

## Software Engineering

### Need of Software Engineering

- Some failures (problems)
- (i) Y2k Problem in 2000
  - This problem was generated in year 2000 b/c in some softwares the format of year was only last 2 digits.

- (ii) Ariane - 5 rocket space problem
  - Ariane - 5 space rocket developed at a cost of 300 millions dollar over a 10 years period.
  - The rocket was destroyed after 39 sec. of its launch.
  - When guidance system own computer try to convert one piece of data from 64 bit format to 16-bit format the program failed and overflow error occurred.

- (iii) Star Wars problem
  - The petrate missile fails 7 times to hit scud missile including 1 that killed 128 US soldier in Saudi Arabia.
  - The reason was software bug due to small timing error in systems clock.

- (iv) US consumer database problem.

This project was developed in 1 million dollar to replace its old customer database into a new customer database software but within 3 weeks the dB was shutdown.

The reason of failure was design team was over optimistic in collecting the requirements.

(v) Microsoft Windows XP problem  
 Launched in (25 October 2001).  
 Microsoft released window xp in 2001 and on  
 the same day the company posted 918 types of patches  
 on its backside for bug fixes.

#### (vi) IBM Report:

As per the IBM report 31% of the projects get  
 cancelled before they are completed and 53% are  
 over run in their cost estimates so the success ratio  
 was 16% of any software will fail in cost.



Software Engineering:  
 Software Engineering deals with concept, strategies,  
 concepts to avoid conflict and to improve Software  
 development process in order to deliver good quality  
 maintainable software in time and within budget.

#### (vii) SDLC (Software Development Life Cycle)

(i) Requirement analysis and Specification → feasibility study

##### ① SRS : Software Requirement Specification

In requirement analysis and specification fails company  
 gathers all the requirements of the client, including the  
 development time and cost, and writing a document  
 that is known as SRS. It will be the starting point.

② Feasibility Study → In feasibility study company  
 checks that whether the software is feasible to develop  
 on the basis of time, cost and technology.

(ii) Software Project planning: In this company plan all the activity of software development acc. to time, cost and human resource.

(iii) Software Designing.

(1) SDD: Software design Document

In this ways company design the entire software with the help of diff. tools and the output of this phase is SDD.

(iv) Coding / Implementation, it gives detailed two main types methods work with specified tool

(v) Testing

(vi) Operation & maintenance

(v) Software Vs Programs.

A software is a collection of computer programs, documentation and operating procedures.

Documentation manual contains Analysis / specification, design, implementation and testing (for maintenance)

The operating procedures contain user manual & operational manuals.

## Software Products.

It may be of 2 types

- (i) Generic → developed to be sold to a range of diff. customers.
- (ii) Bespoke (Custom) → developed for a single customer according to their specification.

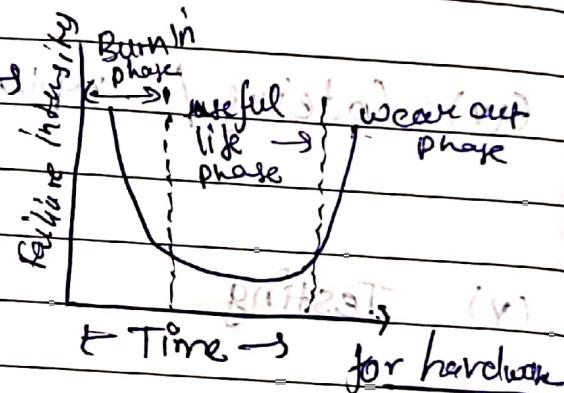
→ Software is not a physical product.

→ Software Characteristics:

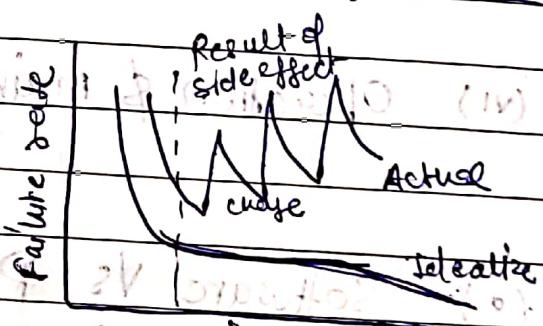
- (i) Software doesn't wear out.

Bath-tub curve for hardware.

Software doesn't wear out  
but hardware does wear out.



Failure intensity curve for software.



- (ii) Software is not manufactured.

→ Hardware can be manufactured but a software is developed.

- (iii) Reliability of Components.

Software is flexible.

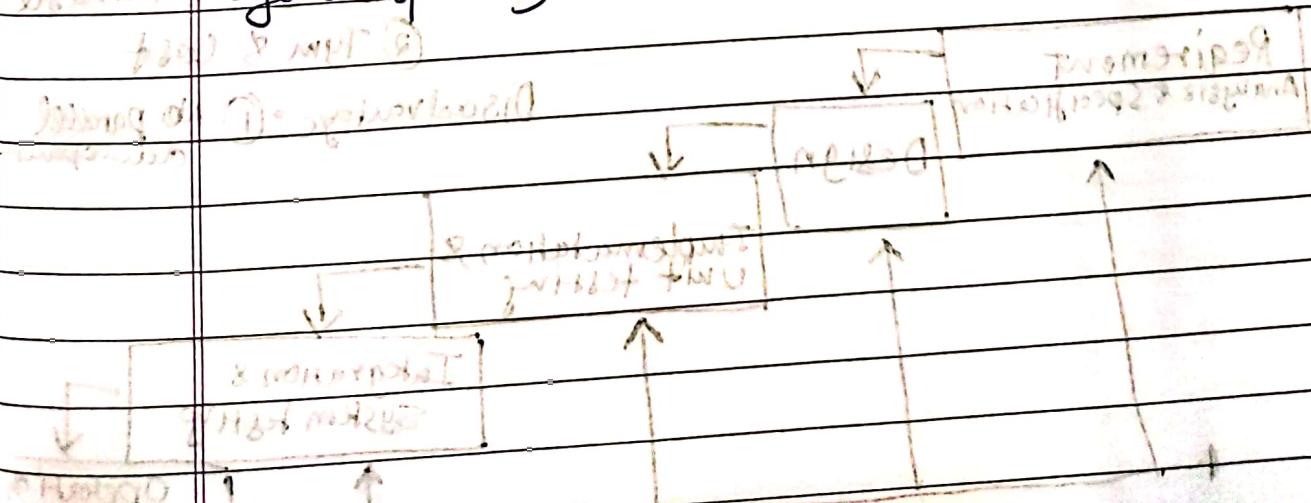
- (i) The Changing nature of Software (Types).
- (ii) System Software → as interface b/w app & hardware.
- (iii) Real time Software → *Job shop & Job shop*
- (iv) Embedded Software
- (v) Business Software
- (vi) Personal Computer Software
- (vii) Artificial Intelligence Software
- (viii) Web based Software
- (ix) Engineering & Scientific Software

### (c) Some technologies

Process VS Product: *Job shop & Job shop*

Product: What is delivered to the customer is called a product. It may include source code, specification document, manuals, documentation etc. Basically it is nothing but a set of deliverables only.

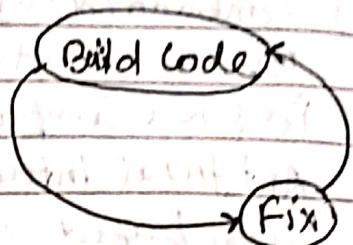
Process: A process is the way in which we produce software. It is the collection of activities that leads to a part or a product. An efficient process is required to produce a good quality product.



## (c) Software Lifecycle Models / development models

### 1.) Build & Fix Model.

- Product is constructed without specifications or any attempt at design.



- Adhoc approach and not well defined.

- Simple two phase model,

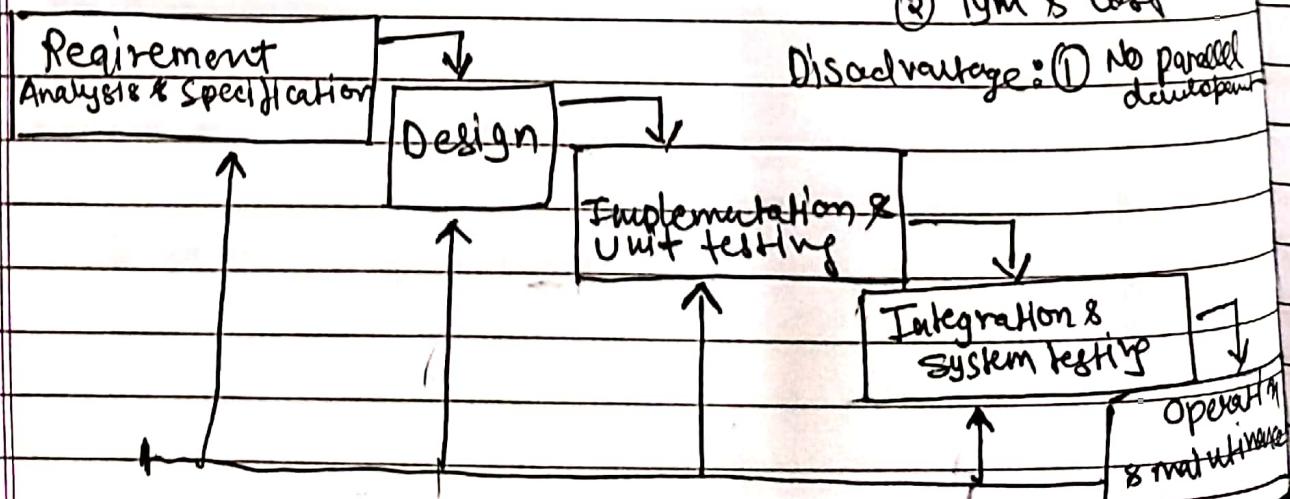
- i) This model is feasible when requirements of a user are very less.
- ii) It is simply a two phase model.
- iii) Suitable for small programming exercise of 100-200 lines.

- iv) The problem with this model was code soon become unfixable and unenhanceable ∵ a software is not maintainable.

### (2) Waterfall Model

Advantage: ① maintainability

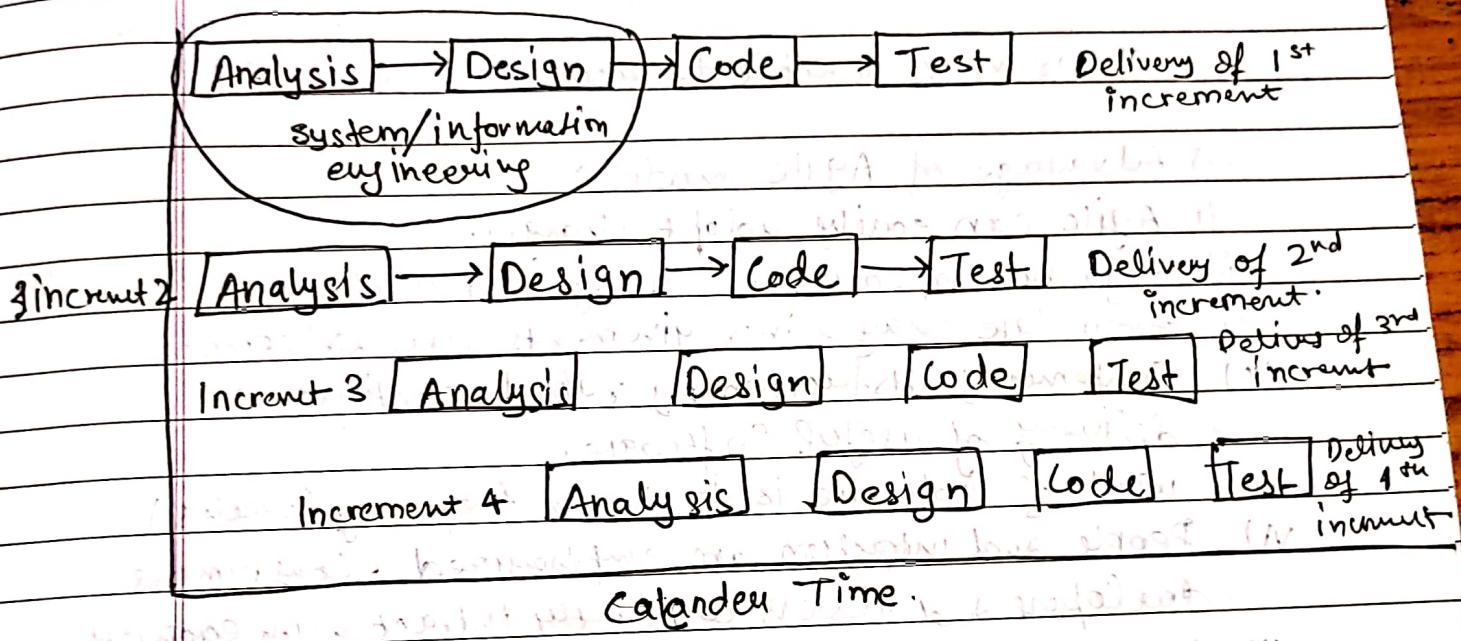
② Time & Cost



Disadvantage: ① No parallel development

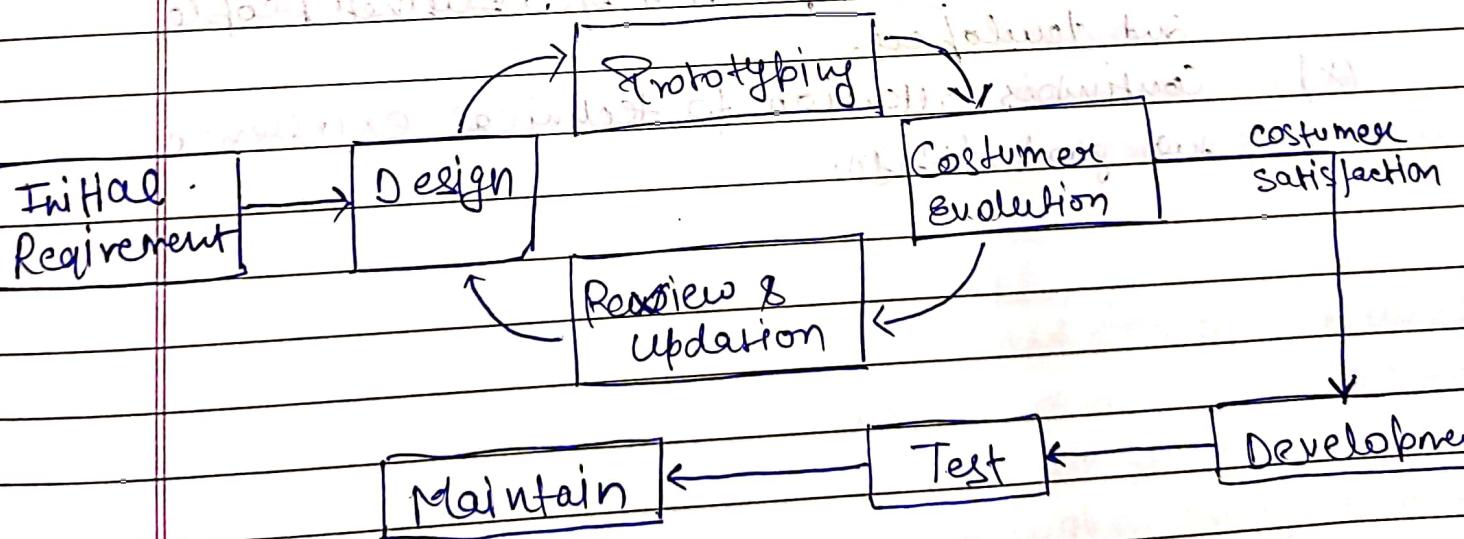
Unit testing - Testing of a individual module is known as unit testing..

### (3) Increment Process Model : Iterative Enhancement Model



### (4) Prototyping Model

(i) user is confused about requirements, he withdraws some requirements.



(c) Combination of all project.

(5) The Spiral Model

(6) Agile Software Development Model.

• used by the

It is very flexible to accomodate changes.

(c) Advantage of Agile model:-

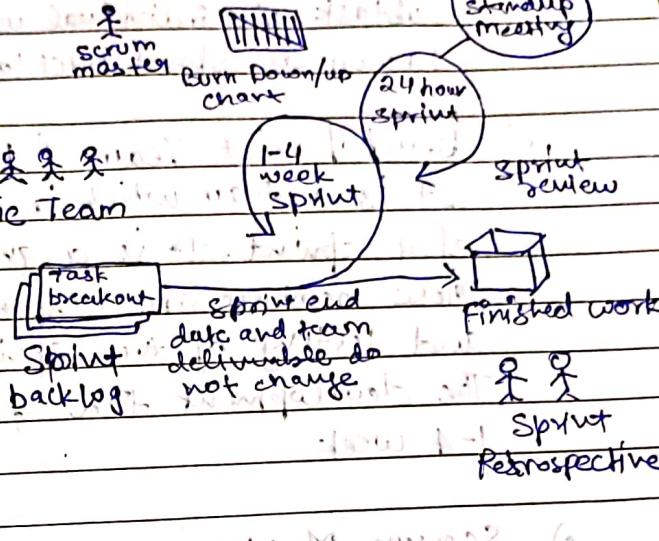
- i) Agile can easily adapt changes.
- ii) Regular adaptation to changing circumstances.
- iii) Even late changes in requirements are welcome.
- iv) Customer satisfaction by rapid, continual delivery of useful software.
- v) Working software is delivered frequently (weeks)
- vi) People and interaction are emphasized. Customers, developers and testers constantly interact with each other.
- vii) Face to face conversation is the best form of communication.
- viii) close, daily cooperation b/w business people and developers.
- (ix) Continuous attention to technical excellence and good design.

## The Agile Scrum Framework at a glance

Inputs from customer,  
Team Manager, Execs

↓  
Product owner, The Team

Product Backlog  
Sprint Planning meeting



### • Product Owner

Product Owner is a person from client side who is responsible for delivering all the requirements of the software to the company.

### • Product Backlog

All the requirements that are gathered from the product owner will be written in product backlog in the form of user stories.

### • User Stories

format:

Ex:- As a student  
I need to make payment

Agar vasti refom of Insta  
I need to today to instal

So that I can submit my fees.

So that I can submit my fees.

### • Sprint planning Meeting

In this space one user story is picked from product backlog and meeting will be held to discuss what subtask development team need to perform to complete a user story.

\* The user story which is having the highest priority is taken.

.) Sprint backlog (Sprint → module)  
Subtask of a individual user story are written in a backlog known as sprint backlog

.) Development Team (Dev team)  
It is a team which is responsible for development of a sprint. It is a group of 2-8 members. These members are responsible for designing, coding and testing of the sprint.  
The development time of a sprint is fixed from 1-4 weeks.

.) Scrum Master

Scrum master is responsible to manage the dev team and entire development work like Team Lead (TL)

.) Daily Standup Meeting (Daily-Scrum)

Daily standup meeting will be held once in 24-hour to discuss any problem faced by dev team. The meeting will be held only for (15 min).

.) Burn Down/Up chart

This chart is used to monitor the progress of project, i.e. what task is completed & what are the subtask which are not completed.

.) Sprint Review

Show final product to the product owner.

.) Sprint Retrospective

## (i) Requirement analysis and specification

In this phase, company gathers the requirement from the client by considering the following steps.

Requirement → Requirements → Requirements → Requirements  
Elicitation      Analysis      Documentation      Review

## (ii) Requirement elicitation

Types of Requirements:

(i) Known requirement

(ii) Unknown requirement

(iii) Undreamed requirement

Classification of requirements:

(i) Functional requirement

They describe what software has to do. They are often called product features.

(ii) Non-functional requirements (quality of software)

They are mostly quality requirements. They stipulate how well the software does what it has to do.

Availability

Reliability

Usability

Flexibility

for User

Maintainability

Portability

Testability

for developer

- (c) User and System requirements
- User requirements are written for the users and include functional and non-functional requirement.
  - System requirement are derived from user requirement.
  - The user system requirement are the parts of software requirement and specification (SRS) document.

### (i) Requirement Elicitation

In Requirement Elicitation company uses different methods to gather requirements of user with the help of different methods.

- 1.) Interview
- 2.) Brain storming Session / Group discussion
- 3.) Facilitation Application specification technique (FAST)
  - Similar to brain storming session
  - Team oriented approach
  - Creation of joint teams of customers & developers.

### 4.) Quality function Deployment (QFD)

This method incorporate voice of the customer.

- In this method requirements are categorized in following 3 categories

i) normal requirement

ii) expected " "

iii) existing " "

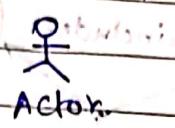
In this method, requirements are prioritized on the basis of requirement given by user.

## 5.) The Use Case Approach

→ Use case diagram is proposed by Ivar Jacobson & others introduced use case approach

for elicitation & modeling.

Notations



Actor



Use Case

Relation b/w actor & use case  
and/or b/w two use cases.

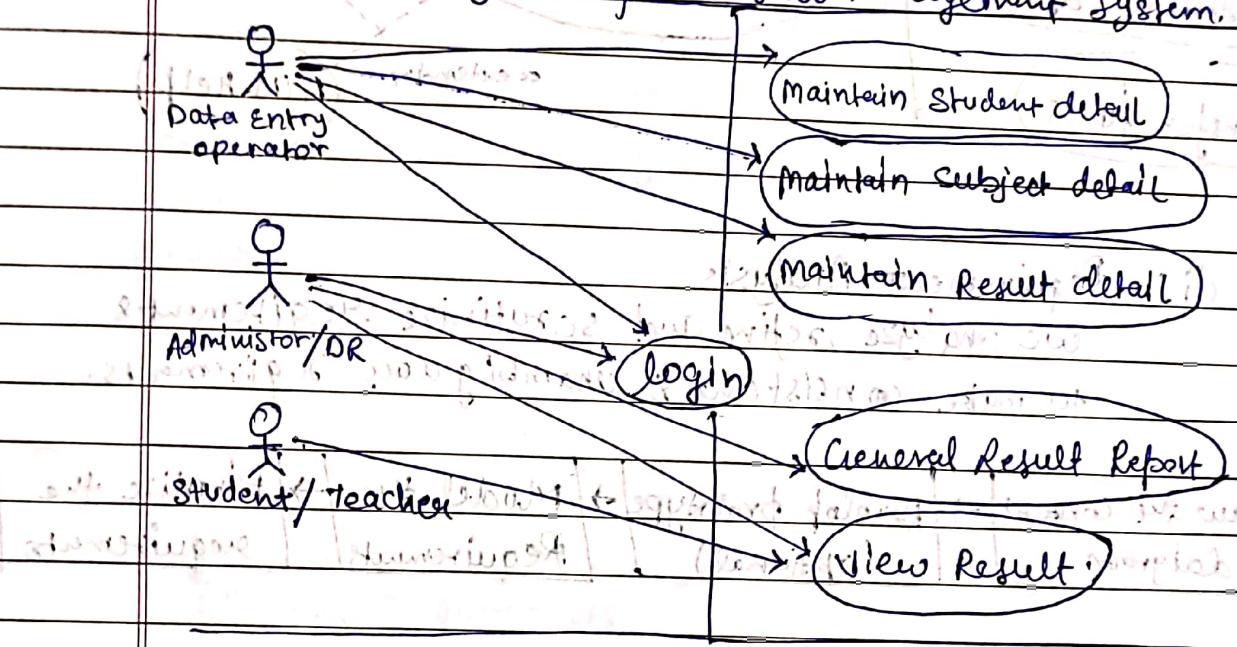
Relationships

Generalization

Specialization

Q

use case diagram for Result Management System.



Steps of use case diagram:

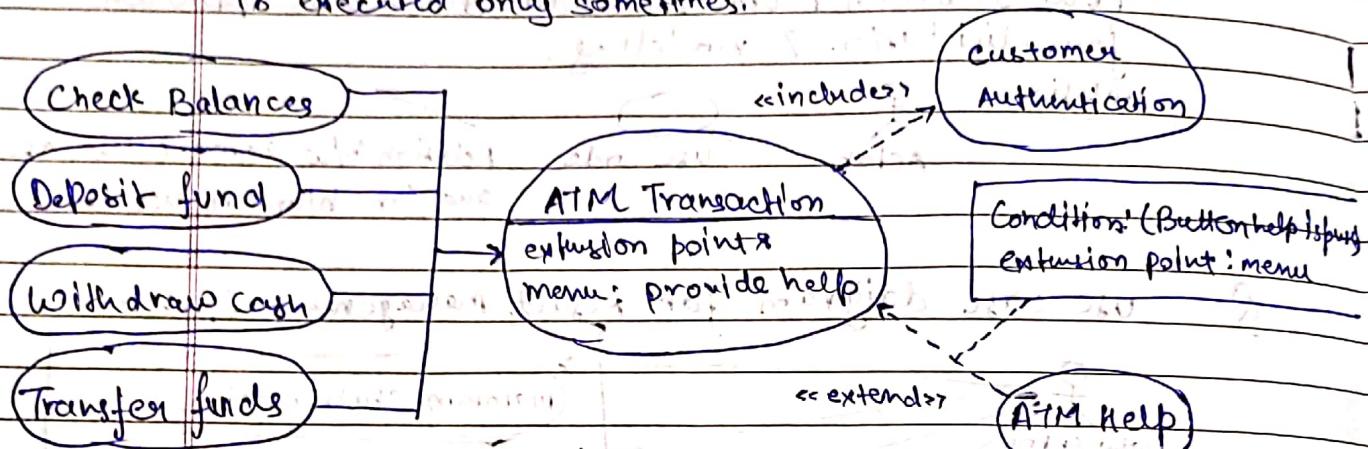
- 1.) Identify the module based on functional requirement.
- 2.) Identify the actor or in morghish terms.
- 3.) Actor interact with which module.

Use case diagram can be extended using include and extract extend a feature of small size.

When base usecase is executed the included usecase is executed everywhere time but we should note it.

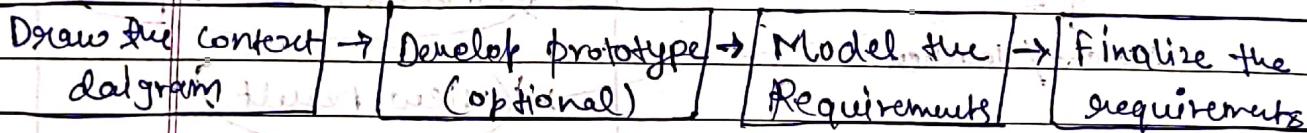
Planning up to level of reasonably much

When base usecase is executed the extended usecase is executed only sometimes.



## (ii) Requirement Analysis

We analyze, refine and scrutinize requirements & to make consistent & unambiguous requirements.



### (a) Draw the Context Diagram

In this phase, context diagram (or level-0 DFD (data flow diagram)) is created. Context diagram is also known as level-0 DFD.

### (b) Develop prototype

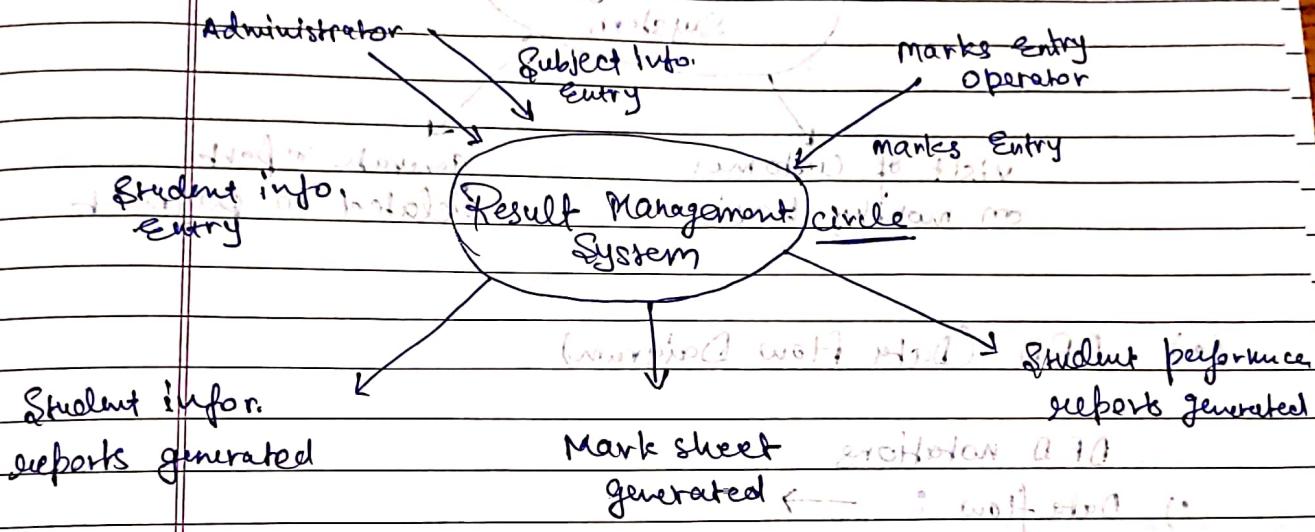
If requirements are not clear then we can use this phase to develop a prototype.

### (c) Model the Requirements

In this phase we develop DFD and ER diagrams and data dictionaries to model the requirements.

(e) Context diagram.

In context diagram we show the entire system as well as no. of inputs in the system & no. of outputs from the system.

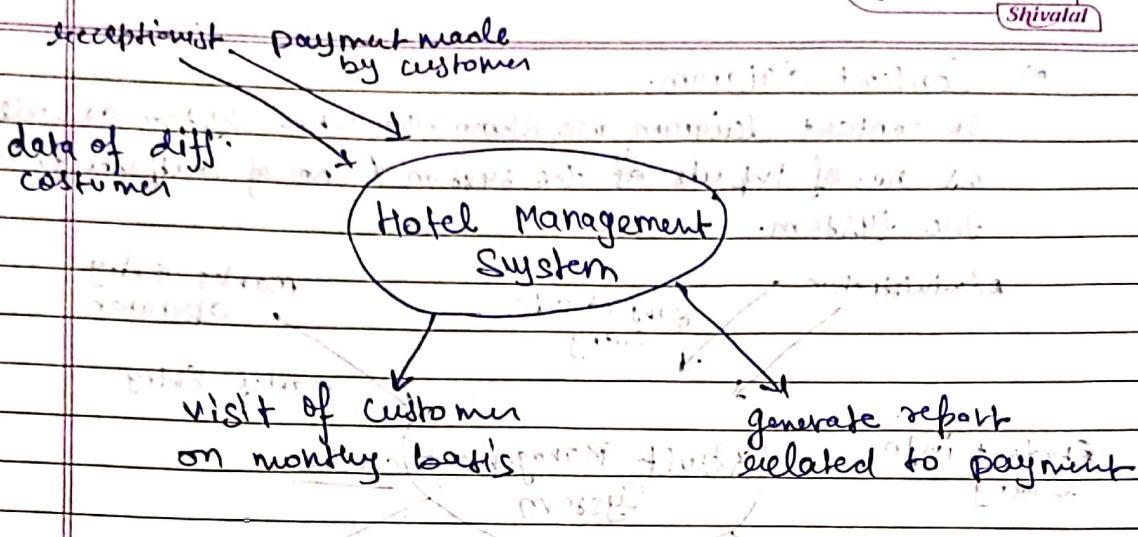


Ques

Suppose Company wants to develop a software for hotel management system by considering the following requirements

- 1.) In software receptionist can insert data of diff. customers.
- 2.) Receptionist can insert data related to the payment made by customer.
- 3.) Manager can generate the report related to visit of customers on monthly basis.
- 4.) Company owner can generate reports related to payment details of the customer.

Create context diagram.



### (c) DFD : (Data Flow Diagram)

DFD notations

i) Data flow :  $\rightarrow$

ii) Source & sink of info/ data (also called external entity)

iii) Process

iv) Available data (data store)

Customer

Order

Product

Supplier

Employee

Customer

Order

Product

Supplier

Ques Consider a software for restaurant management system and create content diagram on Level-0-DFD by considering the following activities.

- 1.) Create a list of activities

Customer Order

Serve Product

Collect payment

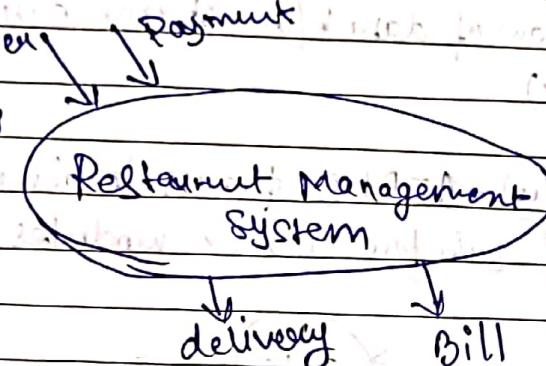
Produce product

Store product

Order raw material

Pay for raw material

Pay for labour



Customer order → collect payment

Customer order

R M S

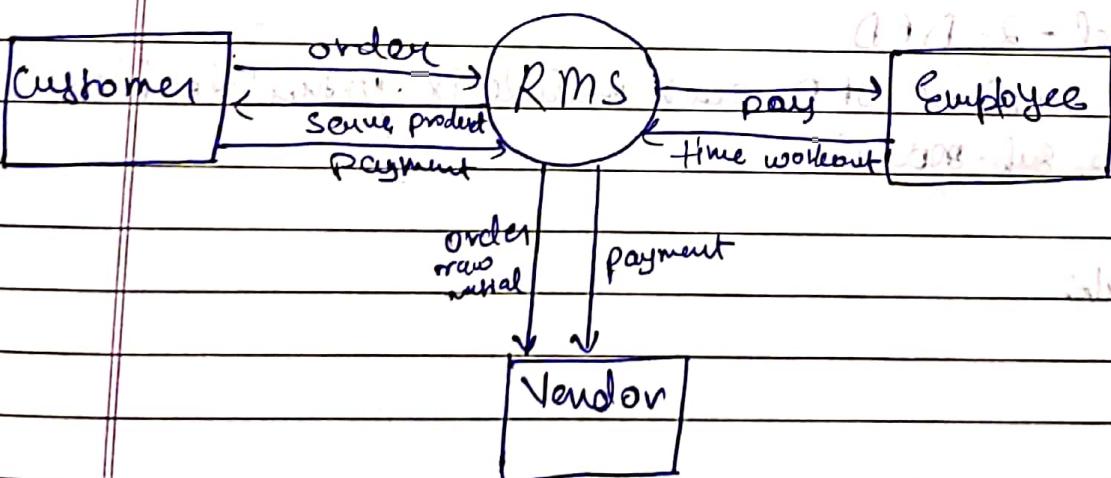
→ Pay for labour

pay for

pay for raw material

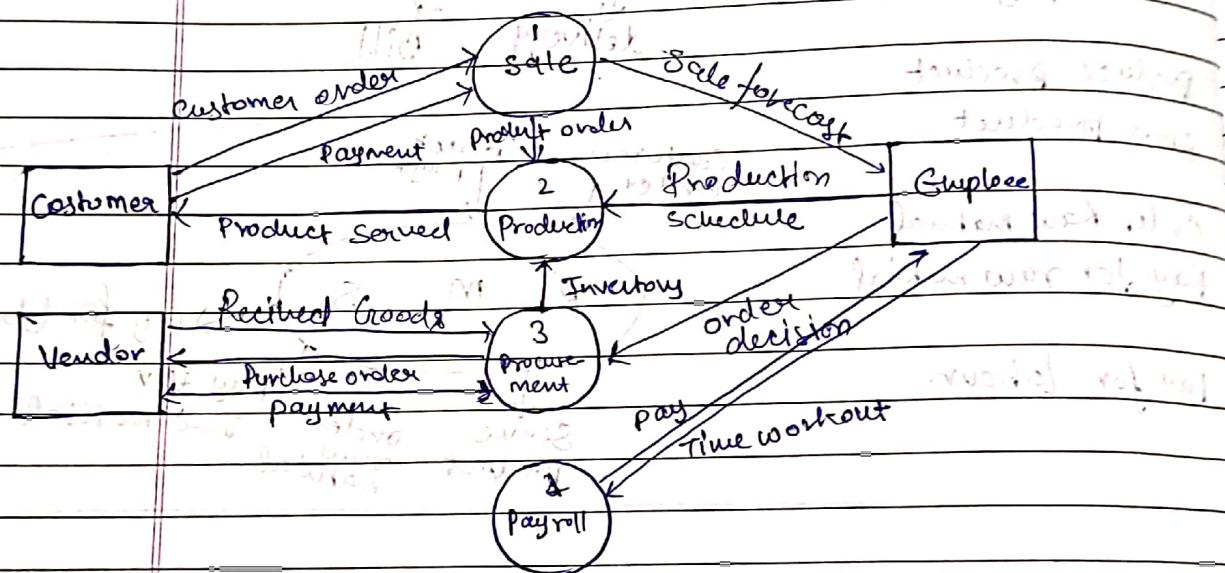
Serve product

order raw material



(\*) Level - 1 - DFD

- (i) In level-0-DFD we show diff external entities and flow of data b/w external entities & process or (system)
- (ii) In Level-1-DFD: process is decomposed into no. of sub processes or modules.



(\*) Level - 2 - DFD

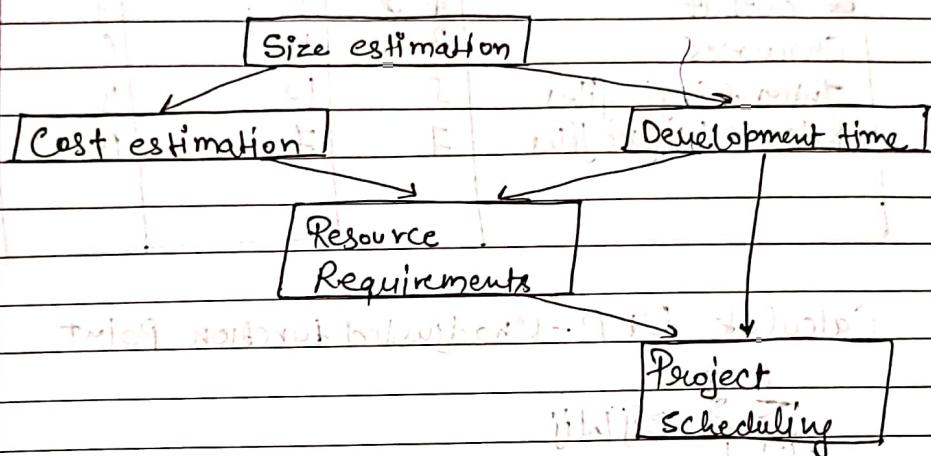
In Level-2-DFD, each module is further divided into sub-modules.

(\*) Sale

Product

(c) Software Project Planning

Software planning begins before technical work starts, continues as the software evolves from concept to reality, and culminates only when the software is retired.



(i) Size Estimation

Method 1: Lines of Code (LOC)

If LOC is simply a count of the no. of lines then figure given below contains 18 LOC.

$$12 + 10 + 7 + 3 = 32$$

Method 2: Function Count / Function Point Analysis.

Step-1 In this method system is decomposed into 5 functional units.

- (i) Inputs
- (ii) Outputs
- (iii) Enquiries
- (iv) Internal logic files
- (v) External interface files

14 Questions

10 Simple  
4 Moderate  
3 Difficult

Step 2: Assign weighting factors to each functional units.

Functional Units	weighting factor
External Input (EI)	3 (Low)
Output	4 (S)
Enquiries	3 (Low)
Internal logic files	5
External Interface files	7

Step 3 Calculate UFP - Unadjusted function Point

$$\sum_{i=1}^n \sum_{j=1}^{m_i} z_{ij} w_{ij}$$

Step 4 Calculate FP (Function Point)

FP = UFP \* CAF (Complexity Adjustment factor)

$$CAF = 0.65 + 0.01 * \sum F_i$$

Rate of  $F_i$

- 0 - No Influence
- 1 - Incidental
- 2 - Moderate
- 3 - Average
- 4 - Significant
- 5 - Essential

14 Questions

$$C = 128$$

$$C^{\alpha} = 29$$

Date / /  
Page No. \_\_\_\_\_  
Submitted \_\_\_\_\_

Ques Consider a project with the following functional units.

No. of user input = 50

No. of user output = 40

No. of user enquiries = 35

No. of user files = 06

No. of external interface = 04

Assume all complexity adjustment factors and weighting factor are average. Compute the function point for project.

Soln

$$FP = UFP * CAF$$

$$UFP = 50 \times 4 + 40 \times 5 + 35 \times 4 + 06 \times 10 + 4 \times 7 = 623$$

$$CAF = 0.65 + 0.01 \times (14 \times 3) \rightarrow 1.07$$

$$FP = 671.96$$

To find out the size in Loc we need to multiply FP with Language multiplier in which software is going to be developed.

Suppose the software is going to be implement in c language then the size of the project will be

$$LOC = 671.96 \times 128$$

$$= 86010.88 \text{ loc } \cancel{+ 1/1000}$$

$$\Rightarrow 86 \text{ k loc}$$

Ques An application has the following : 10 low external input,

12 high external output, 10 low internal logical files, 15 high external interface files, 12 average external inquiries and a value of complexity adjustment factor of 1.10. What are unadjusted and adjusted

$$UFP = 10 \times 3 +$$

$$= 452$$

$$FP = UFP \times CAF$$

$$= 452 \times 1.10 = 497.2$$

Ques Consider a project with the following parameter

(i) External Inputs

- (a) 10 with low complexity
- (b) 15 with average complexity
- (c) 17 with high complexity

(ii) External Outputs

- (a) 6 with low complexity
- (b) 13 with high complexity

(iii) External inquiries

- (a) 3 with low complexity
- (b) 4 with average complexity
- (c) 2 with high complexity

(iv) Internal logical files

- (a) 2 with average complexity
- (b) 1 with high complexity

(v) External interface files

- (a) 9 with low complexity

$$UFP = 10 \times 3 + 15 \times 4 + 6 \times 17 + 6 \times 4 + 13 \times 7 + 3 \times 3 + 4 \times 4 +$$

$$2 \times 6 + 10 \times 2 + 1 \times 15 + 9 \times 7$$

$$\underline{UFP = 424}$$

$$f_i = 4+5+2+0 \rightarrow (0 \times 3) \\ \Rightarrow 41$$

$$CAF = 0.65 + 0.01 + 41 \\ = 1.06$$

$$FP = UFP \times CAF \\ = 1.06 \times 424 \\ \Rightarrow 449.44$$

$$\therefore FP = 449$$

(c) Suppose the software is going to be implemented in C language then calculate size in LOC, effort, duration.

$$LOC = 449 \times 128 \Rightarrow 57472$$

Size of software - 57 K LOC	DOC = 1 = 1156.54	$E = 1.4 \times 57$ $\Rightarrow 60.13 \text{ p.m.}$
		$D = 13.16 \text{ months}$

### (c) Cost Estimation

i) static, single Variable Model (Sel model)

$$E = 1.4 L^{0.93} \quad (\text{in person months})$$

$$DOC = 30.4 L^{0.90}$$

$$D = 4.6 L^{0.26} \quad (\text{in months})$$

## a (c) Static, Multivariable models (Valdson & Falice model)

$$E = 5.2 L^{0.91}$$

$$D = 4.1 L^{0.36}$$

### (3) The Constructive Cost Model (COCOMO)

This method, we have 3 diff submodels.

- (i) basic
- (ii) intermediate.
- (iii) Detailed

In this method each project is classified into any one of three different categories

(i) Organic mode (size - 2-50 kLOC)

(ii) Semidetached mode (size - 50-300 kLOC)

(iii) Embedded mode (size - over 300 kLOC)

Formulae  $E = a_b (kLOC)^{b_b}$

$$\Rightarrow 3.0 (S7)$$

$$D = c_b (E)^{d_b}$$

$$D = 17.91 \text{ month}$$

Software project	$a_b$	$b_b$	$c_b$	$d_b$
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

(c) Average Staff size

$$SS = \frac{E}{D} \text{ person}$$

$$\text{Productivity (P)} = \frac{KLOC}{E} \text{ KLOC/PM}$$

Q Consider a software project for banking system considering the following characteristic, 10 low input, 20 high output, 30 <sup>any</sup> inquiries, 40 internal logical files, 50 external interface. Calculate Effort(E), duration, average staff size and productivity of the project using Cocomo model, where the project is going to be implemented in C language.

$$UFP = 10 \times 3 + 20 \times 7 + 30 \times 4 + 40 \times 10 + 50 \times 7$$

$$UFP = \underline{1040}$$

$$CAF = 0.65 + 0.01 * 14 \times 3 \\ = 1.07$$

$$FP = 1040 \times 1.07 \\ = \underline{1112.8}$$

$$LOC = 1112.8 \times 128 \rightarrow 142,438.4 \text{ LOC} \\ 142 \text{ KLOC}$$

$$E \Rightarrow 3.0 \times 142^{1.12} \rightarrow 772.11$$

$$D \Rightarrow 2.8 \times \cancel{10}(772) \rightarrow 25$$

$$SS = \frac{772.11}{25} \rightarrow$$

$$P = \frac{142}{772.11} \rightarrow$$