

Software Engg.

yogesh
KK. Aggarwal

PAGE NO.
DATE:

- Error - syntactical / semantical error
- Bugs - Error executed successfully → leads to garbage
- Fault - Bug executed successfully
- Failure → Fault executed " - leads to failure of software

Software Engineering - designing / developing a set of programs with series of functions performed.

→ systematic approach of designing, developing, operation and maintenance

Q. Roles of a software

- 1) As a product uses Hardware to run.
- 2) As a delivery vehicle :- Operating system

System software Application software
Drivers MS office

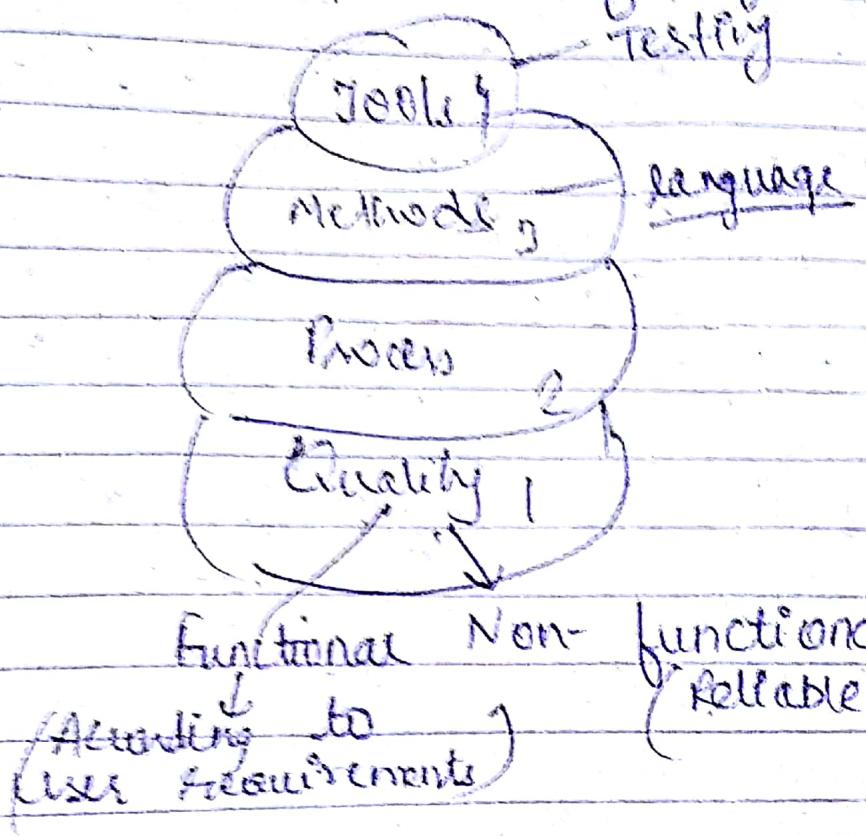
Q. Why Software Engg is useful?

- To enable creating complex programs
- User-friendly, high Quality
- Less chaotic

Q - Focuses on

1. Quality ↑
2. Maintenance ↑

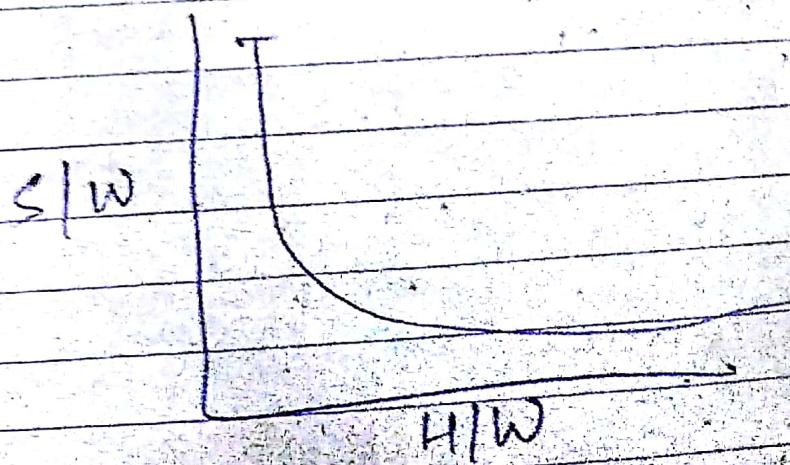
A layered approach of software engineering



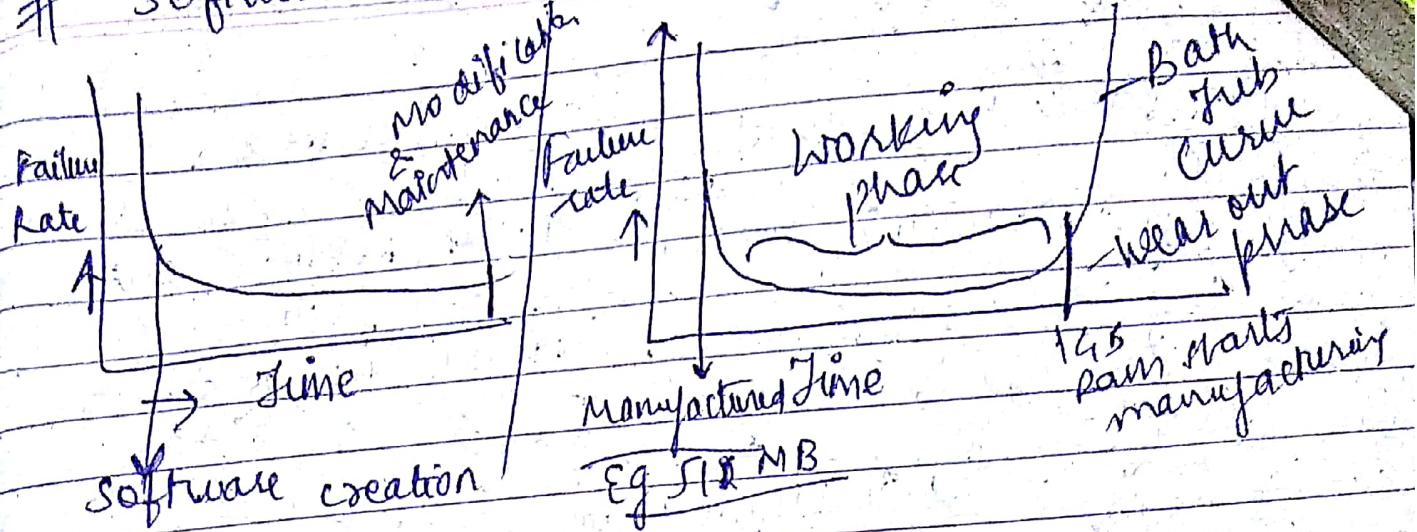
Process -

software Development life cycle (SDLC)

software circs -



Software Vs Hardware



Obsolete - old but not dead
wear out - not in market

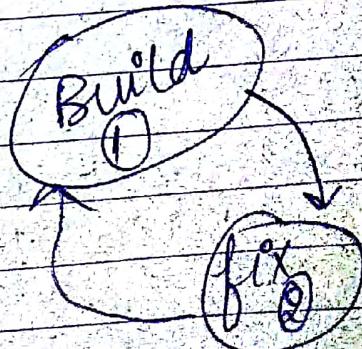
process for software development

SDLC (software development life cycle) :-

1. Build and fix
2. Waterfall Model
3. Incremental Approach
 - (i) Iterative Enhancement model
 - (ii) RAD
4. Evolutionary Models
 - (1) prototyping model
 - (2) spiral model

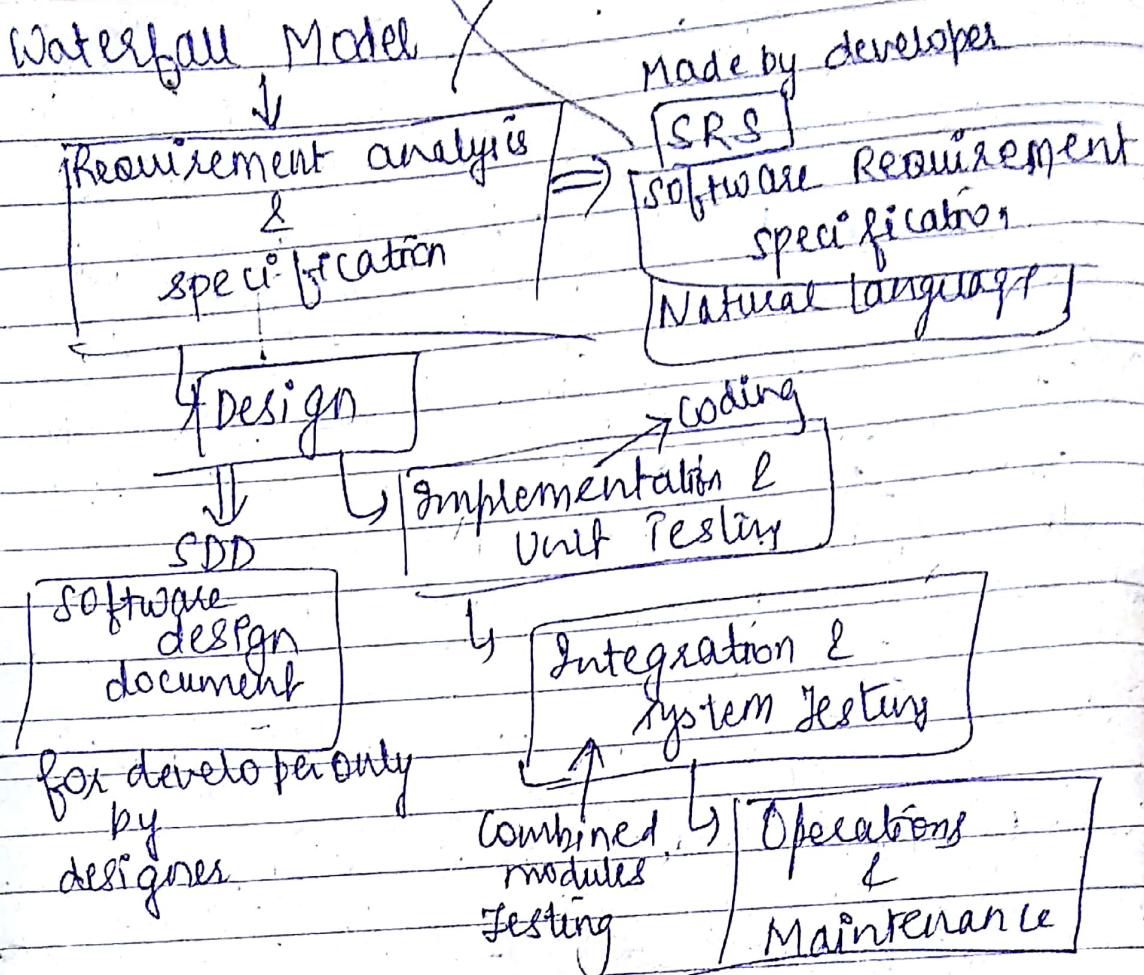
(1) Build & fix

- * only for small software
- * not cost-efficient
- * no provision for user requirement



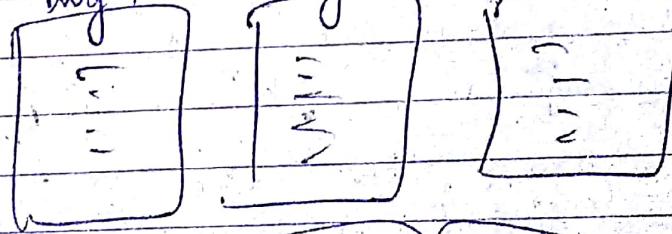
SRS → "what a SW will do not how a SW will do".
Problem statement

Q: Waterfall Model



(Unit Testing) :- Testing smallest unit of software.

Prog. 1 Prog. 2 Prog. 3



Units | Modules

System Testing - After developing the program, we have to deploy it on the system and that testing is called system testing.

Operations - Calling the customer to check software operations at developer site and vice versa

Alpha
Testing at developer site

Beta
Testing at customer site.

Maintaining - future work

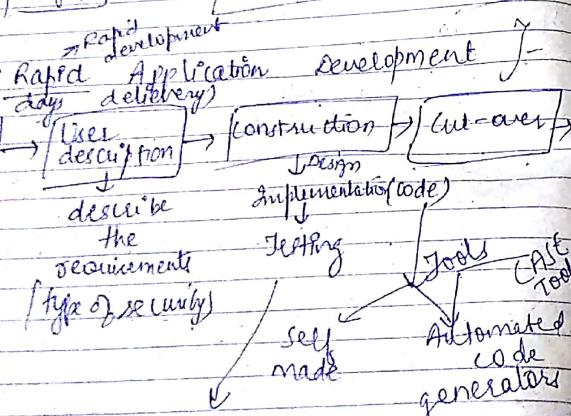
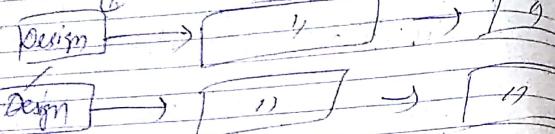
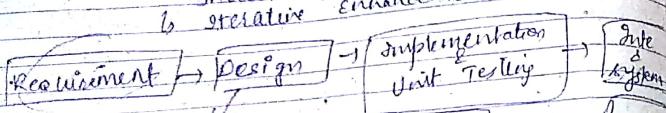
Disadvantage -

1. Expects complete & accurate requirement early in software development process.
2. NO prototype
3. We cannot change the software
4. NO Risk Management

Depends on priorities

Incremental Models

6 iterative enhancement model



Acceptance testing
(Alpha & Beta)

gratuitous → Operation → Release 1

[] → [] → Release 2

[] → [] → Release 3

Active user involvement from first stage to last

Installation of software or operation (deployment)

Verification - Checked by developer

Validation - checked by (user) clients

RAD is not used :-

- ① When user involvement is not there
- ② If we don't have tools | reusable product for making software
- ③ If we don't have highly skilled labour

Evolutionary Models

prototype model

spiral model

1. Prototype model

Prototyping

Requirement

quick Design

Implementation

customer evaluation

Accepted → Customer feedback refinement

Design

Rejected

Implementation
Unit & Testing

Integration &
System testing

Operation & maintenance

similar
to
waterfall
model

Incremental
1. Priorities based

Evolutionary
Categories based

2.

Disadvantage

1. Time-consuming
2. costly (most expensive)

#

21/01/18

Software Engineering

Spiral Model

~~150°~~

1st phase

Determine obj., constraints

R.A. P1 P2 P3

IVth phase
Plan next phaseR.A. P1 P2 P3
Development plan
Integrate & Test

R.A.

concept of operation

Risk analysis

Operational Resolves Risk
prototypeS/W module design
Integration Testing

Detailed Design

Unit Testing

Develop

verify next level produce

Acceptance Testing

1. Planning:

- ↳ Determine objectives
- ↳ Alternatives
- ↳ Constraints

2. Risk Analysis

- ↳ Identification & resolve risk
- ↳ classify

3. Development & Testing

- 4. Assessment: customer evaluation and plan next phase

Disadvantage → Costly

Advantage

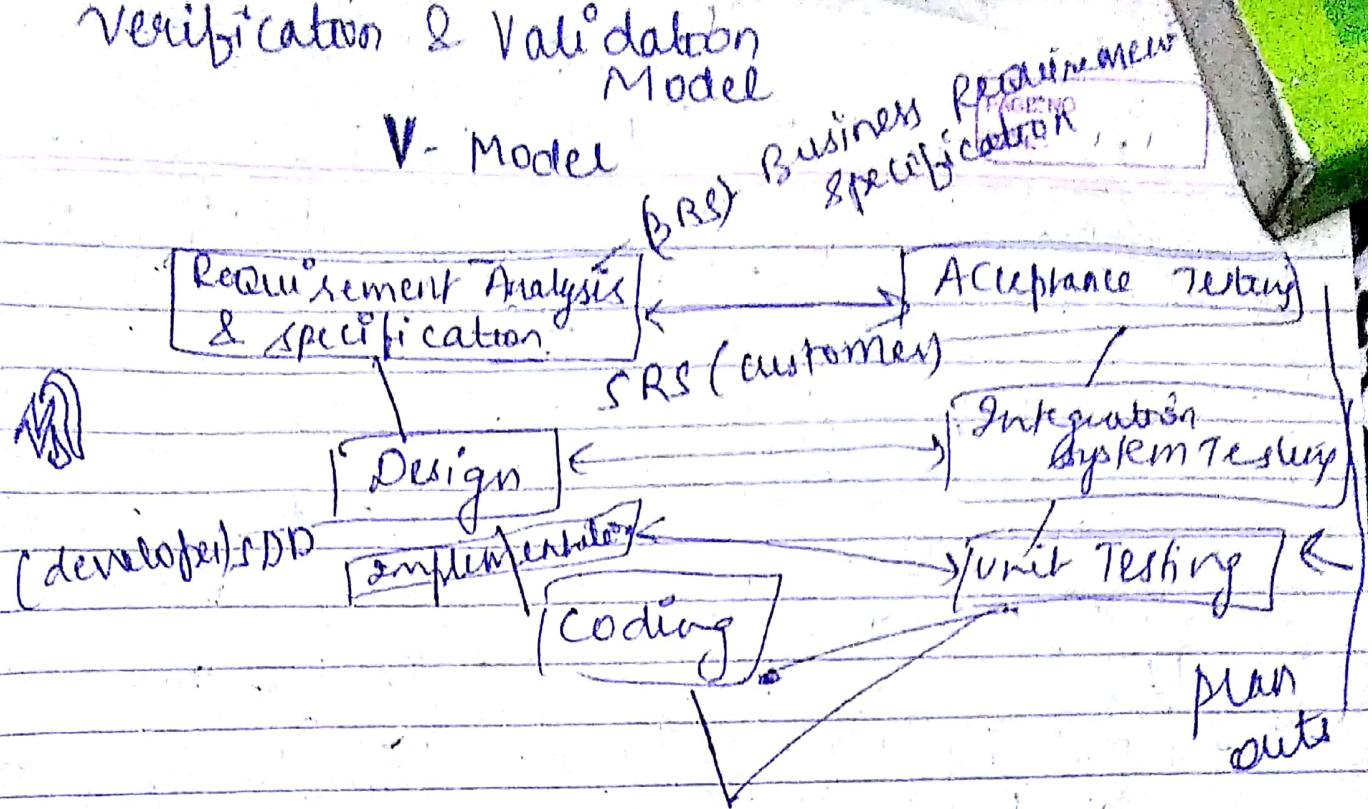
- 1) Accommodates good features of SDLC model
- 2) Software quality is maintained during development.
- 3) By the continuous task analysis and reviews conducted in each cycle software is produced early in the life cycle.
- 4) Strong approval & documentation

Disadvantage

- 1) It is the costly model ever/very high.
- 2) Model is complex.
- 3) Require expertise for Risk Handling
- 4) The spiral may continue indefinitely

Verification & Validation Model

V- Model



Advantages -

1. Very simple and easy to use.
2. Testing activities happens well before the coding which will save lot of time.
3. Defect tracking can be easy to detect in early stages.
4. Works well for small projects/products where req. are easily understood.

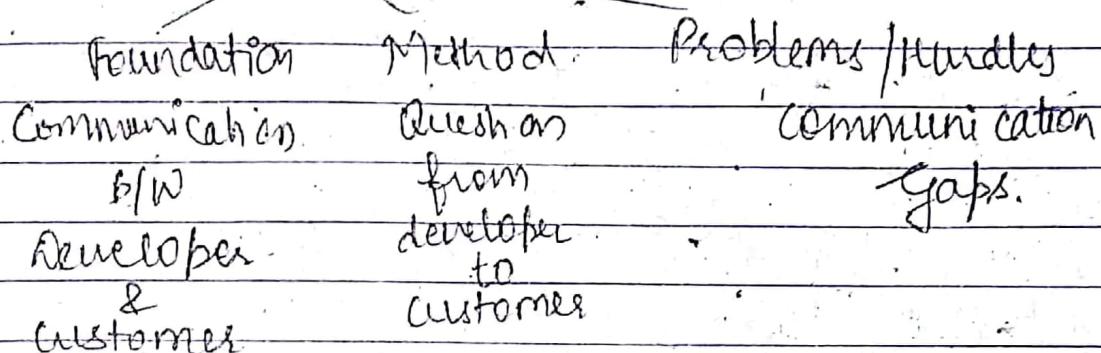
Disadvantage

- least flexible and very rigid.
- no early prototype of software is produced.
- changes come in mid-way of life-cycle, then the test-document along with requirement document not easy to update.

Requirement elicitation (Gathering of Requirements)

- (1) Interviews
- (2) Brainstorming
- (3) F.A.S.T - (Facilitated Application Software Techniques)
- (4) QFD - (Quality Function Deployment)

R.E



(1) Interviews →

Open ended Structured

(list of questions)

No preset agenda is there and we have a set agenda.

No pre-defined list of questions

Project lead

Knowledge criticality

Fresher

of project.

Less experience

~~(2) Brainstorming (Group Discussion)~~

Types of stakeholders for interview →

- 1) Entry level stakeholder (Not much domain knowledge)
- 2) Mid-level stakeholders (Experienced / High knowledge)
- 3) Manager and high-level authorities
- 4) Users of software

stakeholder - person who is directly or indirectly connected with making software

(2) Brainstorming (Group Discussion)

Sorting is done
Priority-wise.

Members Facilitator
Manager Manager | Manager
 Authority

(3) FAST (Facilitated application software technique)

- (1) Identify the problem
- (2) Solution requirement
- (3) Propose elements of S/W
- (4) Negotiate different approaches

set of rules

maintenance period

fix agendas

(4) QFD (Quality Function Deployment)

Natural language requirements converted technically required facilities.

Design

1. Designing of S/W

UML

Unified modelling language

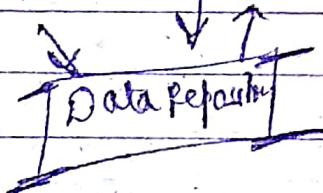
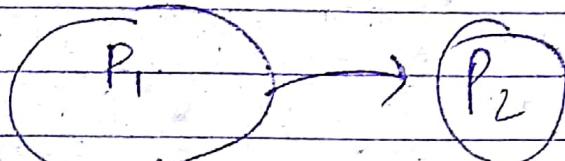
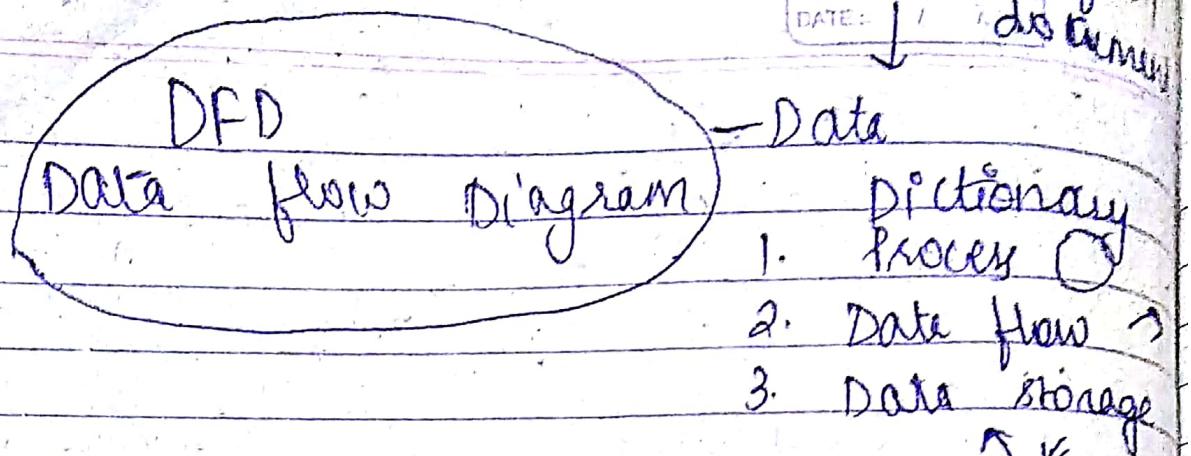
- 1) Use case
- 2) Activity diagram
- 3) Sequence diagram
- 4) Class diagram
- 5) Collaboration diagram
- 6) Deployment diagram

Star UML (open source software)
(earlier we used Rational Rose)

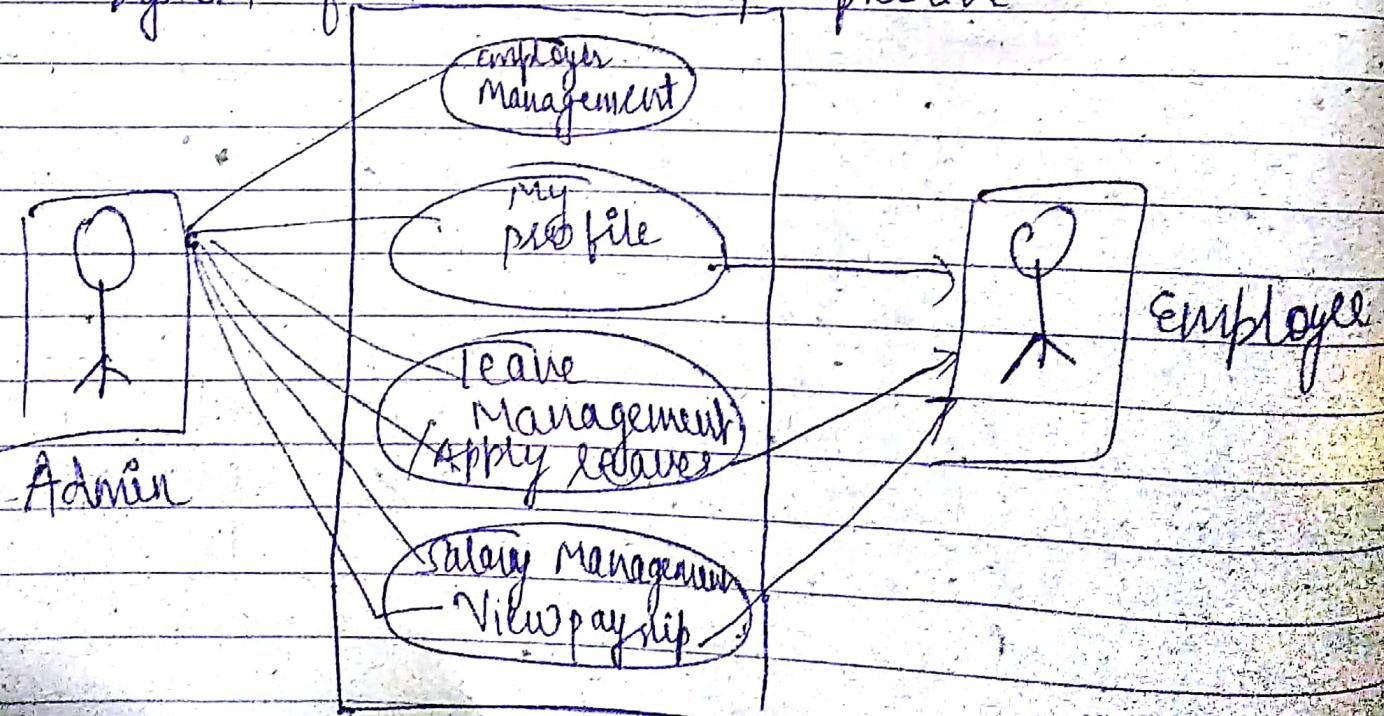
Entity-Relationship diagram

Types of attributes

1. Single	Multivalued (either value)
2. Simple	composite (all the values)
3. Key valued	(uniquely defined)
4. Defined	{Age} — D.O.B



USE CASE :- Shows the complete use of system from User's perspective



Characteristics Of Good SRS :-

1. consistent - No conflicts b/w requirements
2. completeness
3. unambiguous
4. correct
5. Traceable
6. ~~Verifiable~~ Verifiable
7. Ranked for stability / importance
8. Modifiable

Traceable —

