MODULE - 1

Date: 16/07/2025

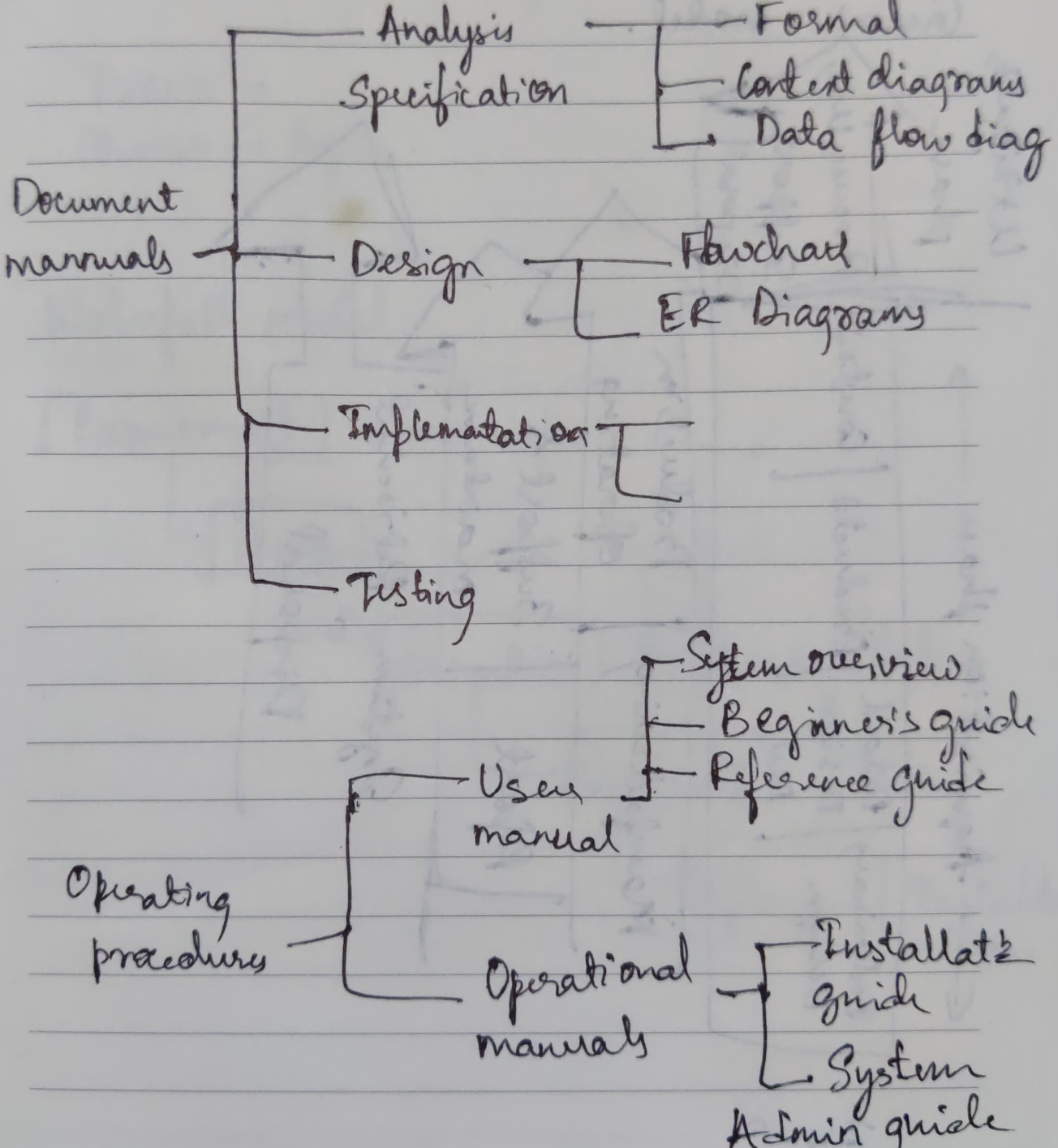
œrlikon balzers
œrlikon metco

SDLC

Proper Software : Create maintain necessary document to make it viable.

To Learn

- Bit manipulation
- Overflow, Underflow of bits



Date / /

Software Process → Differs wrt diff contexts

Developers can systematically used resources and give o/p

SDLC -

Generic model.

Date: / /

oerlikon balzers oerlikon metco

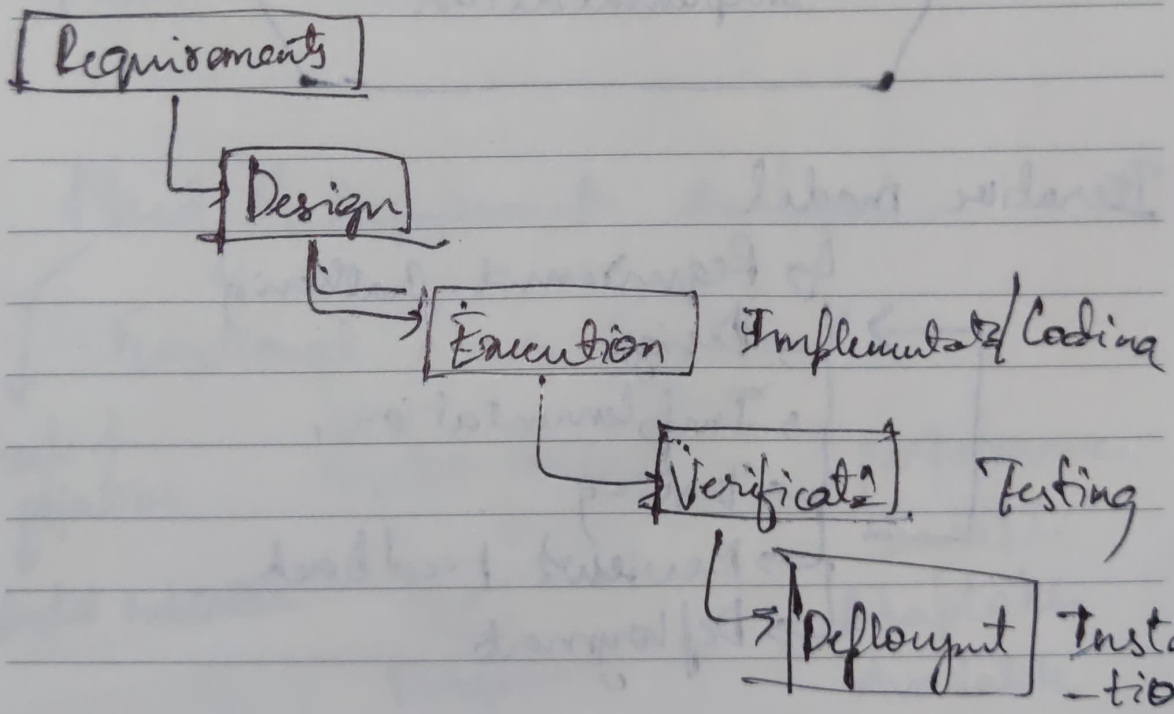
① System Complexity.

Drivers →
Choices → V-model / Spiral model

② Risks & Uncertainity

Drivers →
Choices → Agile.

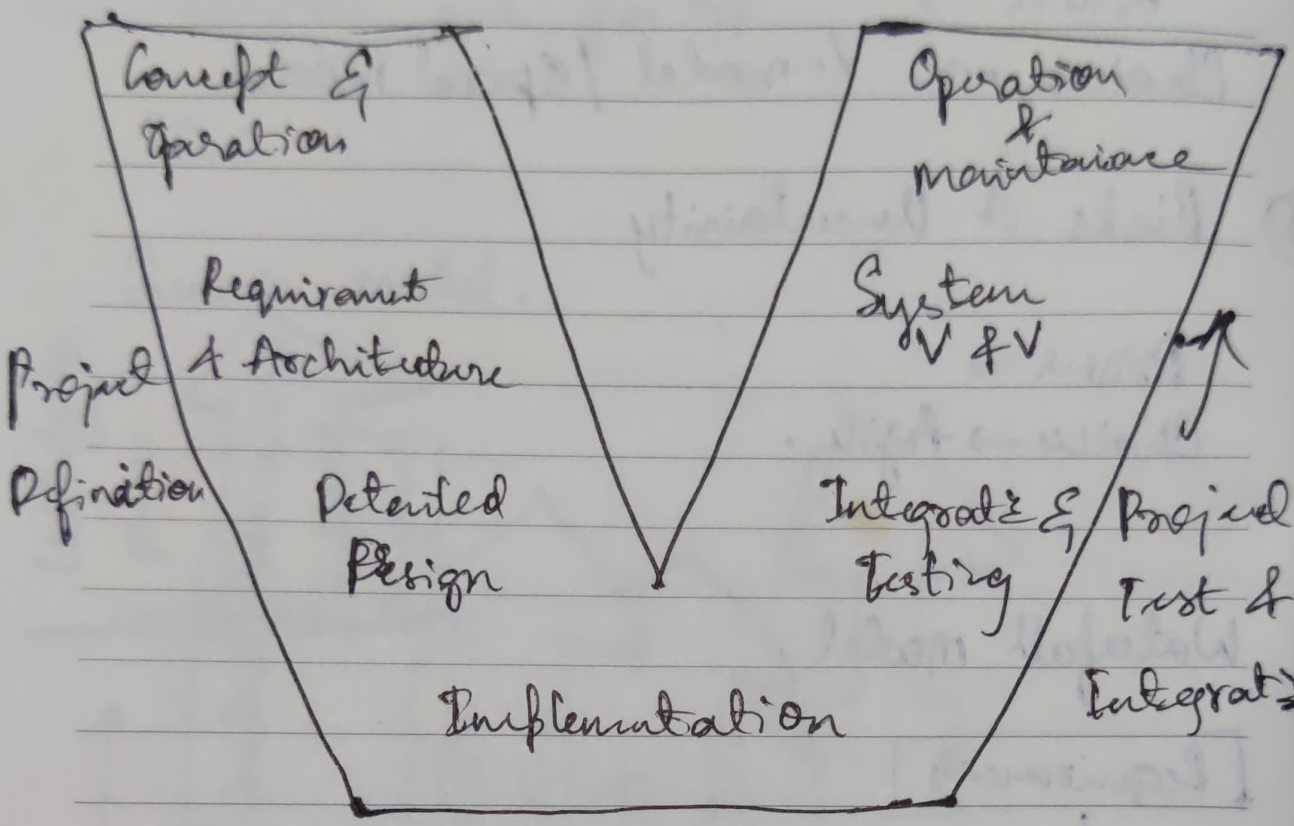
Waterfall model.



œrlikon
balzers
œrlikon
metco

Date: / /

Vee Model.



Iterative model

- Requirement Gathering
- Design
- Implementation
- Testing
- Review Feedback
- Deployment

Date: / /

Incremental Model (Based on requirements. n - number of Releases)

Requirements

A → D → I → T → D → M → Release-3

A → D → I → T → D → M → Release 2

Analyze → Design → Implement → Testing → Deployment → Maintainance → Release 1

AGILE MODEL

Need & Requirement Analysis.

Functional & non-Functional

Goals & objective → expected outcome

System to do → UML Diagrams

performance
Scalable
Adaptable
Available
Secure
Usability
Maintainable
Portability

Date: / /

Case study

Design Robotic surgery Robot.

Stakeholders: Surgeons, Patient, Hospital Admin. Regulatory Authority.

Functional → Robot control interface, Haptic feedback, Surgical Instrumentation Imaging & Visualization (continuous monitoring)

~~Non functional → Imaging &~~ Safety mechanism, communication b/w devices & respective doctors & professional.

Date: / /

SDLC (Basics)

SDLC is structured process followed by software department teams to design, develop, test and deploy high quality software systems efficiently and predictively.

Step by step process that defines the phase involved in developing a software product.

Why?
- Ensure quality & consistency
- Helps in project planning, management
- Reduce poject risk & costs
- Allows for timely delivery
- Ensures the software meets users expectation.

Phase of SDLC.

1. Requirement Gathering & Analysis.
   Goal: Understand what client/user wants
   Activities:-
   Meetings with Stakeholders
   Requirent documents.
   Eg- Client wants e-commerce website with cart & payment option

Date: / /

2. Feasibility study.

Goal. Decide if technically, financially & legally possible.

Eg. Can we build e-comm website with ₹5 lakhs in 3 months.

3. Design.

Goal. Convert requirements into architecture & UI/UX

O/p:

High level / Low level design.

Eg. Designing home page, product page & how data flows in backend

4. Development (coding)

Goal: Developers write actual code based on design.

Tools: VScode, Git, Java/Python etc.

Eg. Frontend developers build UI;
Backend does handle login & dBase.

5. Testing: Goal: Ensure software works as expected

Types: Unit testing, Integration, System

Date: Eg: Test if users can signup & add items to cart & make payment

6- Deployment.

Goal: Make product live for users.

Environment:

Product & server or app store

Eg: Hosting the ecommerce site on AWS or publish on playstore.

7. Maintainance.

Goals: Fix bugs, update features, monitor performance

Eg: Add new payment method or fix login issues.

Really Fine Designers Develop The Dumbest Machines

Requirement Gathering & Analysis

Feasibility Study

Design

Development

Testing

Deployment

Maintainance.

Date: / /

Requirement Gathering.

First step of SDLC
- Understand what client wants from the system.
- Collect details info from stakeholders.
- This prevents misunderstanding
- Helps define scope, time & budget

Who are involved?
- Business Analysts
- Clients / Customers
- End Users
- Project Managers
- Developers & Testers

2 types of Req Gathering based on what do we gather.
1. Functional RG
2. Non-functional RG

1. Functional RG

Features & functionalities expected.

Examples:
- System must allow users to register & log in
- Admin should be able to generate monthly report etc.

Date: / /

Non Functional Requirement (NFR)

These define how system performs - quality attributes or constrains.

Eg:-

- System should load within 3sec
- Appl^{n} must support 10,000 users.
- Data must be encrypted