

(Puja Miss)

Software Engineering

IEEE defn :-

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of the software.

It is a systematic, disciplined, cost-effective techniques, for software development.

Puja Miss
1st class Software Engineering

What is a good software?

Connectness :- The ability of the software to perform its intended tasks effectively and meet user requirements.

Usability :- The ease with which users can learn, operate and navigate the software.

Reliability :- The software consistency in producing accurate results and maintaining performance over time.

Efficiency :- The optimal use of system resources such as memory and processing power to achieve desired outcomes.

Properties

Maintainability :- The ease of updating modifying and fixing software to accommodate changing requirements or fix issues.

Portability :- The ability of the software to operate on different platforms or environment without significant modification.

Scalability :- The software's capacity to handle increased workloads or user demands without compromising performance.

Security :- The software's ability to protect against unauthorized access, data breaches and other potential threats.

Modularity :- The degree to which the software's component into separate manageable units that can be independently developed or updated.

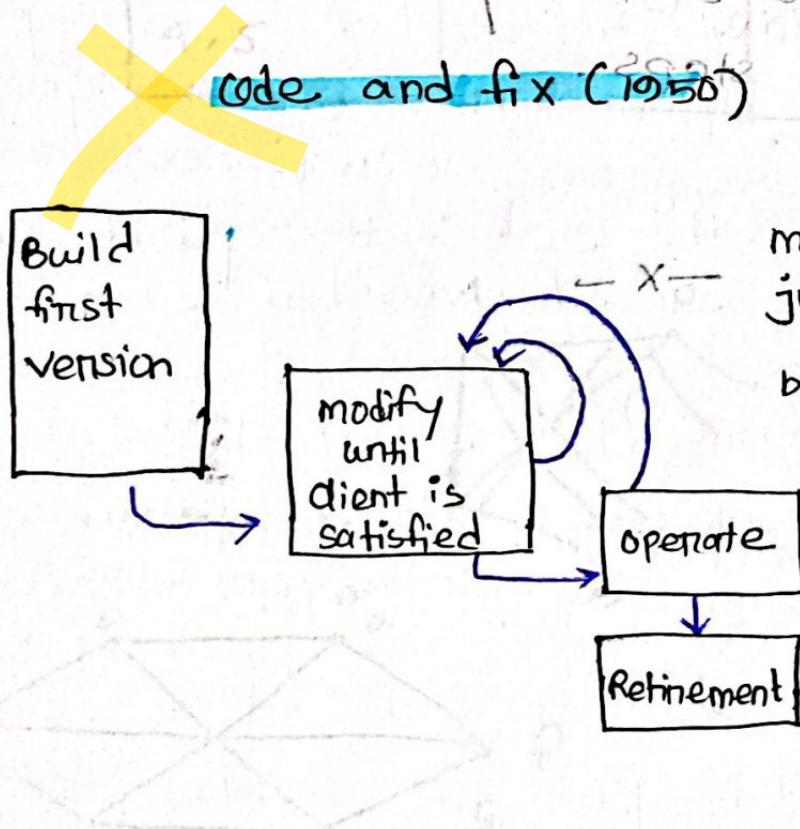
new program as above
removing errors of

Reusability :- The potential for the softwares

components to be used in other applications or contexts reducing development time and cost.

Testability :- The ease with which the software can be tested to ensure if meets its requirements and performs as expected.

Software Life cycle



The code and fix model is the concept of jumping right into and begin to design the system only to fix one any problem

(a) It is a cyclic process consisting of two major steps: coding and Fixing.

(c)

After some initial discussions teams starts the coding process

In this model there is usually no extensive planning concrete strategy and well defined design

planning stages

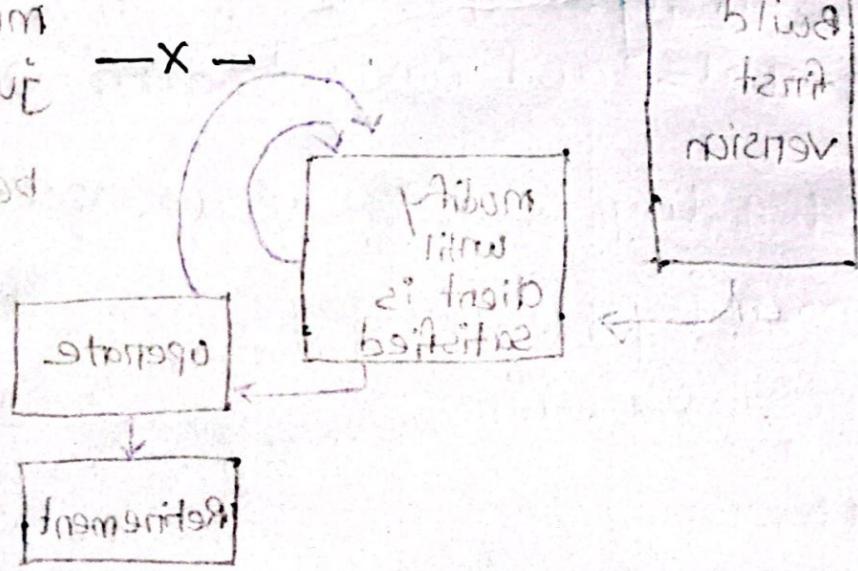
Disadvantages :-

✓ No team work

cannot handle large project
no planning

✓ cannot handle complexity

✓ No process steps



SOFTWARE ENGINEERING

27.08.24

2ND CLASS

Software life cycle:-

2) waterfall model (1968) :-

study and planning about project

Feasibility
Study

Requirement
Analysis and
specifications

customer requirements

design

coding and
unit testing

System Testing
and Integration

maintenance.

Advantage :-

- (1) Base Model
- (2) simple Model
- (3) small project
- (4) Clearly defined stages
- (5) well understood milestones
- (6) Easy to arrange tasks
- (7) Process and results are well documented
- (8) phases are processed and completed one at a time.

Disadvantages :-

- (1) No Feed back
- (2) NO Experiment
- (3) NO parallelism (Multi task)
- (4) High Risk
- (5) 60% work on maintenance. (cannot go back)
- (6) Not a good model for complex and Object oriented projects
- (7) Difficult to measure progress within stages
- (8) Cannot accommodate changing requirements

Incremental Model

Increment 1.

Feasibility Study

Requirement Analysis & Specification

Design

Software Engineering 3rd Class

Increment 2

মন্তব্য : Increment necessary

অর্থাৎ Increment হবে,

code and unit test

system test

Maintainence

60%

* ২০% Increment হবে তত Maintenance

Percentage increase হবে and errors

+ percentage decrease হবে।

* **Advantages of Incremental Model**

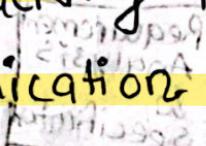
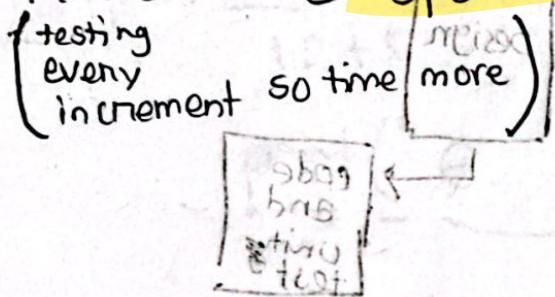
- (1) More customer interaction.
- (2) Reduce cost.
- (3) Flexible (error can detect easily and can solve easily)
- (4) Early release product demand
- (5) Errors are easily identifiable.

100% will be divided in Incremental method.

নির্মাণ → টেস্ট ← প্রযোগ মাস্টার

Disadvantages of Incremental Model

- (1) Good planning is required. (Good planning, increment and iteration and how many steps required and work load should be planned)
- (2) Increased complexity. (Step changes varies so it becomes complex)
- (3) Difficult in tracking progress (increment tracking)
- (4) Increased communication gap overhead.
- (5) More time spent on testing.



to communicate
team members
Goto increment
so communication
not good

Prototype

Requirement

Gathering information to decide what to do *
Requirements from customer *

Quick plan * . More customers (1)
Build prototype . less cost (2)
Customer Evaluation . less risk (3)
Release early (4)
Evolve over time (5)

Refine Needs

Customer Evaluation . less risk (5)
Evolve over time (6)

Design → Implement → Test → Maintain

Quick plan

Build Prototype

Refine Needs

Customer Evaluation

Design

Implement

Test

Maintain

Description

- * Dummy Model
- * Discuss w/ customer to ensure fit
- * Quick plan
- * Model Ready (and send to customer)
If all requirement
are okay
- * Customer check
- * If Not okay with customer then
it will go to refine.
- * If okay will go to Design ↗
- * Test for Bugs tackle
Implement এর সাথে এখন বলাবা ,
পরে Banaboo & Test করবো ,

w → with
w/o → without

→ live

* Rejected Prototype. is called Throwaway Model.

Advantages of Prototype :-

Disadvantages of Prototype :-

- (1) **costly** (নেক পুরো মাকে)
- (2) It has **poor documentation** because of continuously changing customer requirements (continuously customer change)
→ requirements এর পরিবর্তন
- (3) Customers sometimes demand the actual product to be delivered soon after seeing an early prototype (early delivery করার request)
- (4) There may increase the complexity of the system (complexity increase হবে)
- (5) Customer ask so, it has time complexity
→ for additional requirement
→ for additional idea.
- (6) Customer not clear with idea.
→ এই স্প্রোটোটাইপ কোর্টেস্যু আছে
→ to understand better
→ মেলে দেওয়া হবে
→ no written usage
→ কোর্টেস্যু দেওয়া

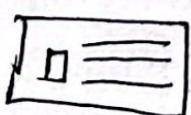
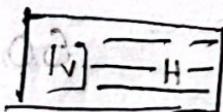
Types of Prototype

* Horizontal Prototype:-

- 1) A horizontal prototype displays the user interface before the product and gives a broader view of the entire system without concentrating on internal functions.
- 2) They give a broad view of the application including sample screens, menus, buttons and sample reports that reflect the current requirements.

* Vertical Prototype:-

- 1) A vertical prototype is a detailed elaboration of a specific function or a sub system of the product.



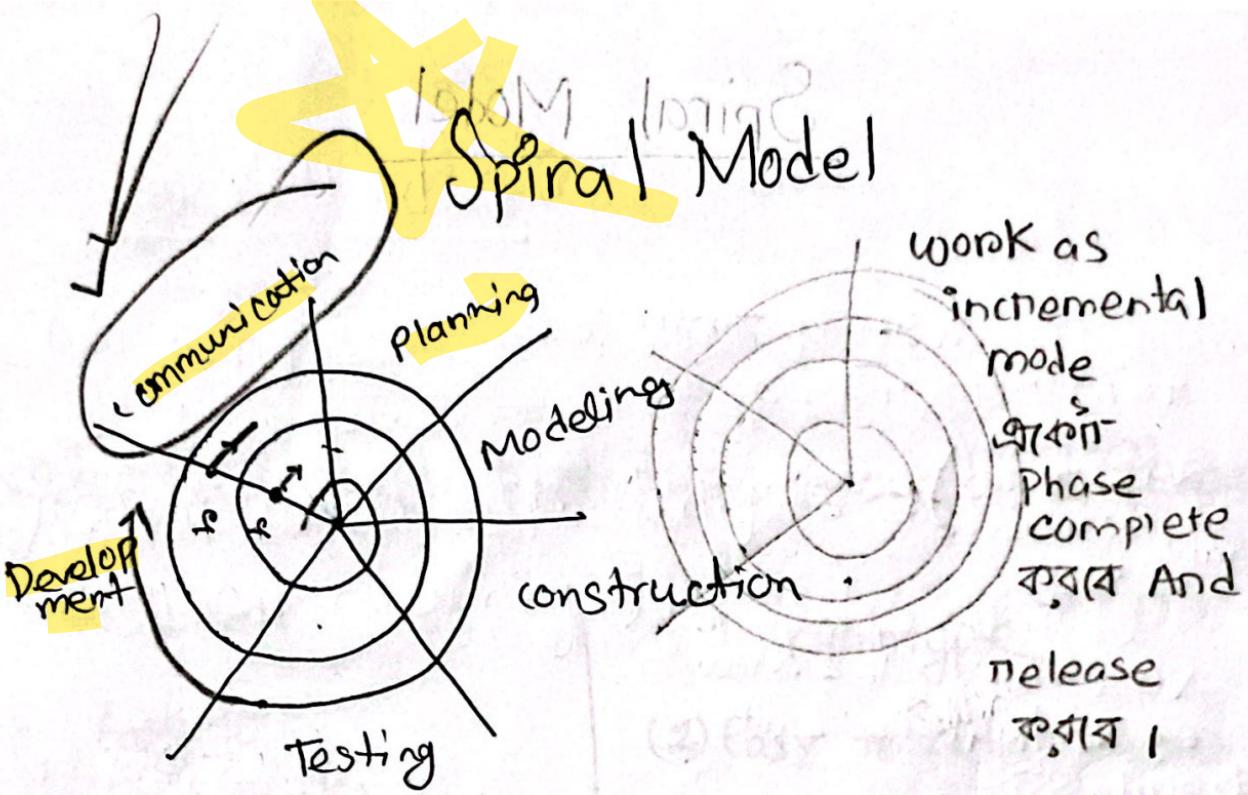
(2) Vertical prototypes are more technical in nature, connect to data base with real data, interface with existing sub-systems, and reflect the nearly exact function of key features.

(connect with database with real data)
(Vertical)

Key features : motor

Java

o - programming



Spiral Model is the combination of waterfall model and iterative model.

This model is a meta model.

- * Each phase in the spiral model begins with design goal and ends with client reviewing.
- * Software is developed in a series of incremental releases.

Waterfall + Iterative = Spiral model //

communication → Customer Requirements

Analysis.

- * Planning → Estimation
- Scheduling
- Risk Analysis.

communication → customer requirement (waterfall)

planning → scheduling and Risk analysis

Modeling → Design (was in waterfall)

construction :- code and unit testing

Testing → system testing

Maintenance

Deployment → Delivery, Feedback, Support.

Advantages of Spiral Model:-

- (1) Risk Handling → prepared for risk handling
- (2) Large projects → not for small
- (3) Flexible → errors identified easily
- (4) Customer satisfaction → version update and release after every phase.

Disadvantages:-

- (1) Complex → ~~phase~~ phase,,
~~phase for first report~~
- (2) Expensive → ~~phase for first report~~
- (3) To much risk Analysis → ~~risk~~ but no good planning
- (4) Time consuming → too much work so that is not every time report, risk is identified

Q:- Difference between spiral and waterfall / Incremental / Prototype

Q:- Difference between waterfall and prototype / Increment

Q:- Difference between -

- Increment and prototype.

A:

Risk handling

Large projects

Expensive

Cumulative

Small projects

Simple

3/9/24.

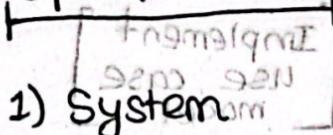
Lab

Software Engineering

Puja Miss Online 1st

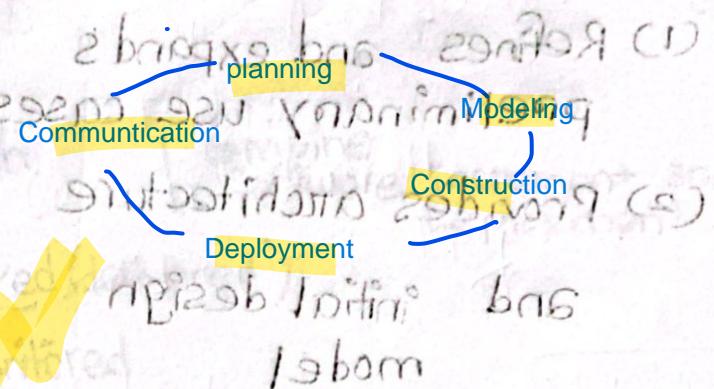
- 1st increment :- concept
2nd " :- system
3rd " :- system enhancement (Improved)
4th " :- system maintenance. (coding start)
2 planning and coding
3 system enhancement (deployed and maintain)

Spiral Model



1) System

Unified Process (1999)



Inception

} Communication → planning {

NO code
NO math
Max is document

(1) Vision the Document

customer KI KI
written} cache, cost, budget
Risk analysis.

This document is called SRS

Software Requirements Specification.

(2) Use case Model.

(Initially

Blueprint of project

(3) Project Plan



(behaviour)
(entities)
state) transmission on
information system
functionalities
business
processes
(activities)

Elaboration

combines Planning & Modelling

(1) Refines and expands preliminary use cases

(2) Provides architecture

and initial design model

Initial
Planning
[Implement
use case
model]

→
[Arch Design
Software
Architecture
and Iteration
Planning
and work flow]

Construction

(1) Builds (or acquires) software components according to architecture.

✓ Unit test :- Individual project
✓ Acceptance :- Whole project

Development

Transition.

[Deployment
and construction]
স্পন্দনা

- (1) Software given to end users for beta testing
- (2) Feedback reports defects and changes
- (3) Support information written

Deployment এ
ভুল থার্ক ফিx করাবে

construction
এ

transitional software given to customers (1)
(2) (3)

Production

[utিসিলিমেশন (E)
combine
software increment and
Deployment]

- (1) Software is deployed (delivered)
- (2) Problems are monitored

Re-Iteration
(Deployment and communication)

- (1) Feed back results in new iteration for new feature release (re-release) (feedback niye আবারু
ফিল করা হবে)

~~Advantage :-~~

~~disadvantage~~



- (1) Regular feedback to stakeholders
- (2) Reducing total project time
- (3) Enhancing task management

~~Disadvantages :-~~

~~disadvantage~~

- (1) challenge of using multiple development
- (2) cost
- (3) complexity & increases

(1) effective in developing (2)
(2) people are involved (3)

✓ Saturday : software Engineering
Upto Spiral Model

(1) fast development in new application for web

(2) iterative (iterative) (3) user involvement

Lab

Software engineering sessional

Types of Requirement.

Lab only have

(1) theory and diagram

Not know extra

(1) User requirement.

[User's requirement]

[User's requirements]

(2)

Subiect smit (3)

for

calculator large

project

extra ei hi (4)

(2) Feedback from

customers

requirement

hi application (5)

televison

phase is not possible

not solution (6)

design analysis

Subiect smit (3)

new

(4) All com

to give

stage

group

(5) No feedback

hi application (5)

final

hi solution is not right (7)

Difference between Waterfall Model and Prototype

Waterfall

- (1) Waterfall cost low [cost low]
- (2) Works well for small project [small project]
- (3) Time schedule very low [time schedule low]
- (4) Complexity of system is simple [complexity simple]
- (5) No feedback [No feedback]
- (6) Approach is linear [Approach linear]
- (7) High risk in waterfall High risk

Prototype

- (1) Prototype cost expensive [Expensive]
- (2) Works well for both small and large project [large project]
- (3) Time schedule long [time schedule long]
- (4) It is complex [Complex]
- (5) Feedback from customer [Feedback customer]
- (6) Approach is iterative [Iterative]
- (7) suitable for high-risk projects High risk

Waterfall and Incremental

Waterfall

- (1) Not flexible / Rigid. ✓
- (2) High Risk ✓
- (3) Early release not possible ✓
- (4) Can't handle large project. //
- (5) cost low ✓
- (6) Large team required. //
- (7) Overlapping of phases is not possible
- (8) Reusability is least possible

Incremental

- (1) Flexible ✓
- (2) Low amount of Risk ✓
- (3) Early release is possible ✓
- (4) Also can handle large project ✓
- (5) cost low ✓
- (6) Large team is not required //
- (7) Overlapping of phases = possible
- (8) Reusability is possible to some extent.

Difference between Waterfall and Spiral

Waterfall

Waterfall model is

- (1) Simple and easy
- (2) Low cost
- (3) Small project
- (4) High amount of risk
- (5) Flexibility to change is difficult
- (6) requires least maintenance
- (7) linear framework
- (8) Reusability is least possible

Spiral

Spiral model is

- (1) complex
- (2) Expensive
- (3) Large project
- (4) low amount of risk
- (5) Flexibility to change is not difficult
- (6) typical maintenance
- (7) iterative and framework
- (8) Reusability is possible

Differences between

Prototype

- (1) closed-ended prototype
- (2) Best ~~prototype~~ for those client who are not clear with idea.
- (3) client not clear with idea and is supposed to change
- (4) costly
- (5) improved quality does not increase cost
- (6) ensures greater level of customer satisfaction.

Spiral Model

- (1) Meta model
- (2) customer clear with idea/requirement
- (3) costly
- (4) Flexible
- (5) improved quality can increase cost of product.
- (6) customer satisfaction.

Increment

- (1) cost low
- (2) flexible
- (3) can't handle large project
- (4) Large team not required
- (5) overlapping of phases possible
- (6) complex
- (7) More time spent on testing
- (8) Returning to previous phase is possible

Spiral Model

- (1) cost expensive
- (2) not flexible
- (3) can handle large project
- (4) Large team is required
- (5) overlapping of phases not possible
- (6) complex
- (7) Time consuming
- (8) is possible also.

Prototype

- (1) Flexible
- (2) Easy to detect errors
- (3) ~~low cost~~ expensive
- (4) Suitable for high risk projects
- (5) Does not give emphasis on Risk Analysis
- (6) Time consuming

Incremental

- (1) Flexible
- (2) Errors are easily identified
- (3) low cost
- (4) can't handle large project
- (5) Give emphasis on Risk Analysis
- (6) Time consuming

Updated

Waterfall	Incremental	Prototype	Spiral
1) Low cost	1) Low cost	2) Expensive	1) Expensive
2) small project	2) small project	2) small and large project	2) Large project
3) High risk	3) Low Risk	3) suitable for high risk	3) low risk
4) Not flexible	4) Flexible	4) Flexible	4) Flexible
5) Linear Approach	5) Iterative Approach	5) Iterative Approach	5) Linear and Iterative Approach
6) No Feedback	6) Regular after each increment	6) Feedback from customer	6) User feedback frequent after every cycle
7) Overlapping of phase not possible	7) Overlapping of phase is possible	7) Overlapping of phase is possible.	7) Overlapping not possible
waterfall model simple	8) complex	8) complex	8) complex

Agile

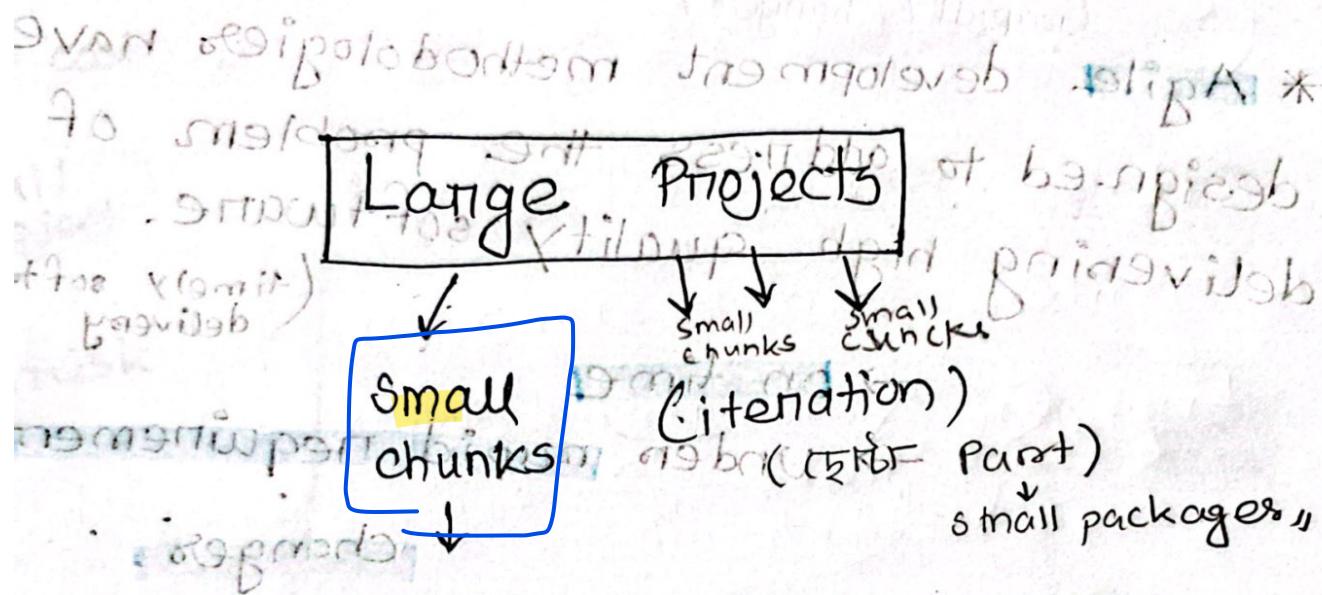
SLIPA

(rapidly changes)

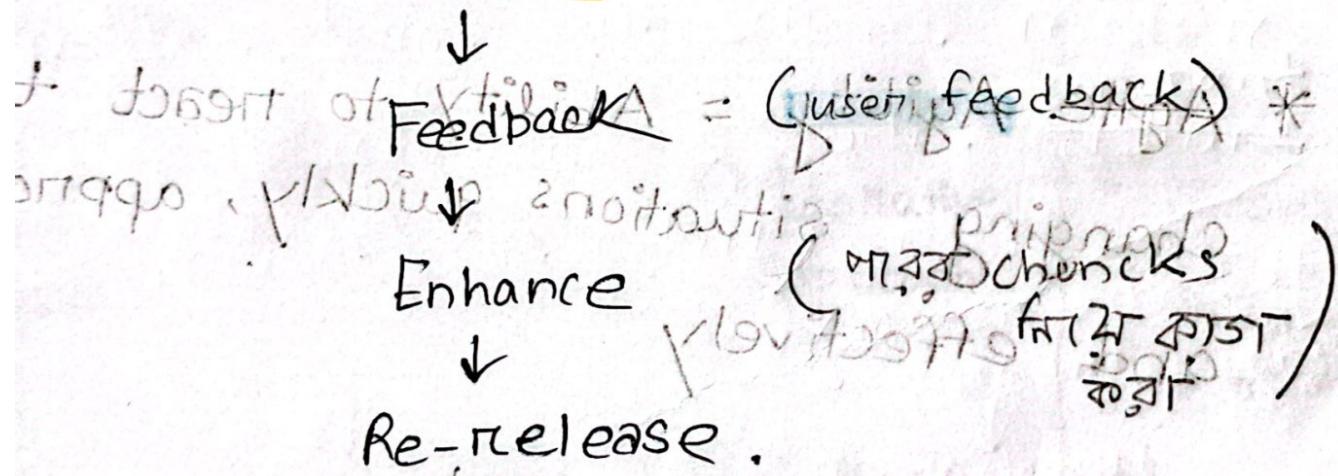
- * Agile development methodologies have been designed to address the problem of delivering high quality software.
 - (on time delivery)
 - (timely software delivery)
- Under rapid requirements changes.

- * Agile Agility = Ability to react to changing situations quickly, appropriately and effectively
- * Notice changes early
- * Initiate action promptly. (তাৰাতাৱি)

Agile



Release



Advantages of Agile

(1) Frequent Delivery ✓

(2) Face to Face Communication.

(3) Changes.

(4) Time (Large bc small parts use)

Time (Large bc small parts use)

Disadvantages of Agile

(1) Less Documentation (documents difficult)

(2) Maintenance problem.

Scrum

* This is a software development model of Agile (everyone work on a project and try to develop it)

* One of the most popular agile methodology

* Scrum is a light weight, iterative and incremental frame work

step by step,

less time maximum effort

* Scrum breaks down the development phases into stages or cycle called "Sprints" (Big project divided into small projects)

* The development time for each

Sprint is maximized and dedicated thereby managing only one sprint

at a time.

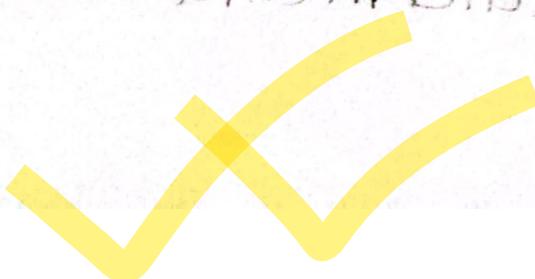
one sprint finish and review and then design in next sprint

* Scrum teams have Scrum master and product owner with constant communication on daily basis.

(master → check if work is done properly)
product
↓
customer
(requirement)
meeting
ব্যবসা

bottom tips reducing team effort to zero *

• Iterative, iterative, iterative is the motto
• Share Smart Information



10.9.24

Software Engineering ✓

✓ (Scrum)

~~Keywords~~

: Backlog, Sprouts

Daily scrum, Scrum Master,

product owner (Description in pdf)

Puja Miss 6th
class //

Backlog → (customers requirements lists)
prioritise list feature and

backlog: form where basic requirements are mentioned
a value is set, so, the value
more than we prioritise
that.

Sprints :- Big project divided into small projects.

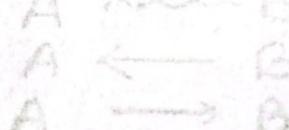
Daily scrum :- meeting about requirement with
client and developer is known as
daily scrum

daily meeting between developer and stakeholders

Scrum Masters :- No leaders in scrum but his work
is to observe (client & working / require-
ment / developer
works)

Product owner :- client

Feedback / etc



Scrum

Page No. 01

Advantages :- ^(minimum risk)
* ^{new thing} ^{Adapt} ^{flexible} ^{short time} ^{iterations} ^{customer satisfaction} ^{new technology} ^{new product}

- * Freedom and Adaption
- * High-quality, low risk product
- * Reduce the development time upto 40% (time reduce) sprint 2 weeks and only focus in next sprint

* Scrum customer satisfaction is very important

* Reviewing the current sprint before moving to new one. Review part

new features added to previous sprint
(Not going back)

Disadvantages :-

easy for small team

- * More efficient for small team size. Manager 2 person
- * No changes in the sprint. (No previous stage Sprint)

From Agile XP (Extreme program)

does not create burden on developer

* XP is a **lightweight**, efficient, low-risk, flexible, predictable, scientific and fun way to develop a software. (requirements)

* Small to medium sized teams that work under vague and rapidly changing requirements. (changing and rapidly changing)

* The five values of XP are communication, simplicity, feedback, courage and respect.

Communication → client and developer talk about requirements and analysis of those requirements

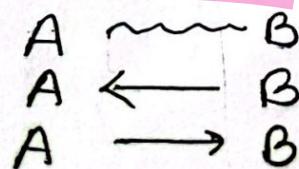
Simplicity → Keep it simple

[removed unnecessary requirements]

Above avoid complexity.

future addition not included.
only work with present.

Feedback / ↕



Feedback :-

customer feedback, unit test, throughout the project test, beta testers, **Budget -**

[feedback নিম্নোক্ত error margin
create কর্তৃ]

(Budget fix কর্তৃ) number of lines *

Courage - STRETCH

Team should have courage [project requirement should be analysis analyzed and should have courage]

Team should have discipline, feedback errors that are quite questioned by customer should be solved with courage.

Respect

Respect between customers and developers.

Also respect between team members.

$$\begin{array}{c} A \\ \rightsquigarrow \\ A \end{array}$$

ALL WORKS

14/9/24

7th class

Puja Miss
Software Engineering

Communication:- Communication with customer & Analysis of customer requirements.

Simplicity:- Avoid complex project try to keep project simple.

Feedback:- Feedback from customer. To have a good feedback team must be careful with budget. Feed back includes unit test.

Courage:- Must have courage about the project.

XP Processing -

- (1) XP ~~Process~~ Planning
- (2) XP Design
- (3) XP coding
- (4) XP Testing

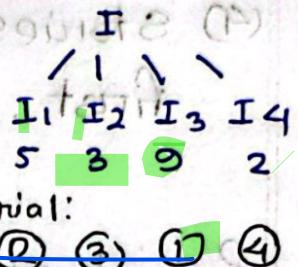
XP Model :-
At first planning customer communication and requirements provide by customer and should part the project into small parts (chunks) [part into stories & provide

Project Velocity :-

value in those stories

The number of customer stories implemented in the first release.

App → ← [1st release stories]



Design :- Basic Principle

serial priority :- by highest value.

KiS' → keep it simple

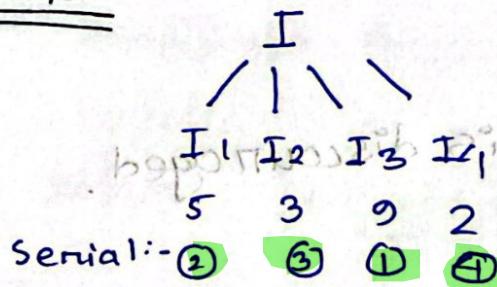
XP Planning :-

Here customer provides his requirements and then these are divided into ~~so many~~ small stories. These stories will set values. Highest value story will be implemented first.

cost and time will be planned by developer.

[Agile needs to move quickly]

Example,



- (1) XP team orders implementation by
- (2) All stories implemented immediately
- (3) Stories with highest priority implemented first
- (4) Stories with highest risk implemented first.

Project Velocity :- (Factor 1)

The Number of customer stories implemented in the first release.

(1) Helps in setting delivery schedule.

(2) Reduces over-commitment

XP Design :-

(1) → **KIS' (Keep it simple)**

(2) Design provides story implementation guidance

(3) Extra functionality is discouraged.

XP Coding and Testing :-

- (1) Does not follow **story** implementation and design
- (2) At first unit test will be done the coding will start

(3)

code by 2 person

one will code and one give feedback

Pair Programming :- (Factor 2)

- (1) In XP all code is built by 2 programming sitting side by side at the same machine.

continuous Integration :- (Factor 3)

- (1) Run all the unit test , on the integration machine
- (2) Tests must run at 100%
- (3) When a test fails , the problem must have been caused by the last change

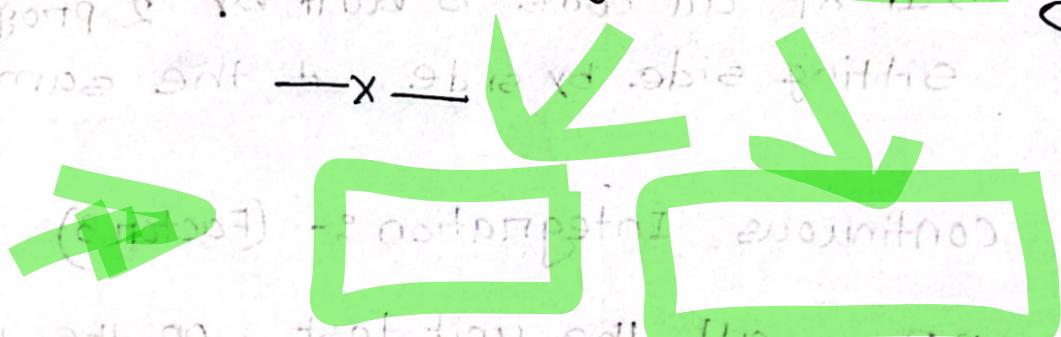
Refactoring :- (Factor 4)

improve internal structure will be developed

- (1) ~~Polish~~ Polish existing code to make it better
- (2) Improve /change design , architecture .

XP Testing :-

- (1) Implementation of unit tests should be such that they can be automated
- (2) Testing done upon modified code release
- (3) A continual indication of progress
- (4) Acceptance test are specified by customer to test featuring and functionality



Software Engineering

17/19/24

8th class

metaphor: comparison between own project and other project

Lecture 3

(Metaphor in English)

(1) ~~to~~ Collective Ownership :-

→ Every owner gets permit

→ 10 pairs use 10 methods

→ Individually each Malik तार,

→ Everyone responsible for unit test.
(you do it you fix it)

→ ~~40 hours/week~~

(2) 40 hours/week

→ (over time work is not precise)

→ Specific time is fixed (40 hours)

→ Personal life lives consequences
will fall on project.

(3) Onsite customers :-

→ ~~Feedback and real time feedback~~

→ Programmers don't know everything

Lecture 4

[Diagram PDF and Lab same]

Project Develop কর্তৃত যোগান্বহ Design করা হবে।
So, we need a structure.

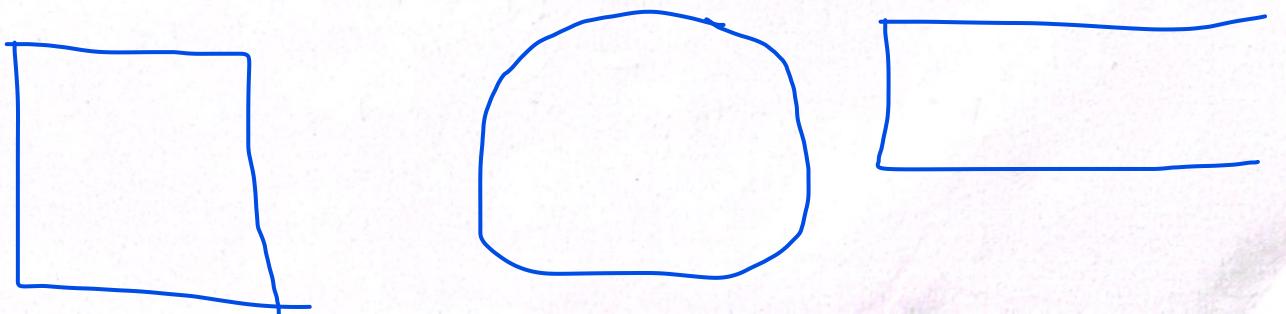
→ Data Flow Diagram is a diagram that shows how
data is processed.

Data এক জায়গা (A) থেকে অন্যেক জায়গা (B) move কর
Data flow বলে,

Defn :- A graphical tool useful for communication
communicating with users, management
managers and other personal communication.

Useful for analyzing existing as well as
proposed system.

Focus on the movement of data between
external entities and process and between
processes and data stores.
database.



Online Class 3rd class

SWE → Project Management

Lab Report Template

(1) title of the lab experiment

(2) Objective / AIM

(3) Procedure / Analysis, Design

(4) Implementation

(5) Test Result / Output

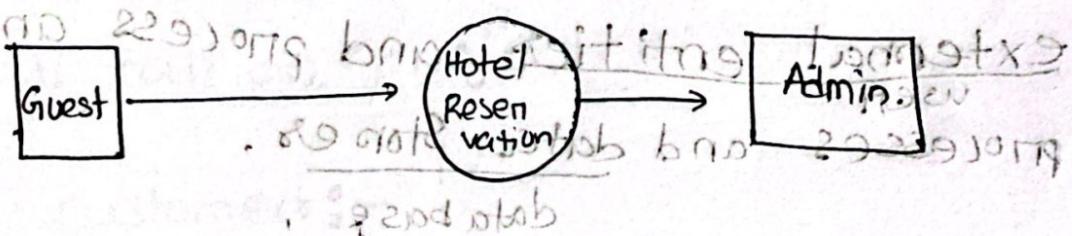
(6) Analysis and Discussion

Lucidchart.com

What is data flow? It is flow of information.

Maps out the flow of information for any

process or system.



External entity

Process

Data store

Data Flow

Two things to note about process and data store in DFD

and tips

(1)

Directly data store & process



between external entity and process

site name [Context level] full DFD upon lab report.

Homework -

Guest → description

WTREKOL MISS HAN,

(2) Form has missing data

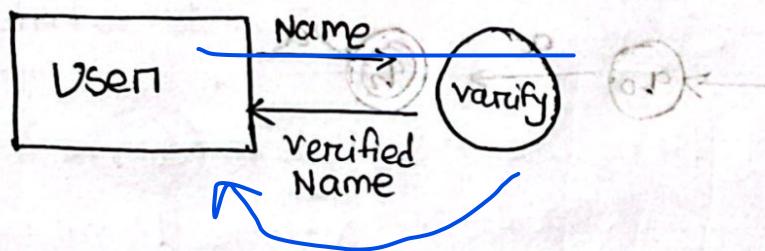
(2) Form has missing data

etc.

* KDFD elements :-

(1) External Entity :-

- A rectangle represents an external entity.
- They either supply or receive data
- They do not process data.

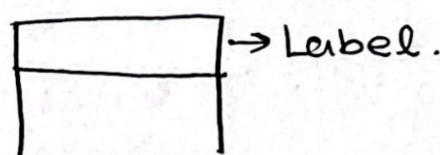


(2) Process :-

- A circle represents a process.
- Straight lines with incoming arrows are input data flows.
- Straight lines with outgoing arrows are output data flows.
- Labels are assigned to data flow.



or

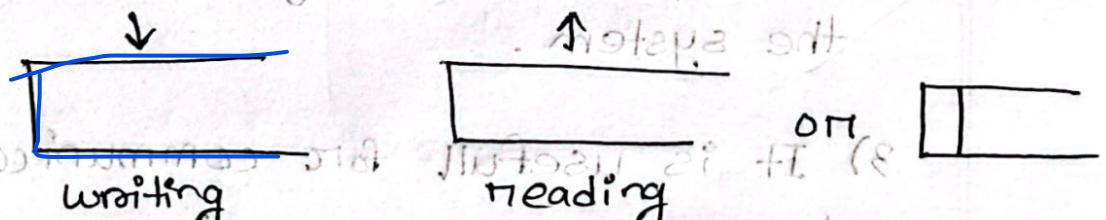


3. Data flow :-

- Marks movement through the system or pipeline to carry data.
- connects the process & external entities and data stores
- Generally unidirectional.

4. Data Store :-

- A data store is a ~~repo~~ repository of data
- Data can be written into the data store. This is depicted by a incoming arrows.
- Data can be read from a data store. This is depicted by an outgoing arrow A
- External entity can not read or write to the data store.



Rules of DFD :-

- Data can be flow :-
 - External entity to process
 - Process to external entity.
 - Process to store and back
 - Process to process
 - Data cannot flow from :-

1) External Entity to External Entity.

2) External entity to store.

3) store to external entity

4) store to store.

Advantages of DFD :-

1) A simple graphical technique which is easy to understand.

2) It helps in defining the boundaries of the system.

3) It is useful for communicating current system knowledge to the users.

4) It is used as the part of system documentation file.

5) It explains the logic behind the data flow within the system.

Errors of DFD :-

1) Forgetting to include a data flow or pointing an arrow in wrong direction.

2) Connecting data stores and external entities directly to each other.

3) Incorrectly labeling processes or data flow.

4) Including more than 9 processes or a data flow diagram.

5) Omitting data flow.

6) Creating unbalanced decomposition in child diagram.

Saturday

Software.
Engineering

28/9/24

10th class.

DFD

Logical DFD: Logical Data Flow diagram is an implementation independent view of system, focusing on the flow of data between process without any concern of storage location or people in the system. It is how it will be constructed mainly focuses on the business and how the business operates. (No implementation focus).

Physical DFD :- It is an implementation dependent view of the current system showing what tasks are carried out and how they are performed. (Implementation focus)

$\{ P \rightarrow S \}$

Lecture 5

ER (Entity Relationship) Diagram.

[Previously studied
Data flow diagram]

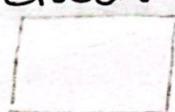
main काठी → entity এবং প্রযোজনীয়ে মধ্যে Relationship আছে
এবং sheta আলোচনা করা যাবে।



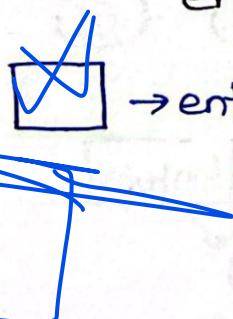
* ER Diagram's focus is on the entities and their relationships with the organizational system (physical body). two or more entity

রিলেশন নাম $\square \leftrightarrow \square$ entity এবং relation
* ER can express the overall logical structure of a database graphically. value এবং মাস্টি relation

* → A entity may be a person, A place, thing or an event.

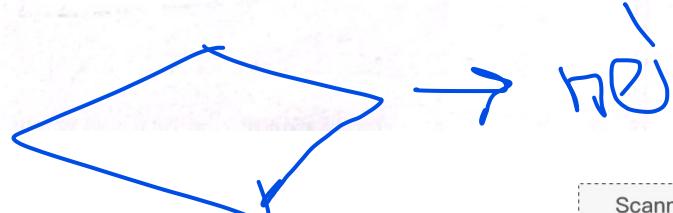


→ A relationship is the association that describes the interaction among the entities.

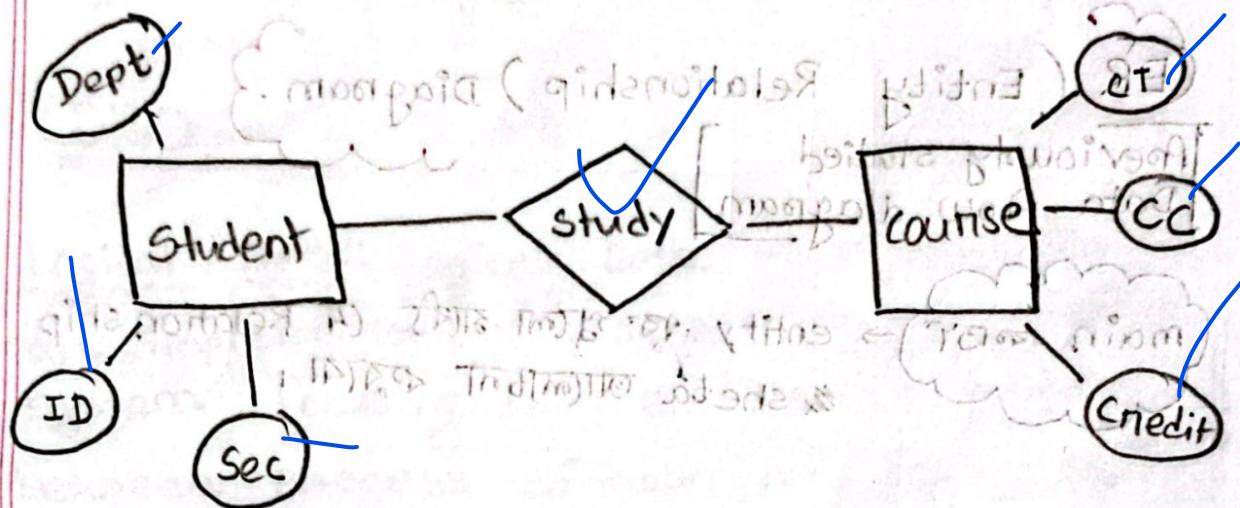


→ entity multiple entity থাকতে হবে,

◇ → relationship



ER Diagram.

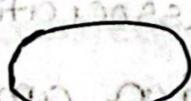


jive **Properties** from entities are known as attributes. it is called attributes.

* The properties from entities are known as attributes.



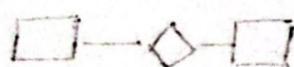
Entity no go birth



Attributes

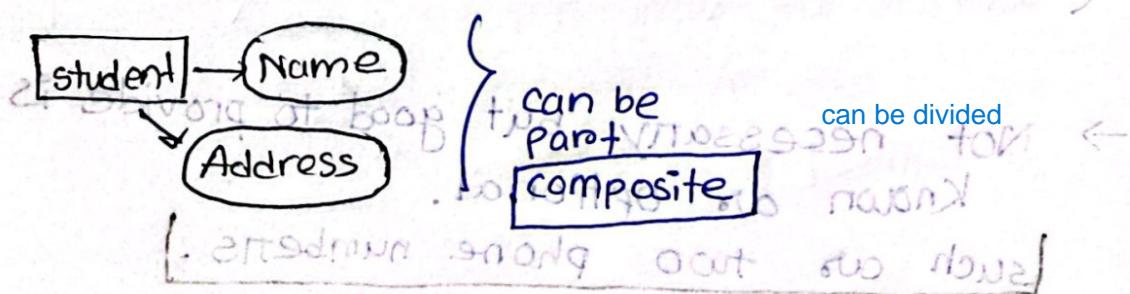
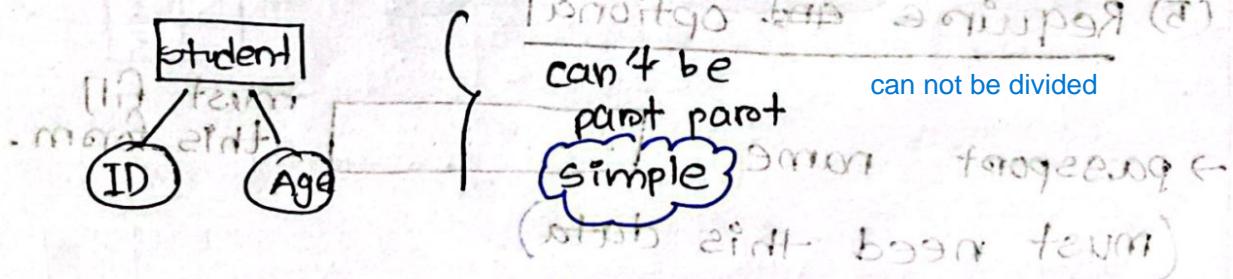
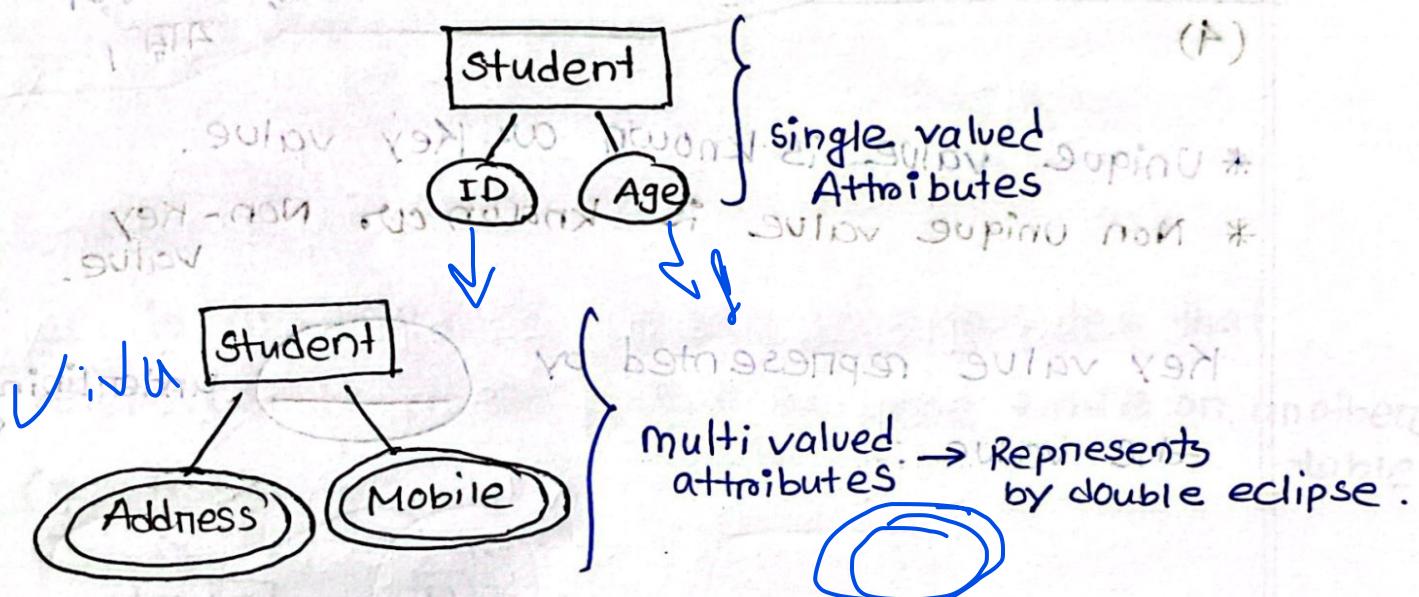
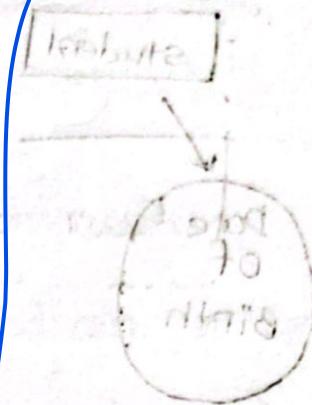


Relation

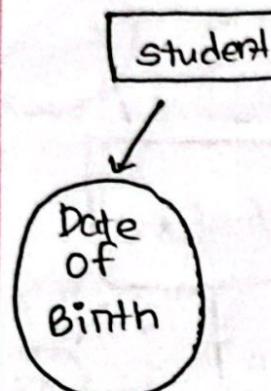


Types of Attributes :-

- (1) single vs multi valued
- (2) simple vs composite
- (3) stored vs derived
- (4) Key vs Non Key
- (5) Required vs optional



(3) Stored vs Derived



DOB	Date

Storage (1)

Composite (2)

Age can be calculated (3)

Key (4)

Derived (5)

Derived is represented by dotted ellipse

(4)

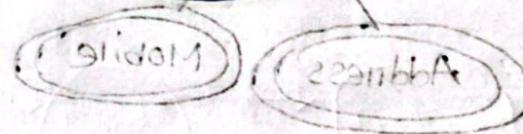
* Unique value is known as Key value.

* Non unique value is known as Non-Key value.

Key value represented by

the value.

Underlining



(5) Required vs optional

→ passport name (must need this data)

must fill this form.

(must need this data)

→ Not necessarily but good to provide is known as optional.

[such as two phone numbers.]

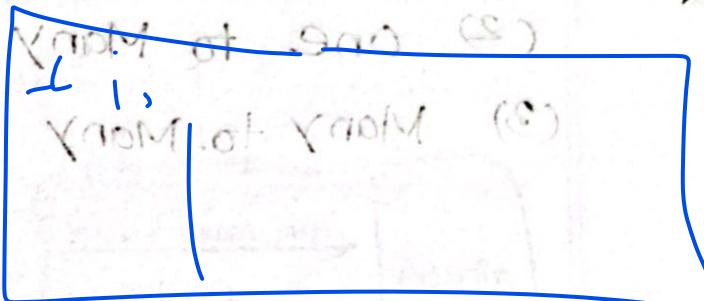
Primary Key and Foreign Key

Primary Key :- Unique and Not null

PK.

Name	ID	Mobile
A	2	012
B	1	013
C	3	016

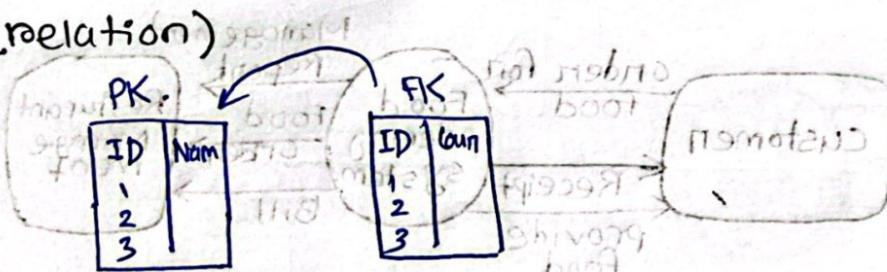
PK → can not be same and can not be empty



Foreign Key :-

It is an attribute or set of attributes that references to primary key of same table or another table

(relation)



Scanned with CamScanner

Relationship

Degree of Relationship /

Types of Relationship /

Cardinality of Relationship.

3 Types

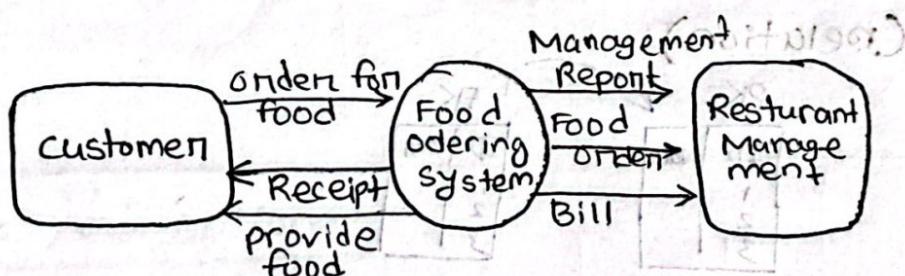
(1) One to One

(2) One to Many / Many to one.

(3) Many to Many

order	date	status
101	8/8/2023	A
102	10/8/2023	B
103	12/8/2023	C
104	14/8/2023	D

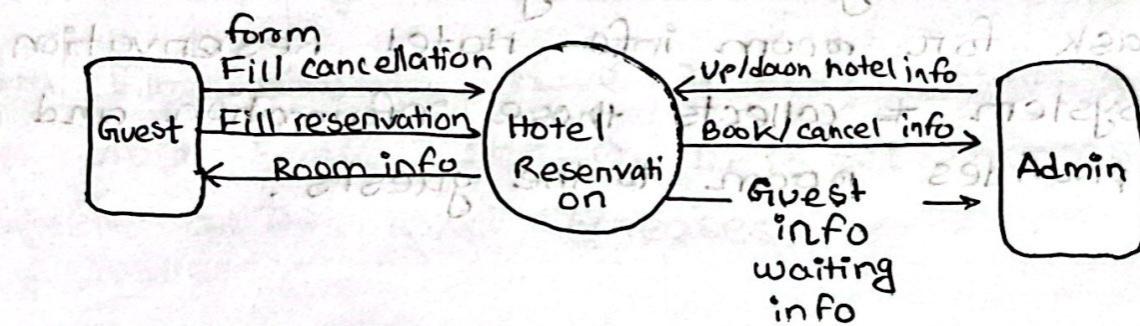
customer will order food . Food ordering system processes the order and provides receipt of the order . The restaurant management gets the report of Bill and Management and food order is passed to the chef and after processing it provides food to the customer .



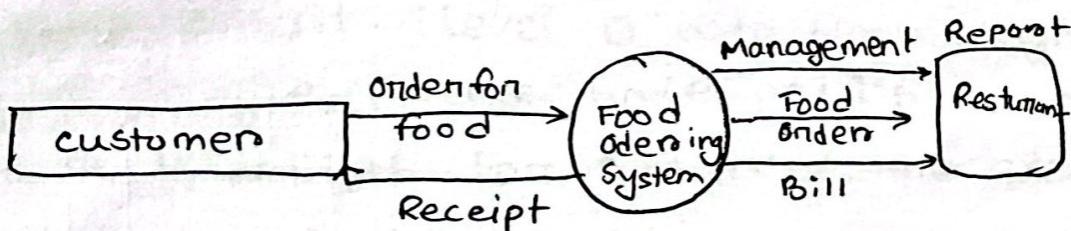
Context Free Diagram

Level 0

- DFD Level 0 is also known as Context Diagram. Basic overview of the whole system or process being analyzed or modeled. It is designed to be an at a glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.



- Food ordering system



Hence, customer ordered food through food ordering system. The system provided receipt by billing process from restaurant. The food order is passed to Restaurant. Restaurant then manages the process and provides food to the food.

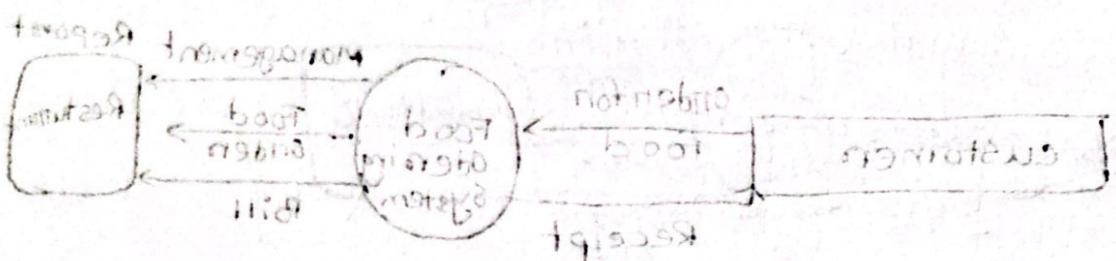
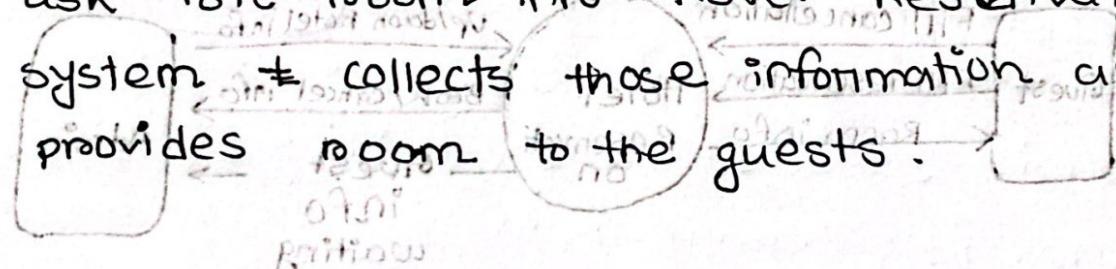
P.T.O //

• Hotel Management and Admin

O level

Here, Guest is the entity and Hotel management are Reservation is the process.

Guest provides information to reserve /cancel a room. This information is processed by Hotel Reservation system and checks with Admin for booking /cancelling confirmation. And ask for room info. Hotel Reservation system collects those information and provides room to the guests.



- No book about how hotel management interacts with guest.
- guest at front office management, stay
- or booking of room. front office most actively but according to experience most front office
- book is at

Software

Online class

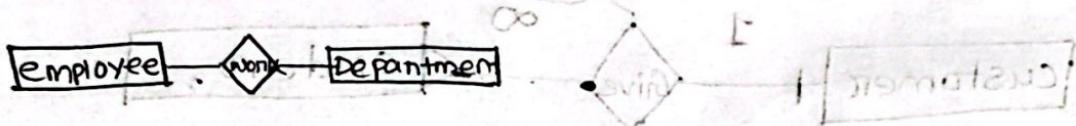
10/1/24

Relationships

3 types

- (1) One to one
- (2) One to many
- (3) many to many

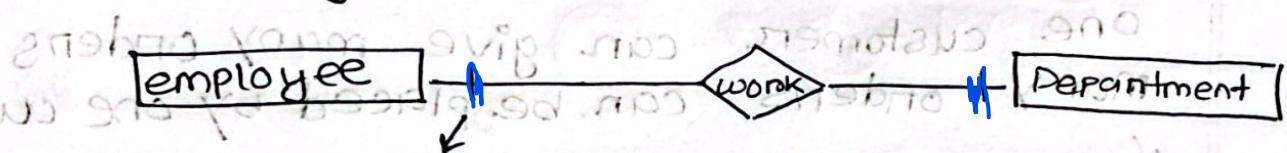
One to one :- one employee is assigned to one phone extension.



→ ekjon employee, ekta department अे रोप फुटा,

→ in one department only one employee works.

✓ How to understand there is one to one relation from a diagram?



through this small line.

→ ER diagram mainly focuses on database.



Table কাম করুণাৰ মাত্ৰ কোটি ফোন মাত্ৰ
so, we merge two tables

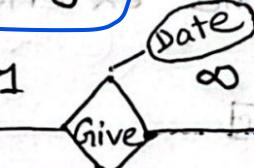
EID	Name	Age	DID
E1	A	10	D2
E2	B	20	D1
E3	C	21	D3

DID	Name	Loc	EID
D1	M	A	E2
D2	N	B	E1
D3	L	C	E3

One to many :-



1 $\quad \infty$



customer

orders.

Name	CID	Add
AB	C1	L
CD	C2	O
EF	C3	Sy

CID	OID	Date
C1	O1	
C2	O2	
C1	O3	
C2	O4	
C3	O5	

OID	Cost	Item Name
O1	60	P
O2	70	Q
O3	100	X
O4	100	Y
O5	120	Z

one customer can give many orders

many orders can be placed by one customer

Customer name have to be different

* * CID can not be primary key

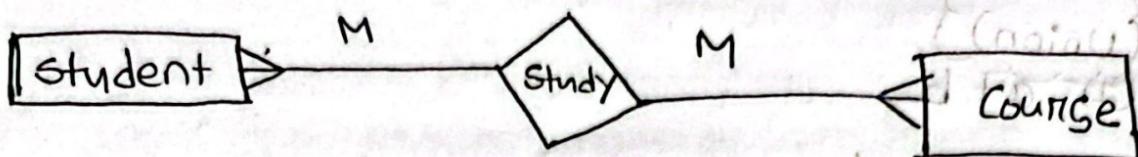
→ OID is primary key.

→ Merge order and OID.

OID	Cost	Item Name	CID
O1	60	P	C1
O2	70	Q	C2
O3	100	X	C1
O4	100	Y	C2
O5	120	Z	C3

Viva Syllabus UPTO ER Diagram.

Many to Many



✓ Many students can study many courses
✓ Many courses can be studied by many students.

SID	Name	Age
1	A	16
2	B	20
3	A	21
4	C	21

ID	CD
1	C1
2	C1
3	C3
4	C4
5	C3
6	C2

C.O	C.T	Credit
C1	MD	2
C2	CAD	3
C3	SE	3
C4	TOL	2

* We can not reduce the table on many to many because there is no primary key.

ER Diagram sign

One —————+

many —————→

one (one and only one)

++ (double sign)

zero or one

0+ (একজন না থাকলেও অন্যজন থাকতে পারে)

one or many

—→ (একজন বা একজন থাকলেও অন্যজন থাকতে পারে)

zero or many

—→ (একজন বা একজন থাকলেও অন্যজন থাকতে পারে)