

Software Engineering

- Program • Software
- Soft. Eng. • System
- character of Software
- High level Language
- Control Flow • flowchart
- Data Flow Oriented (DFO)
- OOPS

- SDLC - Adv • SRS
- 7 Phase SDLC
- classic waterfall Model
- Iterative waterfall
- Prototyping Model
- Evolution Model
- Spiral Model
- Agile Methodology

- Project Planning
- Project character
- Project Management
- Project Manager

CHARACTERISTICS OF SOFTWARE

- 1 Correctness
- 2 Understandability
- 3 Efficiency
- 4 Maintainability

Coupling

COHESION

- ① Coupling is a concept of Inter Module
- ② Coupling is a measure that defines inter-dependability among modules
- ③ Coupling Represents the Relationship between Modules
- ④ HIGH Coupling is not Good for Software
- ⑤ Coupling Represents independency of Module Among
- ⑥ Coupling, Modules are Connected to the other Modules
- ⑦ Type
 1. Content Coupling
 2. Common Coupling
 3. Control Coupling
 4. Stamp Coupling
 5. Data Coupling

Ch 4

Cohesion

COUPLING

Cohesion is a Concept of Intra Module

Cohesion is a measure that defines degree of Intra-dependability within elements of Module

Cohesion Represents the Relationship Within Modules

HIGH Cohesion is very Good to Design Good Software

Cohesion Represent the functional strength of Module

In Cohesion, Modules focuses on the SINGLE THING

Type

1. Co-Incident Cohesion
2. Logical Cohesion
3. Emporal Cohesion
4. Procedural Cohesion
5. Communicational Cohesion
6. Sequential Cohesion
7. Functional Cohesion

PROGRAM

1. A Program is developed by Individual for Personal
2. Small in Size
3. Single User
4. Lack of Proper Documentation
5. Lack of USER INTERFACE [UI]
6. HAVE Limited functionality
7. \$ PROGRAM Cannot be classified further
8. Programs are ~~design~~^{dev} To perform Specific Task Only

SOFTWARE

A Software Product is developed by Team of Engineers.

Large in Size

Large Number of User

Good Document Support

Good USER Interface [UI]

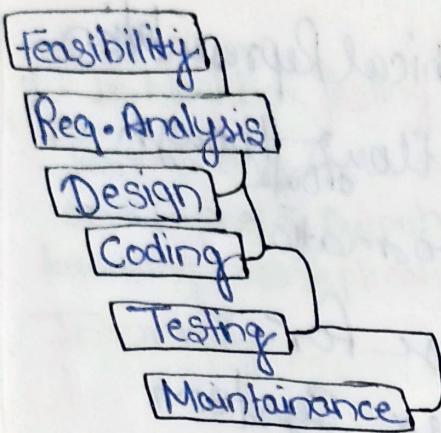
Have More functionality

Software Can be classified in two Categories

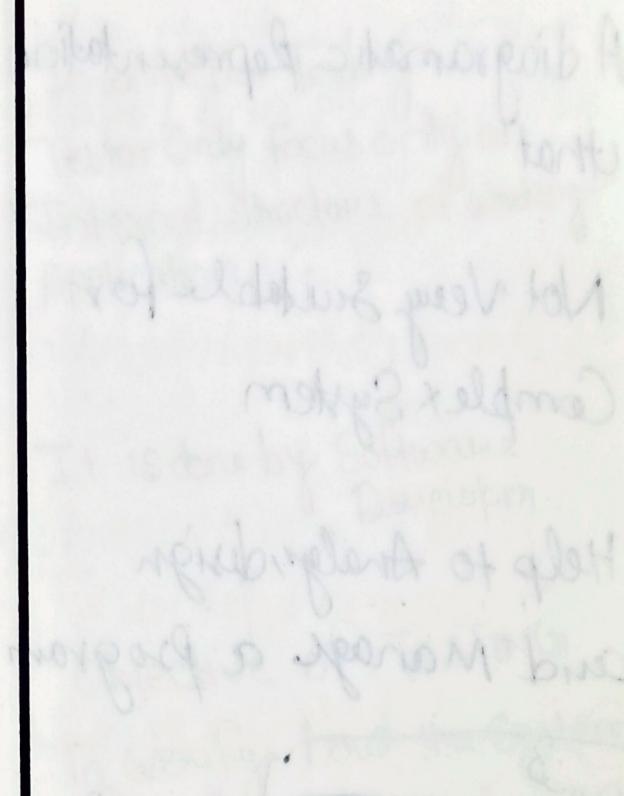
① Application ② System

Software is a Set of Codes & Program To Perform Specific functionality

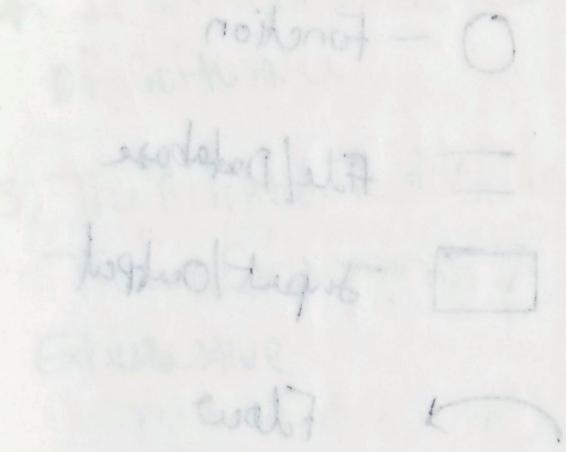
WATERFALL



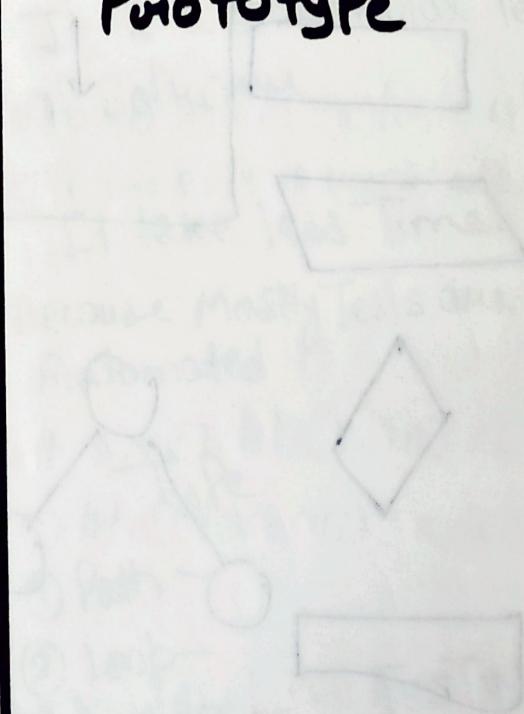
Iterative



SPIRAL



Prototype

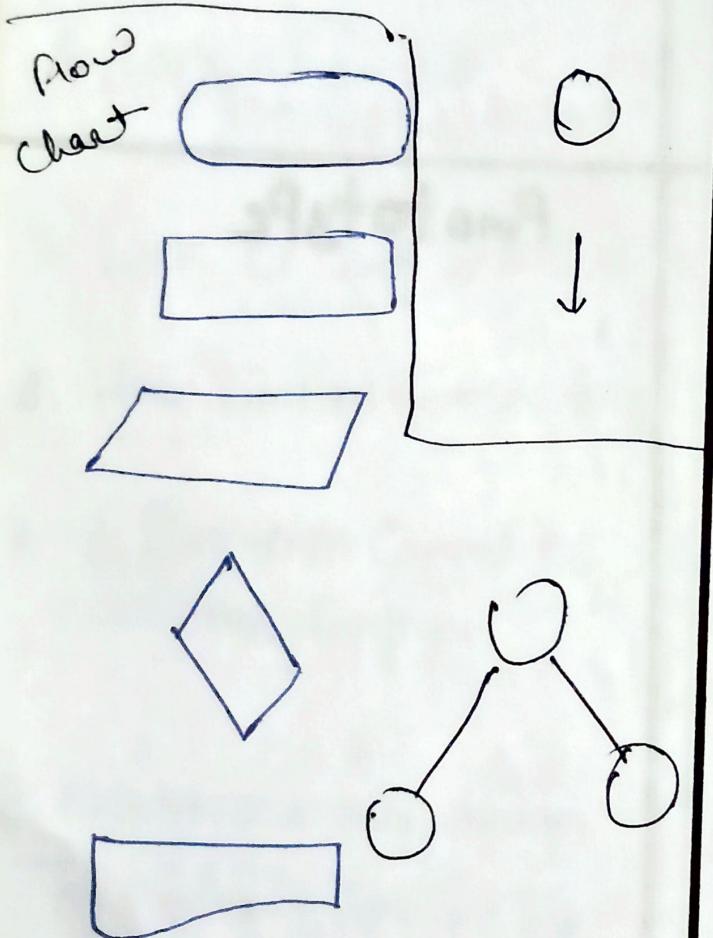


CONTROL FLOW CHART

A diagrammatic Representation
that

Not Very Suitable for
Complex System

Help to Analyse, design
and Manage a program



DFD

A Graphical Representation
of the Flow ^{of Data} through
an information

Can use for
Complex System

Help to understand
the Overview of the
System without
showing More
details

- — Function
- File / Database
- [] — Input / Output
- Flow

BLACK BOX TESTING

1. In blackbox Testing

Testers only focus only Input and Expected Output without knowing how Application work Internally.

2. It is Mainly done by Software Tester

3. Objective of TEST

To Verify what functionality System has to perform

4. It is a Functional test of Software

5. The Blackbox Testing is Time-Consuming & Exhaustive

6. Type

- ① Functional
- ② Non Functional → Testing
- ③ Regression.

7 Not Necessary to have knowledge

8. HIGH Level Testing

9. (SRS) Based Testing

WHITE BOX TESTING

In white box Testing

Tester Only focus only on Internal Structure of Working Application

It is done by Software Developer

Objective of TESTING

To verify how the System is performing

It is a Structural Test of Software

It take less Time

Because Mostly Tests are Automated

Type

① Path -

② Loop -

③ Condition -

④ Unit -

⑤ Integration -

Need to have Knowledge

Low level Testing

Detailed design Based T

co.com Model

V MODEL

SRS

characteristic of Good SRS

1. CORRECTNESS
2. COMPLETENESS
3. TESTIBILITY
4. DOCUMENTATION
5. MODIFICATION
6. DESIGN INDEPENDENCY

DÉSICCIÓN TABIE / TREGG

OOPS

COMPONENTS :-

Object :-

CLASS :-

Abstraction

Encapsulation

~~Encapsulation~~ Inheritance

PolyMorphism

ABSTRACTION

- ① Solve problem in design Level
- ② Abstraction hides unwanted data

ENCAPSULATION

- Solve Problem in Implement Level
- Wrap data in Single Unit

INHERITANCE

- ① Mechanism which allows a new class to use properties & method of a Superclass
- ② Implementation occurs in class level
- ③ Provide Code Reusability

POLYMORPHISM

Ability of an object to have or behave in multiple way or in many forms

It Occurs in Method

Allow Calling Methods at time & Runtime

SDLC STAGES

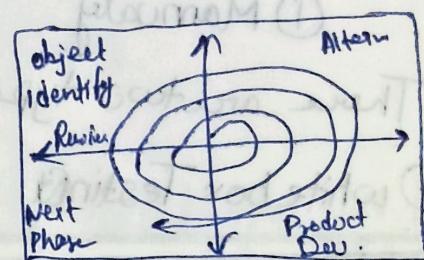
There are 7 phase of SDLC

- ① Planning
- ② Requirement
- ③ Design & prototype
- ④ Software development
- ⑤ Testing
- ⑥ Deployment
- ⑦ Maintenance

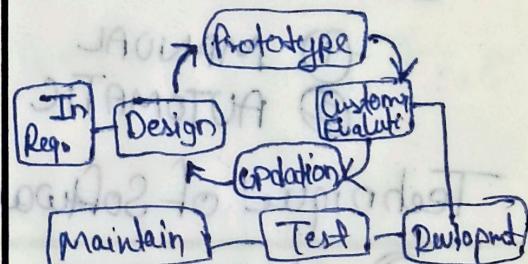
WATERFALL

- 1 Feasibility
- 2 Req. Analysis
- 3 Design
- 4 Coding
- 5 Testing
- 6 Maintenance

SPIRAL



PROTOTYPE



EVOLUTIONARY

SOFTWARE TESTING

Software Testing consist of :-

- check whether Actual Result matches to Expected Result
- Testing helps To Identify ERROR
- MISSING Requirement check
- Finding BUGS
- Software Can be Tested
 - ① Manually
 - ② Automatically

There are two type of Testing

- ① white box Testing
- ② black box Testing

Type of Software Testing

- ① MANUAL
- ② AUTOMATIC

Technique of Software Testing

- ① Black box
- ② white box

Level OF Testing

① Unit TESTING - Type Manual - Automatic

② Integration Testing

Type

① Topdown

② Bottom up

③ Sandwich
Big bang

③ System Test

④ Acceptance Test

⑤ Smoke Testing

• Manual • Automatic

⑥ Alpha Testing

⑦ Beta Testing

Software Project Management

Activity

- 1 Conflict Management
- 2 Risk Management
- 3 Requirement Management
- 4 Change Management
- 5 Customer Satisfaction

Software Project Manager

Role of Project MAN.

1. Planning
2. Leading
3. Execution
4. TIME MANAGEMENT
5. BUDGET
6. MAINTENANCE

PROJECT

Planning :-

- ① Project Size
- ② Project Cost
- ③ Duration
- ④ Effort

Characterstic

1. Project has a Specific Goal
2. has Defined date of Start & End DATE
3. Resources Required for Project planning

Software Engineering

Yashwan

SE is the branch of Engineering under which we study about Software which runs on Computer or other ~~com~~ electronic gadgets.

Hardware

Hardware are physical devices used to collect data to run programs or other physical activities like data input or display

Software

Software Collection of Data Command or Programs used to perform specific task

Software Component: —

- ① Analysis —
- ② Requirement —
- ③ Coding —
- ④ Designing —
- ⑤ Testing —
- ⑥ Updation —
- ⑦ Development —
- ⑧ Maintenance —

need

@

objective

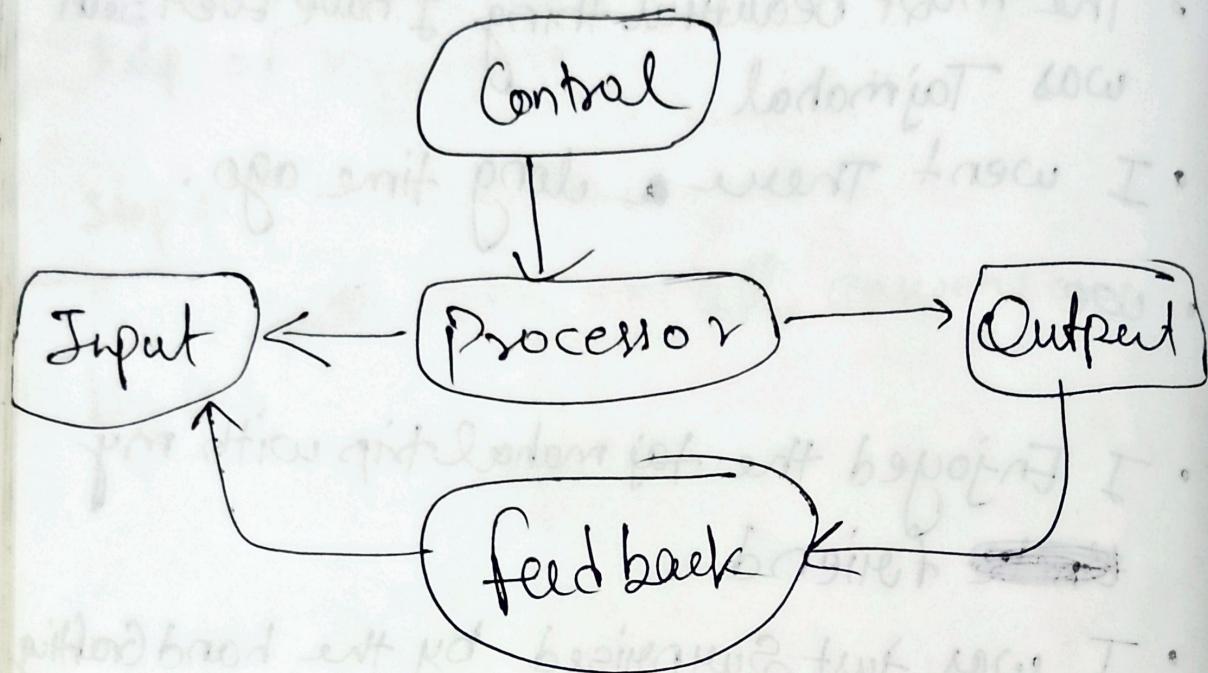
→ DIF

- ① maintainability
- ② Correctness (SRS) Software Requirement Specification
- ③ ~~Re~~usability
- ④ Testability
- ⑤ Reliability
- ⑥ portability (platform Independence)
- ⑦ adaptability (change in system)

Necessity

- ① To maintain software in Large Scale
- ② → manage enterprises,
SAP →
AWS- Amazon Server
- ③ Cost Effectiveness
- ④ To manage ~~Dynamic~~ software Dynamic Nature
- ⑤

Elements of system



System types

- ① open & closed Ex. Android / Mac
- ② static & dynamic

open & closed

- ① An open system interacts with its environment. It receives input from the outside & ~~delivers output~~ delivers output outside.

close System

A close system not interacts with environment. It is not share information or data with its environment.

* * static * *

static system are those system whose output depends upon only present value of input

Dynamic S

It means Capable of changing then output may be variable output depends upon present as well as past value of input

Program

Program is a set of instruction or commands that a Computer follow in order to perform a specific task or function

Software Eng.

- ① clear understanding
- ② Team Coordination (clear guideline) hierarchically pos.
- ③ standup form, structural principle follows



Feasibility mean ^{significance} chance to -
Probability

} financiality ↗ matches to the user budget
(client)

Technical

It is possible to make or not

2 Requirement Analysis & Specification

- Collect info for making Software
- understand & Documentation of user requirement
- SRS - purpose of making ~~features~~ features, abstraction, Reference
Overview & Overall Description about Hardw/Soft. GUI's etc.
- function & characteristics, availability, portability, response
- Hierarchy
- Doc. Approval

3. Design
→ flowchart, DFD, control flow, object oriented

4. Coding & Testing - writing code & testing otherwise
→ ~~Combining~~ Combining coded programs

5. Integration → ~~Combining~~ Combining coded programs

⑦ Deployment

Deployment issue by project manager to check again

⑧ Maintenance

- Bug fixing
- Updates
- Enhancement (Modification in initial software)

Prototyping can be further divided into below:

① Rapid -

② Evolutionary

In Rapid prototyping, a model of final product is created with the help of a group of techniques they is normally used in System where 3D Models of a system is required this type of prototype Normally cannot used in software development

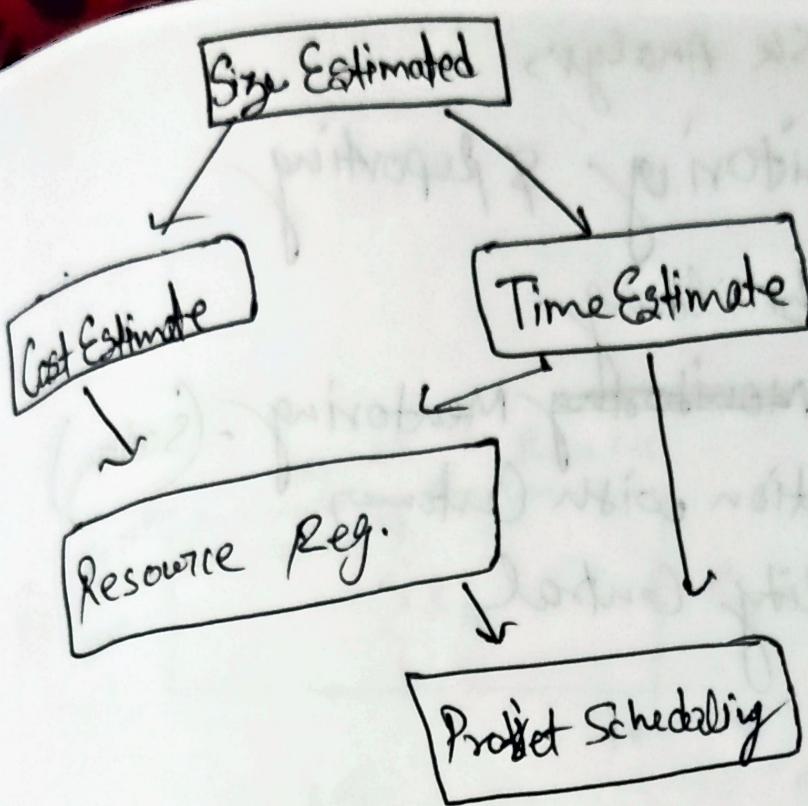
Evolutionary prototyping - in this technique

Developer first construct a prototype after Receiving feedback from Customer Multiple prototypes are constructed Each with Additional ~~equally~~ functionality or improvement until final product evolves from these prototypes

when a project is ~~not~~ feasible and under taken to process, software projects Many have to estimate or plan all activities before start this process is called software project planning or simply software planning

Some of the project planning Activities are :-

- ① Size Estimation project - Project size means how much time & effort will be approximately required to complete the project
- ② Cost Estimation project :-
To estimate how much cost will be inc. in completing the project
- ③ Duration - How does it will take to complete the project Only in terms of time



The responsibility of a project manager includes many activities like

- ① planning & defining scope of project
- ② Activity planning
- ③ Resource planning
- ④ Budgeting ~~planning~~
- ⑤ Staffing
- ⑥ Time Estimation
- ⑦ Documentation