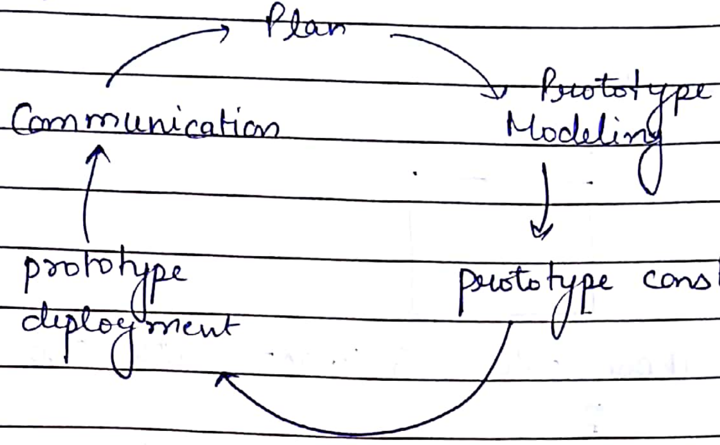
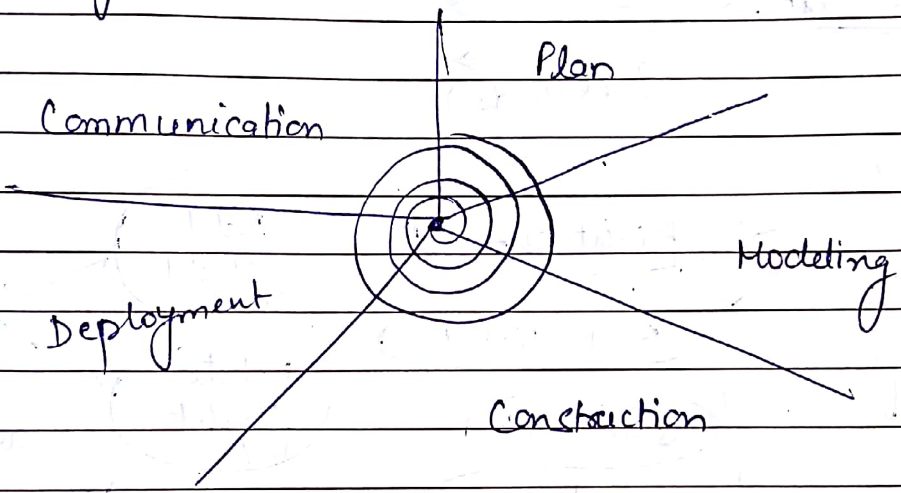
classfellow
DATE : 21/8 2018
PAGE No

Models

Prototype Model :-



Spiral Model → combination of waterfall model and prototype model



Move in the form of spiral.
Have detailed info how to make model.
∴ reduce overhead.
Consist of risk manager at every stage.
Used in high projects
costly

Imp #

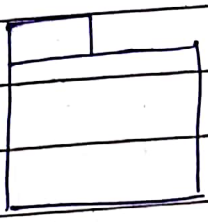
Agile S/w

→ dynamic
→ content specific.
→ change embracing → (change can be made at any stage)

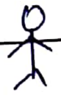
→ growth oriented

→ Use case diagram :-

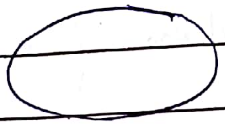
System



Actors → those which interact with s/w

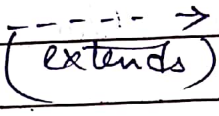


Use cases →

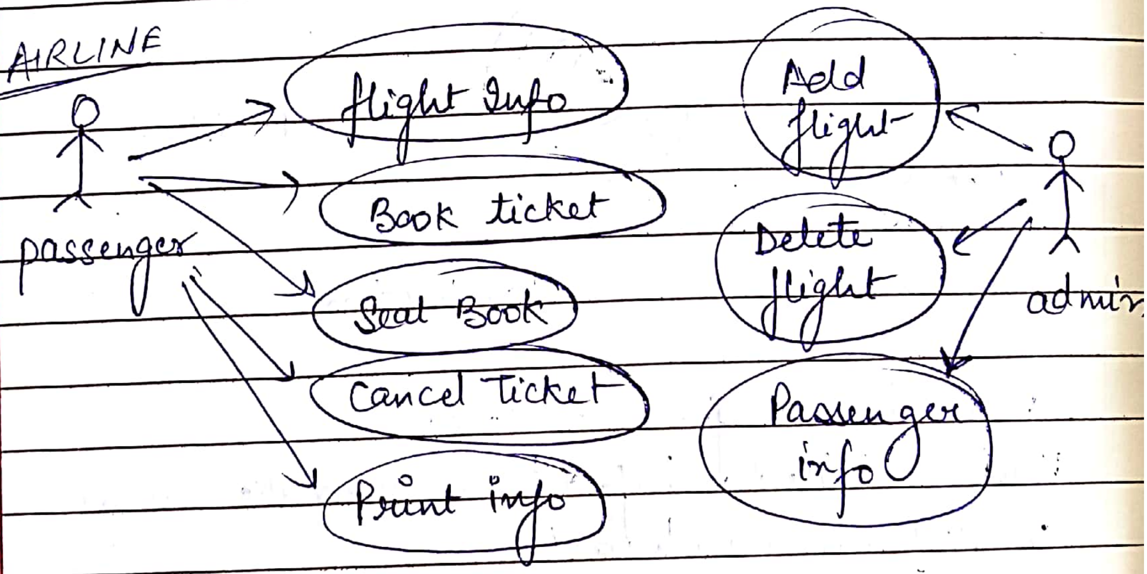


(work done)

Relation ship →



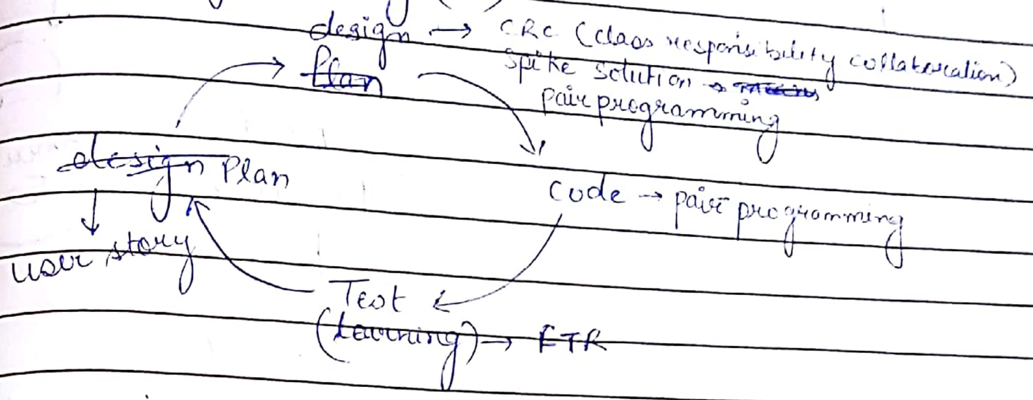
eg.



Q1. Clg info system.
Q2 Library management system.
Q3 Hospital management "
Q4 Online shoping
Q5. Banking

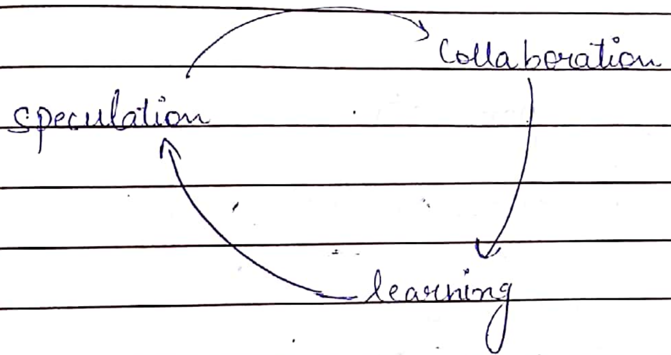
Theory

Extreme Programming (XP)



Refactoring → modify internal structure.

Adaptive s/w development (ASD)



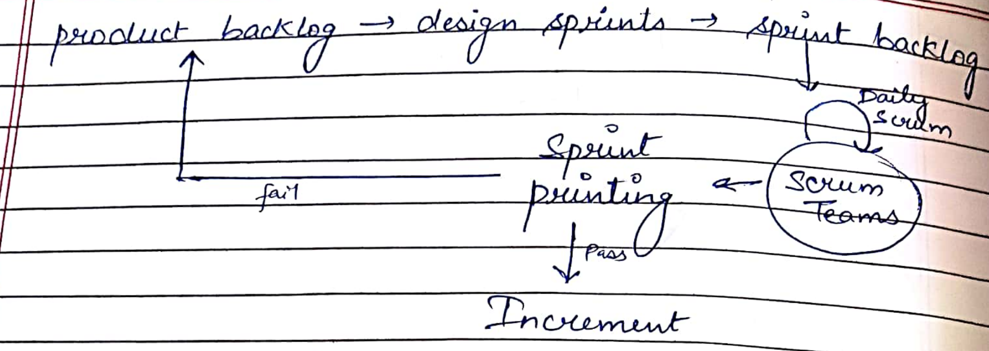
In Speculation we include 1. adaptive cycle planning,
(main motive behind our s/w) 2. Mission statement,
(clients requirements) 3. requirements.
4 Project constraints.
deadline) 5. time based release plan.

In collaboration we include 1. FTR
2. Focus Groups
3. Postmortems

SCRUM (strategy used in agile.)

↓

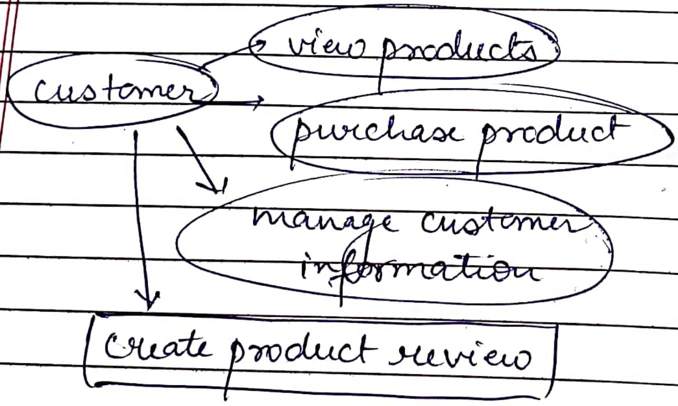
Sprints (work performed

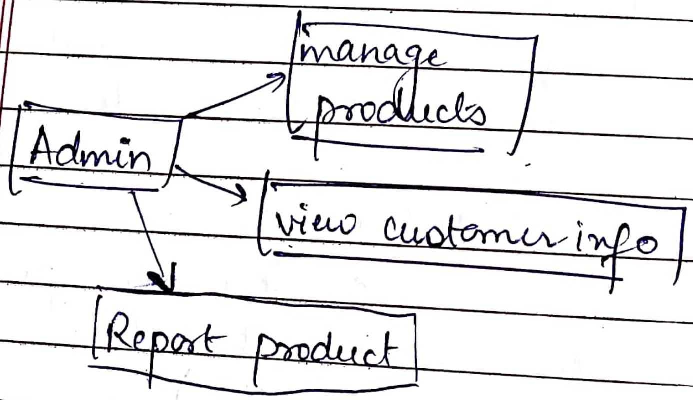


Scrum created by Jeff Sutherland in 1990's

———— × ———— × ————

Lab





class Diag

class Nam
attribut
related

e.g. studen
+id():
- name
view()
attenda

generali
aggrega
diff b/w

fellow
20

…nt backlog

Daily scrum

…rum
Teams

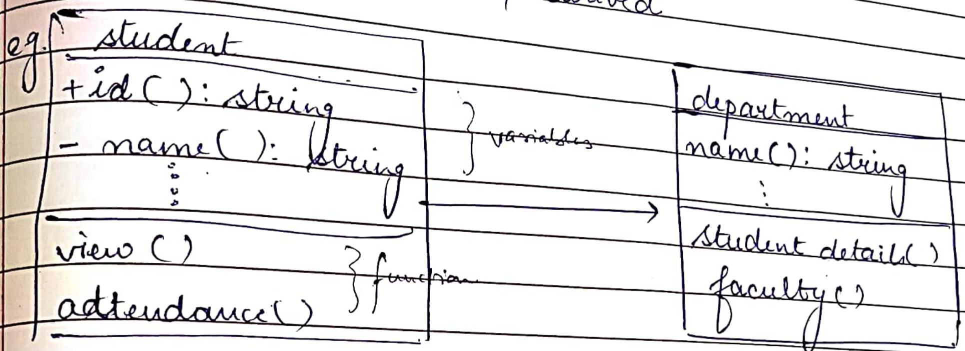
o'/

class Diagram

class Name
attributes
related functions

+ public
– private
# protected
/ derived

e.g.



generalisation → relationship b/w classes. —▷

aggregation → relationship b/w objects →

diff b/w association and aggregation.

↓ ↓

⟶ ~~◆~~ —◇

Req Elicitation :-

→ Interviews

→ Brainstroming sessions

→ FAST

SRS :-

→ functionality

→ external interface.

→ Performance

→ Attributes

→ Design constraints.

Characteristics of SRS → correct

→ unambigious

→ complete

→ consistent

→ verifiable

→ modifiable

→ Traceable

⇒ Data flow

○ function

= file / db

▭ i/p / o/p

↷ flow.

Bubble chart

designing tool used in top-down approach to system design.

→ ER diagram → logical representation.

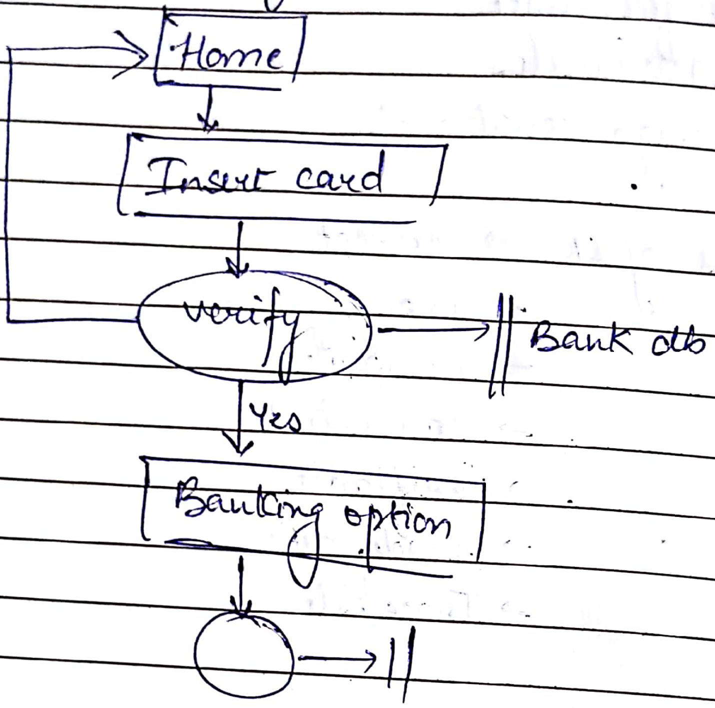
3 basic elements → entity are the 'things' for which we want to store info. An entity is a person,

place, thing or event.

→ attributes are the data we want to collect for an entity

→ Relationship describes the relations b/w the entities

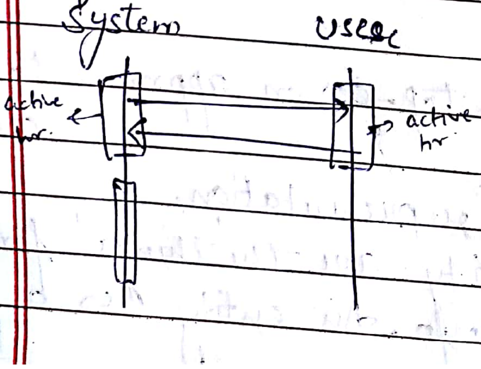
Data flow Diagram (DFD)



Interactive Diagrams:-

Sequence   collaboration

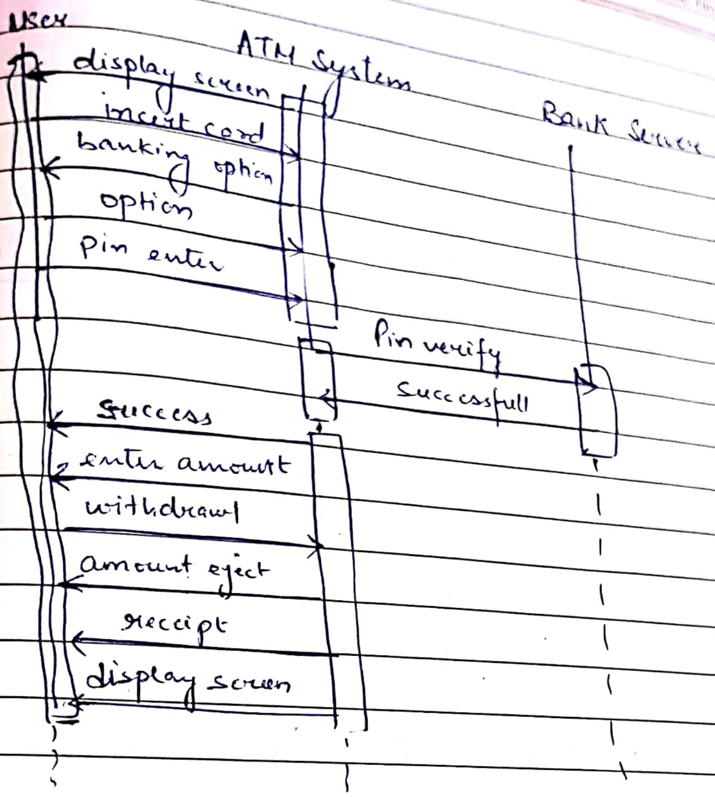
→ Sequence



DATE: / / 20
PAGE No

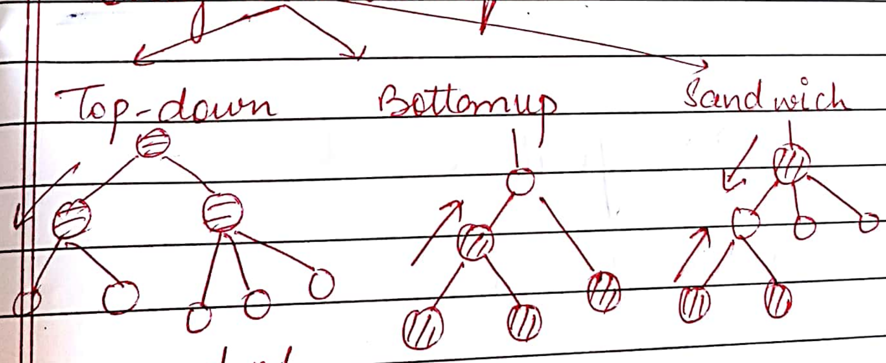
o collect for an
o b/w the

ATM Machine



THEORY

Integration Technique



1. no need of driver

(1.) no ~~de~~ need of stubs

Q. Sum of 2 nos. find error, fault and failure in this code.

```
{ a=10;
  b=20;
  c=a-b;
  printf("sum of nos %d",c);
}
```

we will not get correct output. so whole module will be wasted. so this is failure.

error → coding mistake. c=a-b

fault → fault comes in line c=a-b due to which whole module ~~do~~ fails.

error

→ Verification
→ Validation

Q what is verification and validation?

~~while developing~~

Verification means har module/face ke baad check karenge module sahi chal raha hai ya nahi

Validation → complete ~~model~~ s/w banne ke baad check karenge ki vo requirement ke accn sahi bana hai ya nahi.

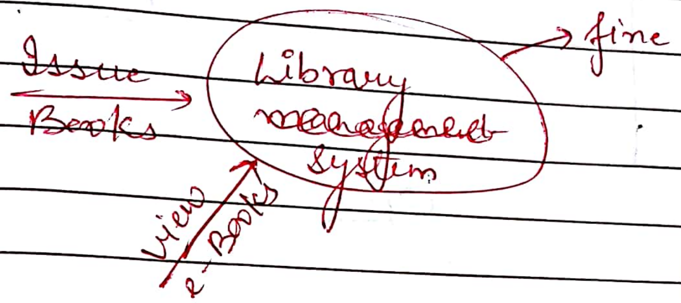
Unit
Integration → Verification
System
Acceptance → validation.

DFD

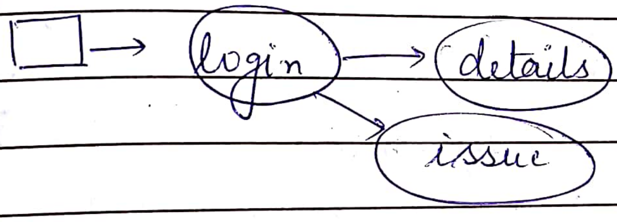
Level 0 → context diagram.
Level 1 → DFD

Level0 → ek cir hi pura system show karega

eg.



Level1


login
details
issue