is need of global database structure.**

- **Design of database involves 4 activities:**

Design Activity	Explanation
Conceptual Modeling	These describe the application domain via entities/objects, attributes of these entities/objects and static and dynamic constraints on these entities/objects, their attributes, and their relationships.
Data Modeling	Conceptual Models need to be translated into data models so that they can be accessed and manipulated by both high-level and low-level programming languages
Storage Structure Design	Decisions must be made on how to linearize and partition the data structure so that it can be stored on some device. For example-tuples (row) in a relational data model must be assigned to records, and relationships among records might be established via symbolic pointer addresses.
Physical Layout Design	Decisions must be made on how to distribute the storage structure across specific storage media and locations –for example, the cylinders, tacks, and sectors on a disk and the computers in a LAN or WAN.

4. User-interface Design:

- Involves determining the ways in which user interact with a system
- Following points to be considered:
  - **Source document to capture data** -- **Hard copy output report**
  - **Screen layout** -- **Inquiry screen for database interrogation**
  - **Graphic & color display** -- **Requirement for special devices**

5. Physical Design:

- In this phase, logical designs are turned into UNITS.
- Such UNITS can be decomposed into **implementation units** such as program and modules.

Design Principle:

A. **There is tendency to develop one design & consider it as a final product.**

B. **The design should be based on the analysis.**

C. **Software functions to be relevant to business.**

E. **The design should follow standards laid down.**

F. **The design should be** **Modular**

MODULE: It is a manageable unit **containing data & instructions** to perform a well defined task.

COHESION: Refers to manner in which element within a module are linked.

COUPLING: It is measure of interconnection between modules

GOOD MODULE: ↑COHESION ↓COUPLING

6. Acquisition of hardware/system software Design:

- In this phase, new system requires hardware & system software,
- Not currently available.
- New hardware/ system software platform required to support application.
- In case, new system & application are not able to communicate with each other, then **subsequent changes** have to be incorporated

SYSTEM DEVELOPMENT

PHASE IV

Development of the system relates first of all with the **acquisition of HRADWARE, SOFTWARE AND SERVICES.**

1. ACQUISTION STANDARDS: Following are the points to be considered:

1. Ensures **SECURITY, RELAIBILTY** & **FUNCTIONALITY** already built into a product.

2.Ensures manager complete appropriate vendor, contract & licensing reviews & acquiring **PRODUCTS COMPATIBLE** with existing system.

3. Including invitation to tenders & request for proposal. It involves soliciting **BIDS from VENDORS.**

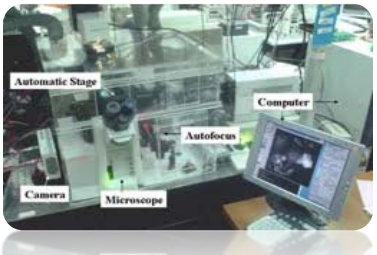
4.Ensuring functional, security & operational requirement to be **accurately identified &** clearly detailed in **REQUEST -FOR- PROPOSAL.**

2. ACQUISTION SYSTEMS COMPONENTS FROM VENDORS:

- ❖ At the end of design phase, organization gets an **idea** of types of **hardware, software & services** it needs for the system.

- ❖ Acquiring appropriate hardware/ software is critical matter.

A. **HARDWARE ACQUISTION:**



- ■ Management normally relies on **time tested** selection techniques.
- ■ & such selection criteria can **be delegated to TECHNICAL SECIALIST.**
- ■ Management relies upon vendor's services for every aspect of the system acquired.

B. **SOFTWARE ACQUISTION:**

- ■ Next step is determining the natures of application software requirement.

- ■ This helps the **development team** to decide what type of application software is needed.

- ■ System Developers must determine whether the application software should be created in house or to be purchased.

C. **CONTRACTS, SOFTWARE LICENSES & COPYRIGHT VIOLATIONS:**

Contracts between an organisation & software vendor should clearly describe the RIGHTS & RESPONSIILITIES of the parties to contract. The contract should be in WRITING

It is a license that grants permission to do things with the computer software. The aim is to ensure that only authorized activities are undertaken.

Copyright laws protect PROPERITARY as well as OPEN-SOURCE software. The use of UNLICENSED software may result in LEGAL LITIGATION against the company.

D. VALIDATION OF VENDOR'S PROPOSAL:

1. FACTORS FOR EVALUATING THE VENDOR'S PROPOSAL

- To acquire software/ hardware requires evaluating & ranking proposal s submitted by the **VENDORS.**

- Following points to be considered while doing evaluating:

CODE TO REMEMBER: V-M.B.-P.C.

WHAT IS VENDORS PC'S (PERSONAL COMPUTER) MB?

- ☐ **Vendor support.**
- ☐ **Maintenance cost of each proposal.**
- ☐ **Cost & Benefit analysis of each proposed.**
- ☐ **Performance capabilities of each proposed system.**
- ☐ **Compatibility of new system with existing one.**

2. FACTORS FOR EVALUATING THE VENDOR'S PROPOSAL

CODE TO REMEMBER: B.P. - C.P.T.

BP(BLOOD PRESSURE) WHILE IN CPT

1. **B**ench marking problem for the vendor proposal:

   -- **These are sample programs that represent at least a part of the buyer's primary computer work load.**

   -- **Benchmarking process is oriented towards TESTING whether system provided by vendor is as per requirement or not.**

2. **P**ublic evaluation report:

   -- **Several consultancy agencies compare & contrast the,**

   -- **Hardware & software performance for various manufacturer & publish their reports.**

   -- **This method is useful when the buying staff has inadequate knowledge of facts.**

3. **C**hecklists:

   -- **It is most simple & a subjective method of validation & evaluation.**

   -- **Various criteria is put up in the CHECKLIST in form of suitable questions against which vendors' proposal are evaluated.**

4. **P**oint-scoring analysis:

-- **It provides an OBJECTIVE means of selecting a final system.**

-- **It provides only guidelines for matching user needs with the software capabilities.**

POINT SCORING ANALYSIS LIST

Software Evaluation Criteria	Possible points	Vendor A	Vendor B	Vendor C
Does the software meet all mandatory specifications?	10	7	9	6
Will program modifications, if any, be minimal to meet company needs?	10	8	9	7
Does the software contain adequate controls?	10	9	9	8
Is the performance (speed, accuracy, reliability, etc.) adequate?	10	7	9	6
Are other users satisfied with the software?	8	6	7	5
Is the software user-friendly?	10	7	8	6
Can the software be demonstrated and test-driven?	9	8	8	7
Does the software have an adequate warranty?	8	6	7	6
Is the software flexible and easily maintained?	8	5	7	5
Is online inquiry of files and records possible?	10	8	9	7
Will the vendor keep the software up to date?	10	8	8	7
Totals	123	94	106	85

5. **T**est problems:

-- **It is developed to evaluate TIME required to translate the,**

Source code [program in high level language]

INTO

Object code [Machine language]

-- **The result achieved by the machine can be compared & PRICE PERFORMANCE judgment can be made.**

DEVELOPMENT OF THE SYSTEM : *Programming Techniques & languages*

(I) Objective: To convert the specifications into a functioning system.

(II) Activities: Application programs are written, tested & documented & system testing.

(III) Good coded program should have following characteristics:

CODE TO REMEMBER: U - R - R.A.R.E.

- **U**sability:
  - ✓ **Refers to user-friendly interface & easy to understand,**
  - ✓ **Document required for any program.**

- **R**eadability:
  - ✓ **It refers to the EASE of MAINTENANCE of program,**
  - ✓ **Even in the absence of the program developer.**

- **R**eliability:
  - ✓ **Refers to the CONSISTENCIES which a program,**
  - ✓ **Provides over a period of time.**

- **A**ccuracy:
  - ✓ **It refers to which program has to be done & which has not to be done.**

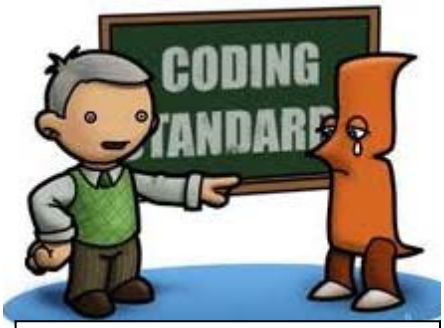
❖ Robustness:

- ✓ **It refers to PROCESS of taking into account,**
- ✓ **All possible inputs & output of a program.**

❖ Efficiency:

- ✓ **It refers to performance which should not be,**
- ✓ **Unduly affected with the increase in INPUT VALUES.**

(IV) PHASES IN THE DEVELOPMENT STAGE:



Make the Programming Code

A. Standards of the programming code→

- Here the logics of the flowcharts are converted into the program statement at this stage.
- Different programmers use different programs but **yielding same results.**
- It is imperative to have a **coding standard** required to minimize setbacks due to programmer turnover

Program changed in computer language

B. Programming Language→

- Programs that are made in 1st step is now, converted into machine language
- The same is converted by the **COMPLIER** to binary machine for the computer.
- **Programming languages that are used:**
  - **a.** COBOL & C language
  - **b.** Object oriented languages i.e. **C++, JAVA**
  - **c.** Scripting language like **JavaScript, VBScript**
  - **d.** Decision support language like **PROLOG.**

C. Program Debugging→

- Debugging refers to **correcting programming language SYNTAX AND DIAGONISTIC ERRORS.**

That program is now made ERROR free by **debugging**

- Debugging consists of following steps:
  - **a.** Inputting source program to compiler
  - **b.** Let the compiler find error in program
  - **c.** Correcting lines of code containing errors.
  - **d.** Resubmitting the correct source program.

D. Test Program →

- After resubmitting the correct program, next step is
- **CAREFUL & THOROUGH** testing of each program.
- Test plan requires **execution of all standards processing logics.**

After debugging I test the program

E. Program Documentation →

- Documentation is done of the **procedures & instructions** for people who will use the software throughout **program life cycle.**
- **Users** must carefully **review the documents.**
- **Documentation** to be done for better **understandability.**

F. Program Maintenance →

- Requirement of business data processing,
- Application are subject to **continual changes**
- **MAINTENANCE PROGRAMMER** are entrusted with the task of maintaining it.

SYSTEM TESTING

PHASE V

Objective: Testing is a process used to identify **the correctness, completeness and quality** of developed computer software. Data collected through **testing** can also provide an indication of the **software's reliability and quality.**

DIFFERENT LEVELS OF TESTING ARE AS FOLLOW:

A. UNIT TESTING:

INTRODUCTION:

- It is a software verification & validation method in which
- a programmer test of individual source code.
- **UNIT TESTING** = It is smallest testable part of an application which may be an individual program.
- Unit tests **are typically written & run** by software developers to ensure that code meets its design as needed.
- 5 categories of **TEST** that a programmer performs:

CODE TO REMEMBER: **F- S^2P^2**

A. **F**unctional tests:

- It checks **"whether program do what they are supposed to do"**
- **TEST plan specifies the INPUT values, expected results &** according to this **programmer checks** the inputs & actual result.

B. **S**tructural tests:

- These tests are concerned with **examining the INTERNAL PROCESSING** logic of the software system.

C. **S**tress Test:

- It is form of testing that is used **to determine the STABILITY** of a given system.
- It involves **TESTING BEYOND NORMAL OPERATIONAL CAPACITY.**

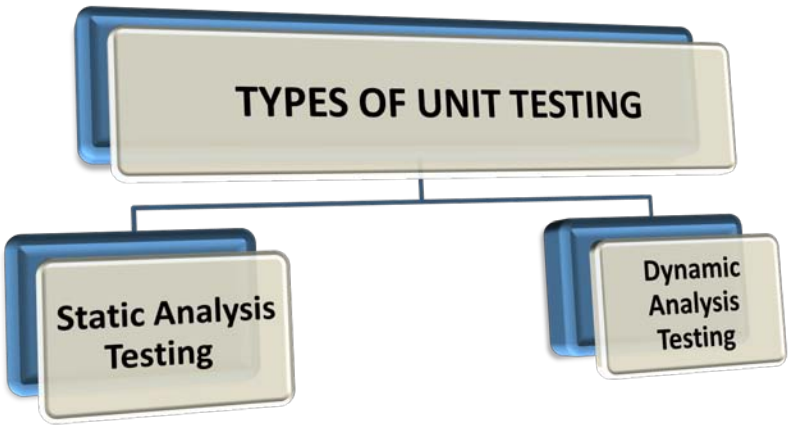
D. **P**arallel tests:

- Here the **SAME DATA** is used in the **NEW & OLD** system & the output results from both the system are compared.
- **These tests are designed to OVERLOAD** a program in various ways.
- Purpose of the stress test: **to determine limitations of program.**

E. **P**erformance tests:

- It should be designed to **verify the response time, execution time, the throughput, memory utilization etc.**

TYPES OF UNIT TESTING:



SOME IMPORTANT STATIC ANALYSIS TESTING:

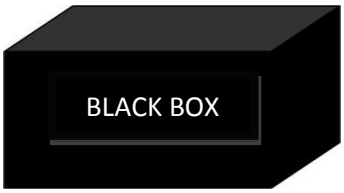
CODE TO REMEMBER: **C.S.D..**

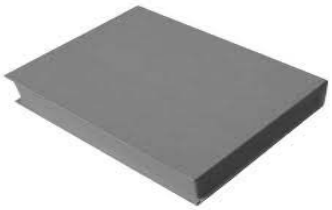
1. Code inspection:
   - **Program is reviewed by the FORMAL COMMIITTEE.**
   - **Review is done with FORMAL CHECKLIST.**

2. Structured Walk-through:
   - **Application developers lead other PROGRAMMERS through the TEXT of the program.**

3. Desk Check:
   - **This is done by the PROGRAMMER himself.**
   - **He checks the logical SYNTAX errors, & deviation from code standards.**

DYNAMIC ANALYSIS TESTING:





WHITE BOX TESTING

BLACK BOX TESTING

GRAY BOX TESTING

I am hereby explaining each of the testing type in detail:

WHITE BOX TESTING

- It uses the **internal perspective of the system** to design **TEST CASES** based on internal structure.
- It requires **programming skills** to identify all paths through the software. The tester chooses:

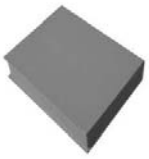
  Test case inputs ⟶ Exercise paths through code ⟶ App. Output

- Since test is based on **actual implementation,** if the implementation changes then test will need change too. **It can also TEST PATHS between UNITS during INTEGRATION & between SUBSYSTEMS.**
- After getting a clear picture of the **internal working** of a product, **test can be conducted** to ensure that internal operations of the product are as per **specification.**

BLACK BOX TESTING

- It uses the **external perspective of the TEST OBJECT** to **derive** test **CASES.**
- The test designer **selects VALID & INVALID** inputs & determines the correct output. **There is NO KNOWLEDGE of the TEST OBJECT'S internal structure.**

- This method of testing is applicable to **ALL LEVELS OF SOFTWARE TESTING** i.e. **unit, integration, functional**



BLACK BOX TESTING

It is the technique that uses both **black & white testing.**

GRAY TESTING = BLACK BOX TESTING + WHITE BOX TESTING

Here the tester applies a **limited number of test cases to INTERNAL WORKINGS OF THE SOFTWARE** (White Box).

In remaining part of the gray box testing, **one takes a BLACK BOX approach in applying INPUTS to the software under test & observing the OUTPUT.**

B. INTEGRATION TESTING:

INTRODUCTION:

- It is an **activity** in which **individual software modules** are combined and tested as a group.
- It occurs **after UNIT TESTING & before SYSTEM TESTING,**
- With an objective **to evaluate the CONNECTION** of 2 or more components that pass information.

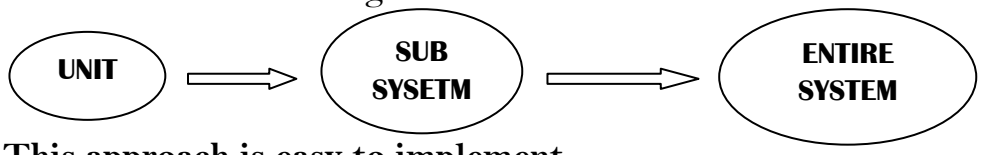
Takes as its **input-MODULES that have been unit tested,** ⇒	Groups **them in larger aggregates, applies tests defined in a plan,** ⇒	To those integrates **& delivers as its OUTPUT.**

❖ **INTEGRATION testing can be carried in following ways:**

1· BOTTOM-UP INTEGRATION:

- **Traditional strategy used to INTEGRATE the components of a software system into a FUNCTIONING WHOLE.**

- **Manner** in which testing to be done:



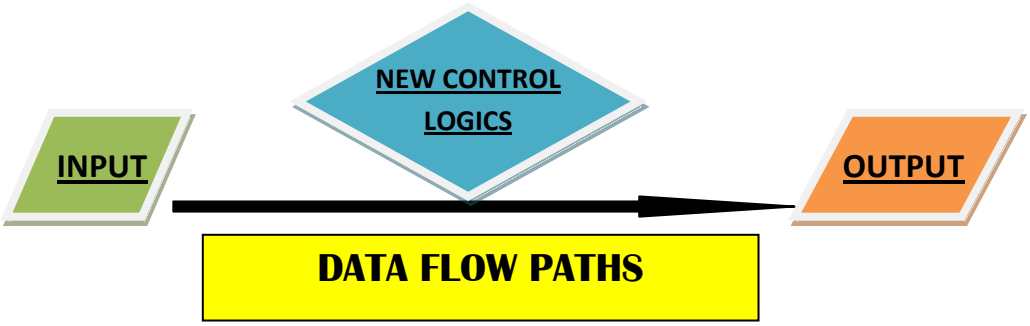
- **This approach is easy to implement.**

2· TOP-DOWN INTEGRATION:

- **In this an INCOMPLETE portion of program code that is put under a function,**
- **In order to allow the FUNCTION & the program to be COMPILED** & **TESTED.**
- **In the initial stage,** STUBS **are substituted, but after the testing STUBS are substituted with the REAL MODULES.**
- **This approach emphasis on MAJOR CONTROL DECISION POINTS** encountered in early stage of the process.

3· REGRESSION TESTING:

- Each time a **new module** is added as part of INTEGRATION TESTING, **the SOFTWARE CHANGES.**



- This may cause **problems** in the functioning of the **PREVIOUS SYSTEM** that run **FLAWLESSLY.**

- **REGRESSION TESTING** ensures that **CHANGES OR CORRECTIONS** have not introduced **NEW ERRORS.**

Types of SYSTEM TESTING:

RECOVERY TESTING

This testing involves **"How well the application is able to recover from CRASHES, HARDWARE FAILURES & other problems."**

Recovery testing is the **forced failure of the software** in a variety of ways to verify that **recovery is properly performed.**

SECURITY TESTING

This is the process to determine **that an INFORMATION SYSTEM** protects the DATA and maintains functionality as required or not.

6 basic security testing are: **Confidentiality, Integrity, Authentication, Authorization, Availability & Non-Repudiation.**

PERFORMANCE TESTING

This testing is used to **determine THE SPEED/EFFECTIVENESS OF A** computer, network, software programs and devices.

This technique **compares the NEW SYSTEM'S PERFORMANCE** with that of similar system with well defined benchmarks.

STRESS TESTING

This testing is used to **determine the STABILITY of a given system or entity. It involves** test beyond the **NORMAL OPERATIONAL CAPACITY.**

Stress test is performed by testing the **application with LARGE QUANTITY** of data during the **PEAK HOURS** to test its performance.

FINAL ACCEPTANCE TESTING:

When the system is just ready for **implementation, FINAL ACCEPTANCE TESTING** is done. Here it is **ensured that NEW SYSTEM satisfies the QUALITY STANDARDS** adopted by the business & system satisfies the users. **2 major parts of FINAL ACCEPTANCE:**

1. **Quality Assurance Testing:**

   Ensures that new system satisfy the prescribed quality standard.

2. **User Acceptance Testing:**

   Ensures that functional aspects expected by the users have been well addressed in the NEW SYSTEM.

System Implementation

Phase VI

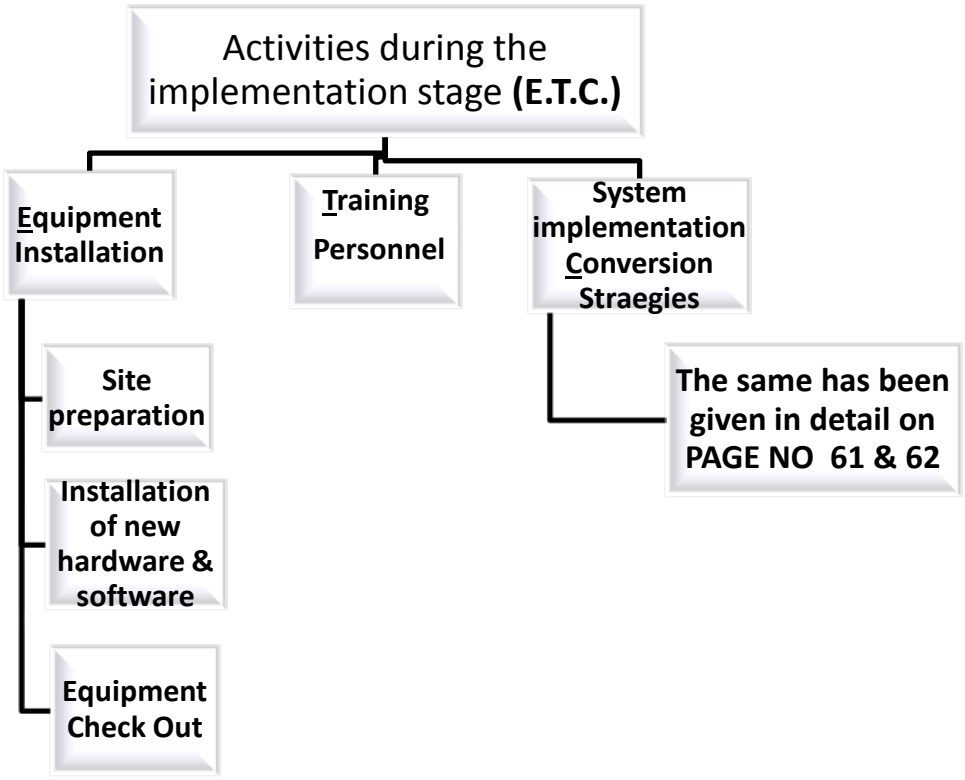
Objective: To implement the new system.

Activities: Following are the activities undertaken:

Code to remember : D.C. - S.E.T.

- **D**ocumentation(User) Completion
- **C**onversion of data into new system files
- **S**ystem changeover
- **E**valuation of system on a regular basis
- **T**raining of end users

Activities involved in a system implementation stage:



Equipment installation:

Hardware that is required to support new system is selected **PROIR TO THE IMPLEMENTATION PHASE.** An **installation checklist** is to be prepared with the advice from the **VENDORS & SYSTEM DEVELLPMENT TEAM.** Following activities are being performed:

- ✓ SITE PREPARARTION:
  - -- Appropriate location must be found to provide an operating environment for the equipment that will meet **the vendor's TEMPERATURE, HUMIDITY & DUST CONTROL SPECIFICATIONS.**

- ✓ INSTALLATION OF NEW HARDWARE:
  - -- Installation to be done by the **MANUFACTURER.**
  - -- If the installation needs **interface of the new system** with the other system, **then some TESTS of the production environment is to be performed.**

- ✓ EQUIPMENT CHECK OUT:
  - -- Equipment must be turned on for TESTING under normal operating conditions.

Training Personnel:

-- A system can **succeed or fail** depending on the way it is operated and used.

-- As such the **QUALITY OF TRAINING** has serious impact on the system. Thus **TRAINING IS MAJOR COMPONENT** of system implementation.

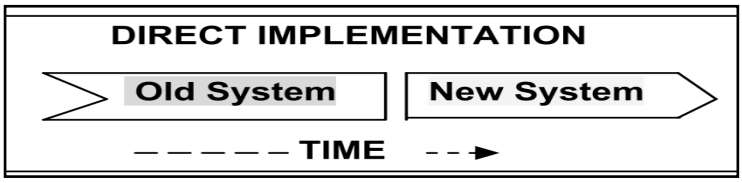
-- Such training can be imparted through **CLASSES, HANDS-ON LEARNING TECHNIQUES.**

System Implementation **C**onversion Strategies:

-- It is the process of changing from **OLD SYSTEM (manual system) to the NEW SYSTEM.**

THERE ARE 4 TYPES OF IMPLEMENTATION STRATEGIES→

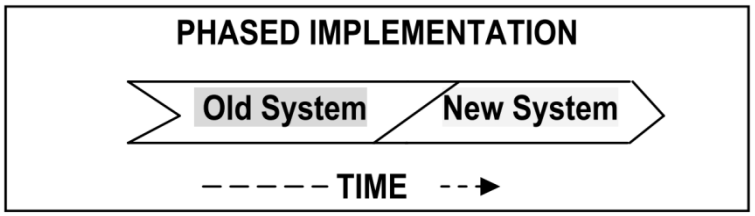
I. DIRECT IMPLEMENTATION/ABRUPT CHANGE OVER:



Under this strategy, **the changeover is done in ONE OPERATION** completely replacing the OLD SYSTEM.

This conversion takes place often after a break **IN PRODUCTION.**

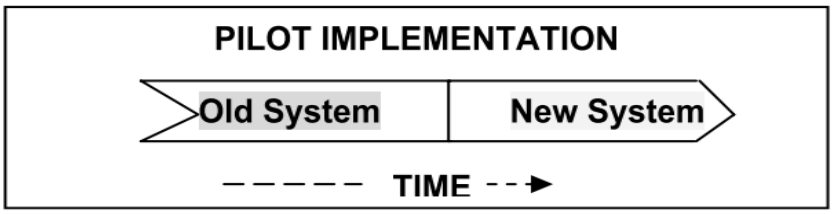
II. PHASED IMPLEMENTATION:



Under this strategy, **the changeover is done** by **DEGREES.**

Some new files may be converted & used by the employees while other files continue to be used in **OLD SYSTEM.** If that phase is successful, then next phase is started.

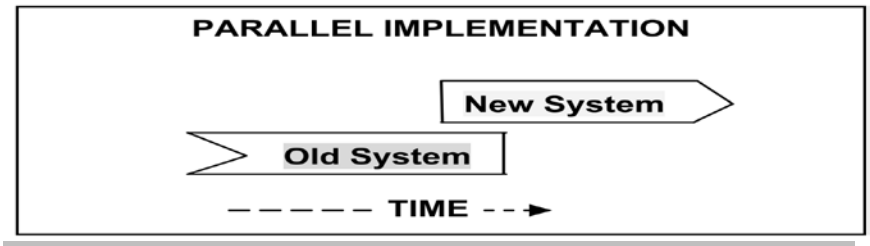
III. Pilot implementation:



Under this strategy, **new system replaces the OLD system in ONE operation** **but on a SMALL SCALE.**

If successful then the **pilot is extended until it eventually replaces the OLD SYSTEM completely.**

IV. Parallel Running implementation:



Under this strategy, **old system remains fully OPERATIONAL** while the new system comes **ONLINE.**

Both the system operates **independently.** If all goes well then **the old system is STOPPED** & new system carries on **as the ONLY SYSTEM.**

ACTIVITIES INVOLVED IN THE CONVERSION→

1. PROCEDURE CONVERSION:

- Operating procedure to be documented completely for the new system.
- Before any conversion activities take place, operating procedures must be clearly spelled out.
- Written operating procedures must be supplemented by oral communication during the training session.
- Information on inputs, data files, methods, procedure etc must be **presented in a way** that is understood by the reader easily.

2. FILE CONVERSION:

- Here the large information files are to be converted from one medium to another.
- This phase has to start before **programming & testing phase.**
- The major consideration is the cost involved in the file conversion. i.e. **OFF-LINE OR ON-LINE** conversion
- The existing computer files should be **kept** for a period of time until **sufficient files are accumulated for the back-up.**

3. SYSTEM CONVERSION:

- When the **files are converted & reliability of the new system is confirmed,**
- Daily processing can be **shifted from existing to NEW system.**
- All the transaction after this phase is **carried on the new system. Proper development team must be there to assist the QUERIES** that might develop.

4. SCHEDULING PERSONNEL & EQUIPMENT:

- Scheduling data processing operations of a new system is a difficult task.
- When the users are familiar with the new system, it becomes the routine job.

POST IMPLEMENTATION REVIEW & SYSTEM MAINTENANCE

PHASE VII

Objective: To assess and review the complete working solution.

Activities: Some of the system maintenance activities are:

- **Adding new data elements**
- **Modifying reports**
- **Adding new reports**
- **Changing calculations**

POST IMPLEMENTATION REVIEW

Post implementation review **ascertains the degree of success from the project.**

Post implementation review should be scheduled some time after the solution has been implemented. There are 2 dimensions of this:

-- **Whether the newly developed system is operating properly**

-- **Whether the uses are satisfied with the information supplied by it is as per requirement or not.**

Following types of evaluation takes place:

CODE TO REMEMBER: I – D.O.

A. INFORMATION EVALUATION:

- It must be evaluated in terms of information it provides.
- The objective of the information system is to **provide information for the decision making.**

B. DEVELOPMENT EVALUATION:

- It is primarily concerned with the **whether the system was developed on SCHEDULE & within BUDGET.**
- It requires schedule & budgets to be established in advance and that records of **actual performance & cost can be maintained.**

C. OPERATIONAL EVALUATION:

- This evaluation helps in determining whether the **HARDWARE, SOFTWARE** and **PERSONNEL** are capable of performing their duties.

SYSTEM MAINTENANCE:

It is an important aspect of the **SDL.** Most of the information system requires some sorts of modifications after development. Various types of modifications:

CODE TO REMEMBER: P-S.C.R.A.P.

A. Preventive maintenance:

- **Aimed at increasing system's maintainability.**
- **I.e.** updating documentation, adding comments etc.
- **As large program is continuously changed,** its complexity increases unless **work is done to maintain it**

B. Scheduled Maintenance:

- **It is anticipated & can be planned for.**
- For example: **Coding scheme for inventory planned in advance.**

C. Corrective Maintenance:

- **It deals with FIXING BUGS in the code.**

- **Defect can be due to:** Design, logic, coding, data processing, errors etc.

D. Rescue Maintenance:

- **Refers to previous undetected malfunctions that require immediate solutions.**

E. Adaptive Maintenance:

- **Consists of adapting software to changes in environment.**

- **ENVIRONMENT:** Totality of all conditions which act from outside upon the system. **For example:** business rules, political conditions, government policies etc.

F. Perfective Maintenance:

- Deals with the **accommodating to NEW/CHANGED user requirement.**

- - - - - I AM HEREBY ENDING THE SECOND CHAPTER. REMAINING TOPIC WILL I UPLOAD ON INTERNET- - - - - -