

4P
CPM → Critical Path Method → draw → Forward
BPM → Backward.

PERT → মাত্রিক।

* * * * * Flow chart → Cyclomatic Complexity

Outmod-
nation

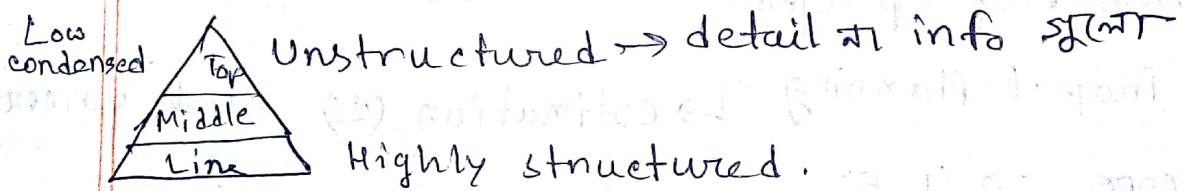
Notation X

SEIS

① Info for management:

Data and info :

Types of info : managerial view → 4BT.



Need for info system

Management structure. → concepts.

21, 22 slide

② System Analysis & design goals:

- Life cycle of sys. analysis & its step (details)
- diagram (নিরূপ নথি)
- Roles of sys. analyst
- Attribute of a sys.
- Tools used by a sys. → DFD, UML, testing tools

③ Feasibility Analysis :-

ফিল্ডে "

- " কার্য পদ্ধতি →
 - a) Technical
 - b) Operational → chaina
fb allow না
 - c) Economic → ব্যবস্থা আপনার

Steps in Feasibility Analysis

Guideline for searching goal

Hostel study → case study না প্রযোজন তৈরী

Normal approach

Present value " } CBA .

↳ r → interest → 11ak rsf দিয়ে একে পর্যবেক্ষণ করা
 রাখতাম তথ্য উপর ভিত্তির present value
 হবে মত। (কর হওয়া present value)

④ Info. Gathering :-

Info Gathering Strategy

" " Sources

" " methods

Planning an interview:- CEO ওর মাঝে int meeting .

" technique :

Questionnaires : কিন্তু info দরকার data survey এবং অন্য-পথে

SRS এর মতো : কিন্তু Ques . এর সমস্ত টোপি করে আনতে হবে।
 DFD " symbol : O □

⑤ DFD

DFD কি ? কের লাগে ? কেমনভাবে উৎ করি ? কোন করি ?

process O entity □

↓
System

মানুষ গতি → student

→ vendor

→ mess manager

DFD draw

Extended dfd

entity (কোন
গুলো
ইবাব নেয়া
হবে না) ,

Software Engineering & Information System

Software Engg as per syllabus
Info System design 2013-14

System Analysis and Design © V. Raja Ramamani

Project Development

RfI : Request for Information

RFP : - - - Proposal

↓
Client IT

Data : 441ABCD 3.89

| Age | Name | CbPA |
|-----|------|------|
| | | |

Information for Management

types of information : Operational : day today दैनिक जीवन
from
tactical : व्यापक/समय/उपकरण
strategic : अधिकारीका फॉर्म वित्तीय संस्था
Statutory : law apply करना/उत्तमा फॉर्म/
government का नियम लागू

TO

07.08.18

656-4103

Software Engg

Role : → Project Manager

→ Team Leader

→ Senior Developer

→ Junior Developer

→ Project Management

→ Developers (Back End) (C++, Java, Database)

→ Developers (Front End)

→ Testing / Quality Assurance

Team
Leader

Bioinformatics

Project Proposal

CSIR Project

Project Title :

Objectives : point wireless

Teacher :

Team Member : 1. P.M :

2. TL :

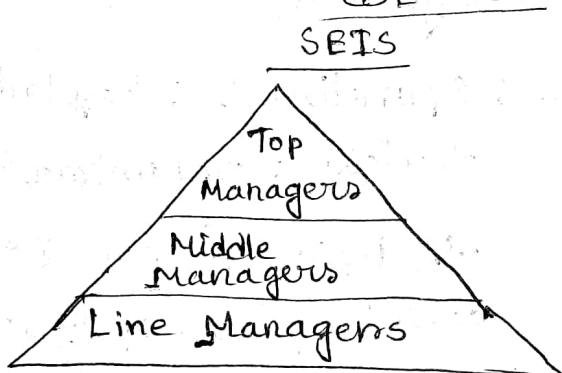
3. Developer :

To : hasan07cse@gmail.com

Sub : Software Project Proposal

body : name, id (team member).

09.05.18



Business data processing :-

Relational database (2D) \rightarrow $3D$ \rightarrow $2A$

OLTP : Online Transaction Processing.

OLAP : " Analytical "

Batch :

Business Intelligence :

Decision Support System :-

=o=

1. AI & Robotics :- Machine Learning
Prolog

2. Big Data Analytics : $VVV \rightarrow$ vast
 \rightarrow variety
 \rightarrow velocity

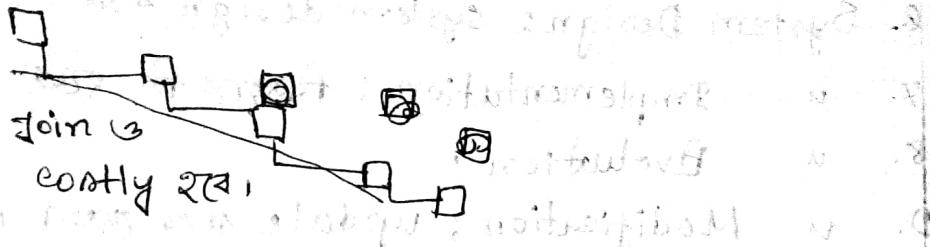
unstructured \rightarrow video, audio, file, image, etc.

Relational DB \rightarrow structured data. e.g. \rightarrow

| | |
|------|------|
| name | CGPA |
|------|------|

ETL framework.

Big Data:



Data Analytics

Warehouse

Business Intelligence

Data Mining

" Science

Machine Learning

Big Data & Machine Learning

SEIS

System Analysis & Design

Types of Information: 4 types

- 1. Operational
- 2. Tactical
- 3. Strategic
- 4. Statutory

Bird's eye view:

Steps involved in Analysis and Design:-

1. Requirements Determinations: client for req / priority
high priority / all possible combination of requirement.

2. " Specifications: after requirement details

3. Feasibility Analysis: req. to be feasible / correct,

4. Final Specifications :- final specification স্থার করানো।
5. Hardware Study : Hardware কি- পরিষেবা এবং নিয়ে analysis/
কোর্টেজ করা হবে / আর idea প্রস্তুত হবে।
6. System Design : system design করা। (building করা এবং বাইবল)
7. u Implementation : ফিল্ডে ফি হবে এবং দেখা।
8. u Evaluation :
9. u Modification : update করে দেখা।

Motivation :

- i) Designing info. system for an org is a very complex job.
- ii) Students should know how to logically divide a complex job into smaller manageable steps.
- iii) Each step must have a logical beg and end and must be self-contained.

3. Feasibility : Human Resource - কর্মচারী সম্পর্ক নথিব।
Time and Money - কোর্ট time
Equipment needed - equipment কি দরকার।

8. System Design : object \rightarrow independently represent করা,
identify এবং প্রতিকর্তব্য করা।

$= 0 =$

24.05.18

Definition of Software Process :-

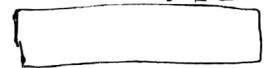
A framework for the activities, actions and tasks that are required to build high-quality software.

Software Process :-

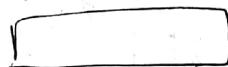
Process framework :-

Umbrella activities

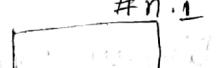
framework activity #1
#1.1



#1.k



framework activity #n
#n.1



#n.m.



Activity

↓

action

↓

task

API : Application Programming Interface

int function(a,b);

int u (int a, int b)

{

 return

3

stakeholder

→ वार प्रोजेक्ट

कर्मचारी / संगठन

इस्तेहास / इस्तेहास

प्रयोगशाला

Education Ministry

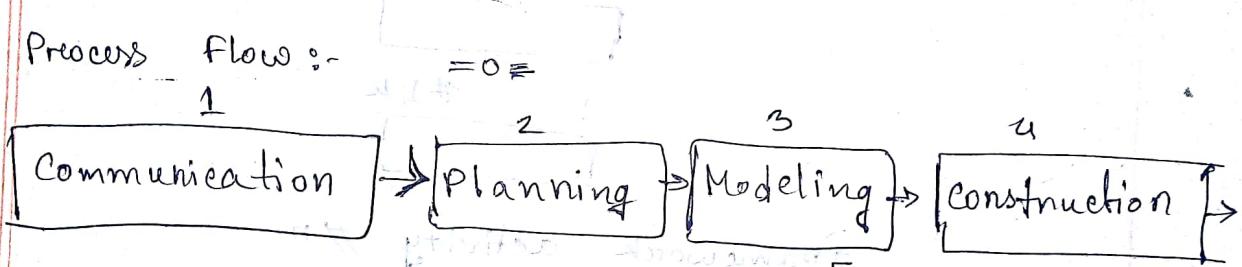
Full database

Final result (Board, Roll, Reg, Year)

BAUST

Find Result (PIN, Roll, Reg, Year)

Process flow :-



a) Linear process flow

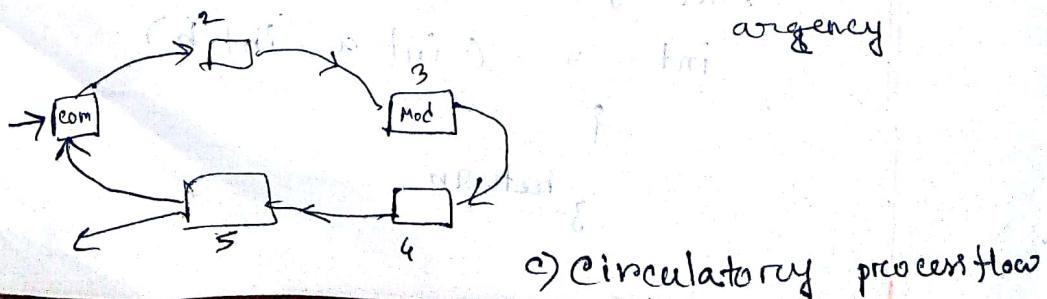


confidence level
high

time cost
low



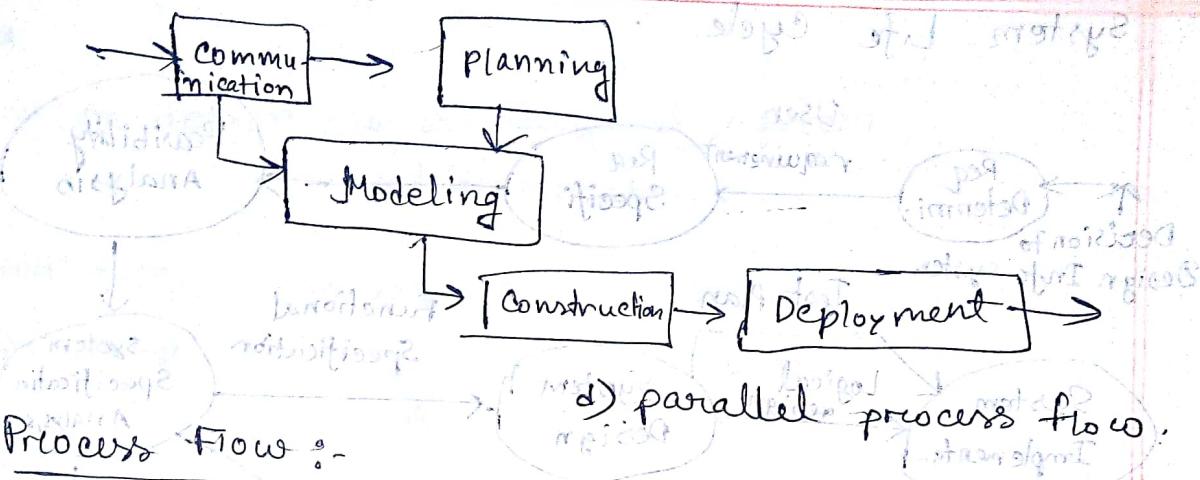
b) Iterative process flow → time cost



urgency

c) Circulatory process flow

(contd.)
 binary utilities #
 DATA & BAMS #
 (1705)



Process Flow :-

A task set defines the actual work to be done to accomplish the objectives of a software engg. action.

The water fall Model :-

Step 1

Requirement Gathering :-

→ Requirement list

→ Feasibility Analysis ..

① Login :

description

① req

③

② req

④ req

develop
2 Dev 4 days

Project Report

1. primary - secondary - tertiary - quaternary - primary

2. primary - secondary - tertiary - quaternary - primary

3. primary - secondary - tertiary - quaternary - primary

4. primary - secondary - tertiary - quaternary - primary

5. primary - secondary - tertiary - quaternary - primary

6. primary - secondary - tertiary - quaternary - primary

16.05.18.

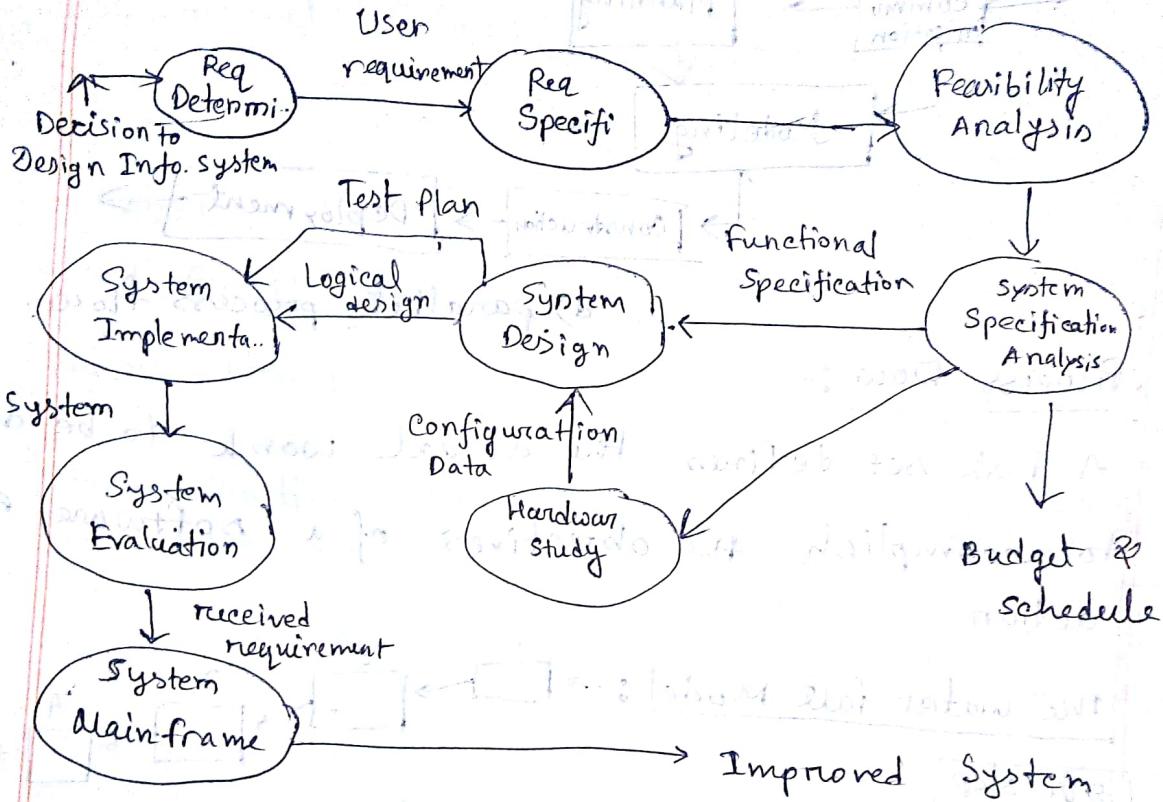
SE IS

(62135)

Baby city period

Sunset → নামায
দেয়া)

System Life Cycle



Log file → evaluation log এবং enables করে।

transaction এর স্বত্ত্ব প্রক্রিয়া,

a lock করলে system slow হবে।

CDR → Call Details Recorded.

Role of System Analyst

- Defining Requirements - Involves Interviewing Users.
- Prioritizing " - Observe obtain Users Consensus.
- Fact Gathering - Lower level এর কৃষি গবাবে।
- Analysis and evaluation -

• Solving problems - ~~Identify problem, analyze, to suggest~~

• ~~Identify problem, propose & recommend~~

System design - Logical design of system.

- Object identification

Attributes of a System Analyst :

Knowledge of org : jargon \rightarrow verbiage

Knowledge of Computer and Software :

Good interpersonal relations :-

Ability to communicate :-

Analytical mind :

Tools used by Systems Analyst : 20.05.18

CMM = Capable Maturity Model \rightarrow Integration.

\rightarrow CMMI = 1 u u

CMMI = 5

CMMI = ~~International standard~~ \rightarrow ~~ISO 9001~~ \rightarrow ~~Quality management standard~~ \rightarrow ~~ISO 9001~~ \rightarrow ~~Quality management standard~~

(ISO, CMMI)

Assignment - 01

Feasibility Analysis

Technical :

Operation :

Economic :

Cost, benefit Analysis (CBA) :

~~we
work
don't
know~~

Results of the Info. gathering phase:

consensus → consensus report → system specification

SRS → System Requirement Specification

Guidelines for searching goals:-

Characteristics of a goal :-

- ✓ ① Proposal
- ✓ ② Requirement Determination & Feasibility Analysis
- ③ Information gathering / collection
- ④ Stakeholders
 - ↳ strategy
 - ↳ list of the collected info

GEIS

Stakeholder analysis is the process of identifying, characterizing, and analyzing the people or groups who have an interest in or affect the success of a project. Stakeholders can be internal or external to the organization. Internal stakeholders include employees, management, and customers. External stakeholders include suppliers, partners, and regulatory agencies.

textile → পোর্ট → কর্মসূলি এবং
Garments → কর্মসূলি design এবং

Point —
Font Size - 10
★

Information Gathering / collection :- and DFD

↳ Info Gathering Strategy

From each requirement → list of collected info...
Model use our own language

Identify Information sources..

DATA System or Report BTW → overview summary chart,

Existing computer programs ↗

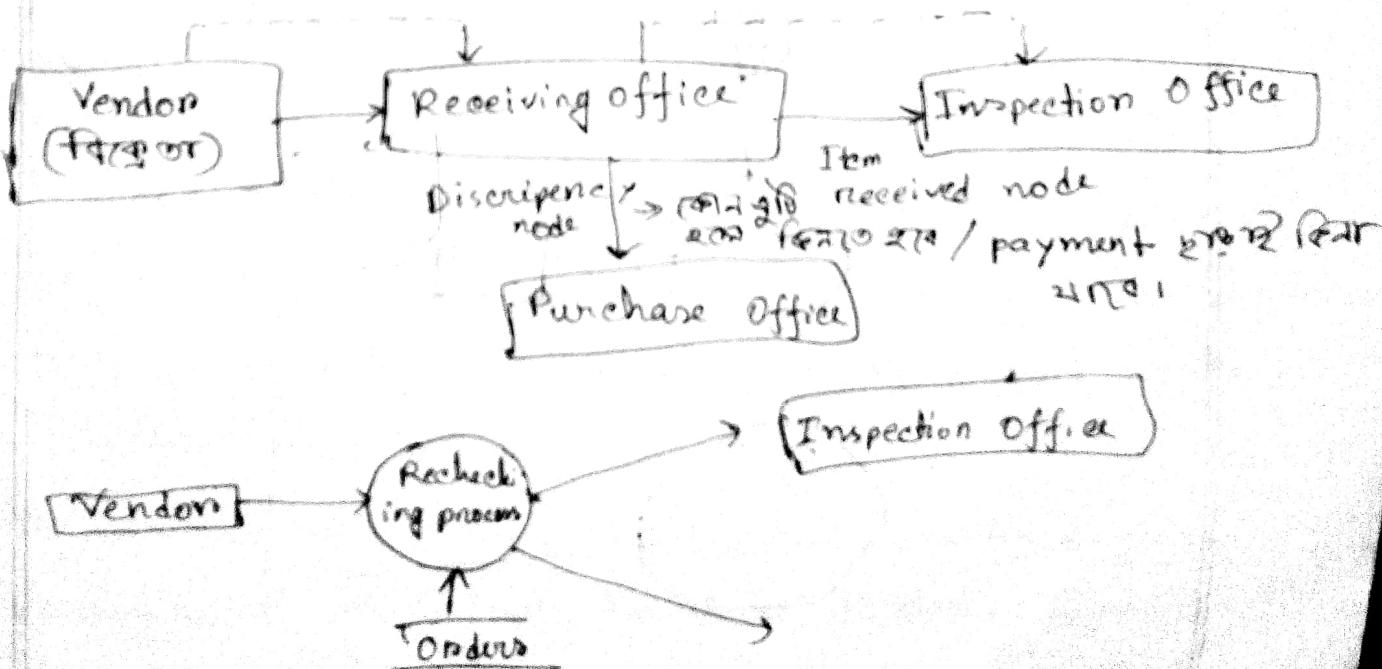
Info Source :

DFD : Document / Flow Diagram.

Data

যথিক করা, Document এরা স্টেট অফ, এবং ব্যক্তি

Database এ entity এর ক্ষেত্রে, class name → tablename
field → column



A circle → process
rectangle → external entity

22.05.1

1. Data store
2. Data flow

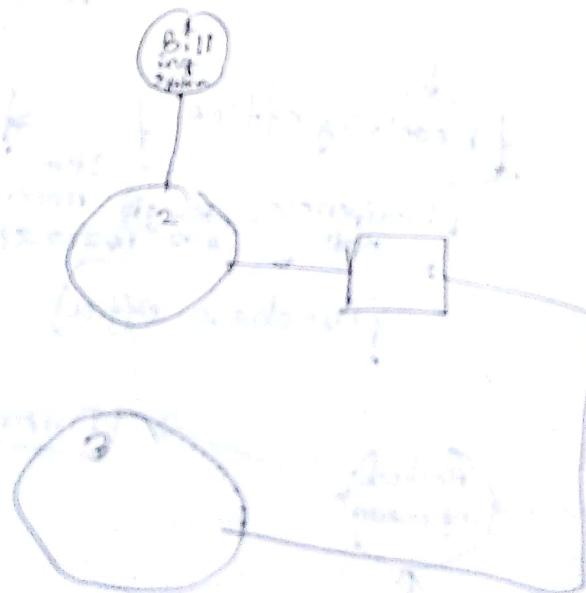
data store ↓
Inventory writing reading

Students → Vendors



Mess mandate

Chief warden



Assignment - 02

International Organization for Standardization (ISO):-

The International Organization for Standardization (ISO) is an international standard-setting body composed of representatives from various national standards organization.

Abbreviation : ISO

Formation : 23 February 1947

Type

: Non-governmental organization

Purpose

: International standardization.

Headquarters

: Geneva, Switzerland.

Membership

: 162 members (January 2018)

Official languages: English, French, Russian

Website

: www.iso.org

Founded on 23 February 1947, the organization promotes worldwide proprietary, industrial and commercial standards. It is headquartered in Geneva, Switzerland, and works in 162 countries.

It was one of the first organizations granted general consultative status with the United Nations Economic and Social Council.

CMMI : The Capability Maturity Model Integration
CMMI is a process and behavioral model that helps organizations streamline process improvement and encourage productive, efficient behaviors that decrease risks in software, product and service development. CMMI maturity levels help improve processes on any company scale.

A maturity level is a well-defined evolutionary plateau toward achieving a mature software process.

Each maturity level provides a layer in the foundation for continuous process improvement.

In CMMI models with a staged representation, there are five maturity levels designated by the numbers

1 through 5.

1. Initial.
2. Managed
3. Defined
4. Quantitatively Managed
5. Optimizing.

=0=

23. 05. 18.

ଅନୁଯାୟୀ
9:30am

81. 80. Eo

Technical Writing

Academic Research Professional writing

ଅବଳମ୍ବନ ଅବଳମ୍ବନ
ପରିଚୟ ପରିଚୟ

Personal info.: ସମ୍ବାଦ କୋଷ୍ଟ,

CV → Resume
detail summary

Int u result for :-

ଫର୍ମ କରିଛି :-

IEB → Institute of engineering

Case study - Hostel Info. System

Deficiency -

Main Goals;

Alternative Solution:

1. Improve manual system: କେନ୍ଦ୍ରିକାନ୍ତ କ୍ରମିକରଣ କରିବାକୁ
2. Computer use କରିବାକୁ
3. software : ମାଟ୍ରିକ୍ସ ଯେବେ ଉପରେ

(Video player ବ୍ୟବସ୍ଥା) - ଡିଜିଟଲ ଲୋଡ଼ିଂ ହେଲ୍ପିଂ

changes localization

SEIS

27.05.18.

28.05

McCall's Triangle of Quality :-

Product
transition

Portability : migration

3rd dimension of parameter

Reusability : अपार्स उपयोग करना।

Interoperability :

Product
operation

Correctness

Maintainability

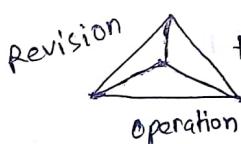
Usability

flexibility

Efficiency

Testability

Reliability



Integrity

Quality measure:

Matrices:

Process Indicator: efficiency.

process: well-defined (step by step task)

Software process: requirement, analysis, coding, design, Testing, deployment.

processes:

reinforcement:

uncover problem:

evaluate:

Project: goal define task (clearly defined objective)

Book: Soft. Engg. A practical approach

28.05.18

SEIS

Software Engineering
" Failure curve

H/W

Types of S/W Application

Netsourcing / open sourcing

Legacy software

Software Myths

" Products

" Cost

Project / Procure

Project Management Concept

4P

Project Role

Technical leadership Model

Project features Impacting Software Team

Software Team Model

Democratic / Deces

=0=

Information Gathering and DFD.

Info sources : interview : user or software (or) bat

* Information Gathering Strategies .

* Data Flow Diagram (DFD) .

users customer

software କେବଳ ଯିବୁ ବେଳେ କରାଯାଇଲା

entity: ମନ୍ଦିର କାର୍ଯ୍ୟ ପ୍ରସ୍ତ୍ରୀ ଥିଲେ ଏହାରେ ଆଜିମାରେ

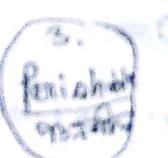
Process କରାଯାଇଲା ପାଇଁ ବିଶ୍ଵାସ କରିବାରେ List,

Entities କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା ,

କରାଯାଇଲା draw କରିବାରେ ,

feasible

$= 0 =$



30.05.18

SEIS

Hostel Info. System $\leftarrow \rightarrow$ TPS

System analysis করা পাবে ও আবশ্যিক নির্মাণ

- a. Improve manual system;
- b. pc based : software dev.
- c. Software বানাতে হলে এটা অনেক অরজন বিষয়।

b choose করো,
CBA → Cost Benefit Analysis:

অর্থনৈতিকভাবে কী সহজ কিমা,
অনসাধারণ সামাজিক example: কো বছর পর ফল দিবে, মাত্র কতুরো।
কোমল ধ্বনির কাটা

Benefit কি কি সরা - tangible : measure করা যায়
-intangible:
 customer
 satisfaction

case study of Hostel Info. System : (CBA) (Exam)

Amount :-

(one time) PC, UPS, Printer = 70 000

Capital :- এই invest করে কো খরচ করে (one time expense)

Rewurring : যার বার কিমা নাগারে, ex - পানি, গৃহী, কাশি,

 floopy disk → এটা নিরীক্ষা করো

(pendrive recurring ar)

: Benefits of Inventory \rightarrow (বর্ননালক্ষণ গুলিকা),

5% কাছলত, Daily rate = 45

$$\text{savings} = 45 \times 0.05 \times 50 \\ \times 400 = 27000$$

Pay back Period = ROI

BDT \rightarrow रुपये :

Trans. cont. = 800 रुपये \rightarrow वार्षिक बढ़ाव
400 रुपये amount

Direct savings = 337.90

recurring
cont.

Net return per month = $337.90 - 2000 = 317.90$.

Total Capital = 1,30,000

Pay back period :-

$$\frac{1,30,000}{317.90} = 41 \text{ months}$$

Monthly interest = $0.015 * 1,30,000 = 1950 \text{ per month}$
(15% loan rate)

Savings/month = $317.90 - 1950 = 297.90$

PP
ROI
of Investment

Cost recovered / Pay back period = $\frac{1,30,000}{297.90}$
 $= 4.4 \text{ month}$

CBA : present value method

If r = interest rate

n = no. of month

x = benefit

Present value = $x/(1+r)^n$

$= 317.90/(1+0.015)^4$

pay back period
4
5

1,22,450 → निवेदित ब्याज राशि
1,53,000 → उपलब्ध राशि
 $= 0 =$

03.06.18.

CT-01 Syllabus : feasibility Analysis.

Info. Gathering :

interview :- user story बनाते वर्षाहि एवं वक्तव्य तथा विषय।

↳ Ques. set करवा।

ट्रेईंग लोक एवं interview विषय तथा background तथा

group discussion

विवाह दस्तावेज आदि " "

Planning interview :- list करवा
note down the listed people
doubt वाले वर्षाहि interview
group dis.

Interviewing technique :- appointment

be punctual : आनुष्ठानिक रूप से अपना रखा।

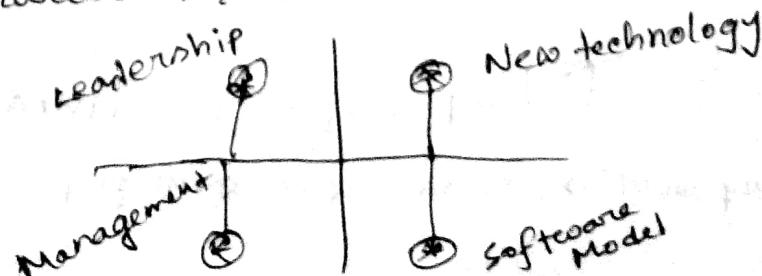
इत्यादि विविधाता
↓

कोरिट basic need
" " essential/desirable

मत point करवाते तर सुने रखें रखें।

prolong → दीर्घ वार्ता

questionnaires → प्रश्नपत्रक,



SRS →

Ideal characteristics of SRS :-

- Complete and unambiguous,

-

-

-

- User interface should not use terms of computer

- Jargon free.

→

The RAD Model

The Process Flow :

deploy :

Communication

Planning

Modeling

Construction → role play

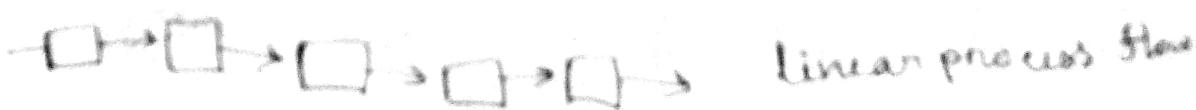
Deployment → homepage server deploy to host.

The waterfall Model : was used in

for software engg. and various other System analysis & time

various engg. (MCA CSE ITI)

"A 6 yrs. fault
8 year
2015-16"

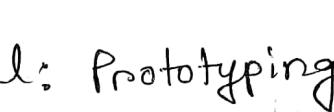


Building soft → diff. soft engg. areas

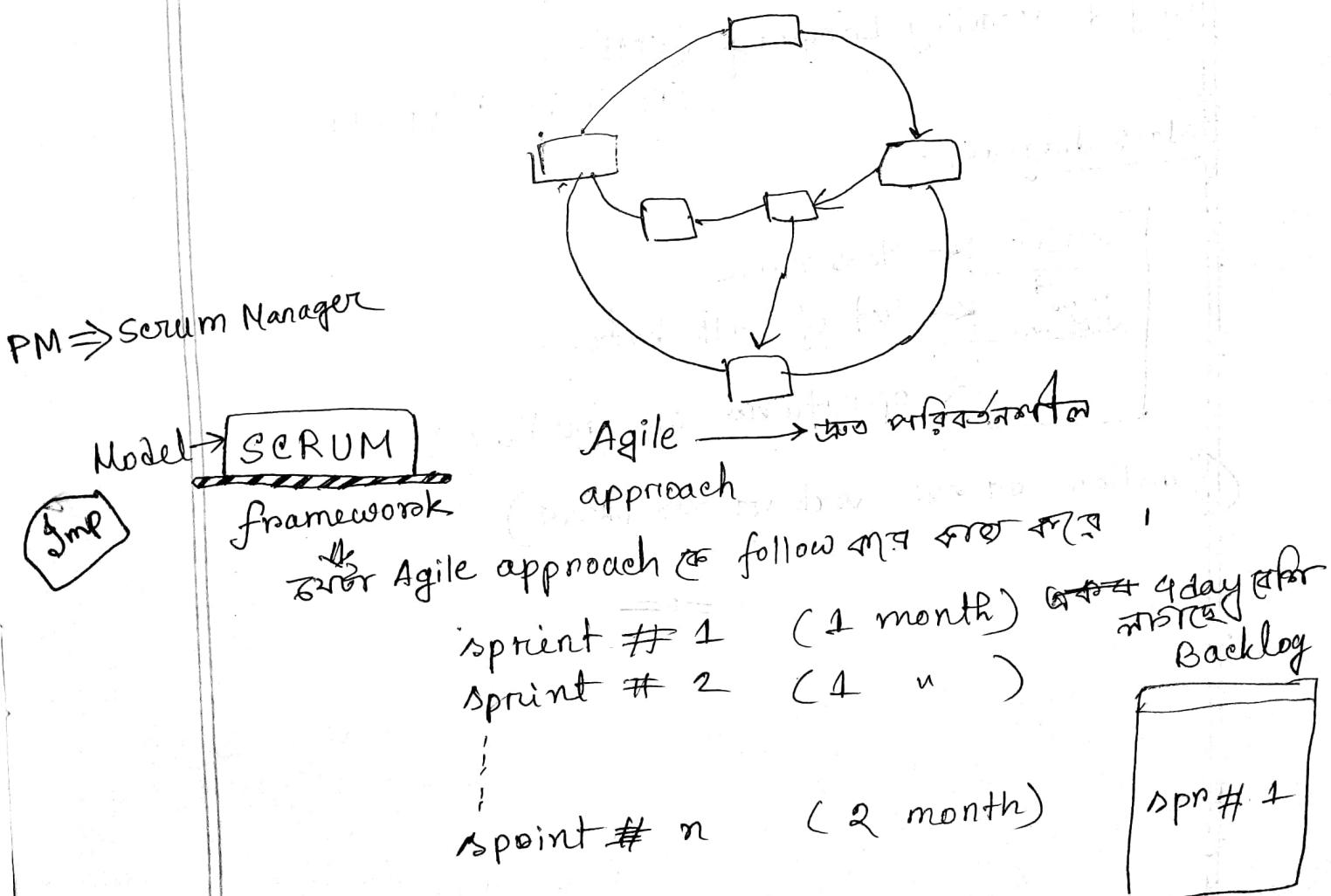
Incremental Model: পোকাপাকি যন্ত্ৰণা বৈজ্ঞানিক ।

- → color
- → ↗
- → ↘
- → ↙
- → ↘

Rapid Application Development (RAD):— অসম্পূর্ণ ফুল প্রকল্প এবং পুরো প্রকল্প একসাথে কৃত হবে।

Evolutionary Model: Prototyping  Demo দ্বারা ইটেলে প্রযোগ কৃত হবে।

Concurrent Model:



06.06.18.

Object Oriented System Modelling :-

ज्ञान सिस्टम के बहुतलों obj में वर्त दिया,

Desirable properties of components:-

component गुणात्रे अपने independent रूप से धूम,

Object oriented modelling

" and their properties ;

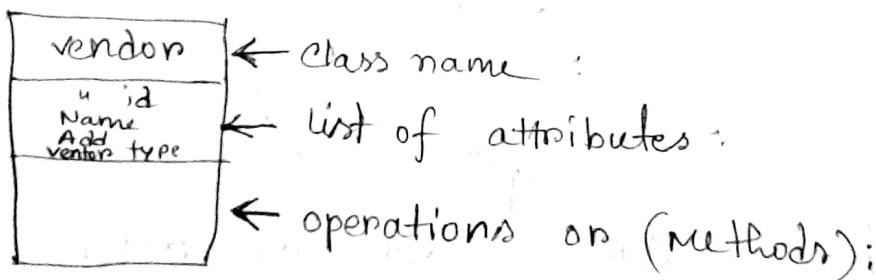
Object → entity

↳ properties → attribute.

Unified Modeling Language (UML) :-

(UCL) :- Diagram

Class diagram :



(function का नाम वर्ब का भी लिखा)

=o=

25.06.18

Agile method দ্বাৰা follow কৰিব কৈ Scrum framework.

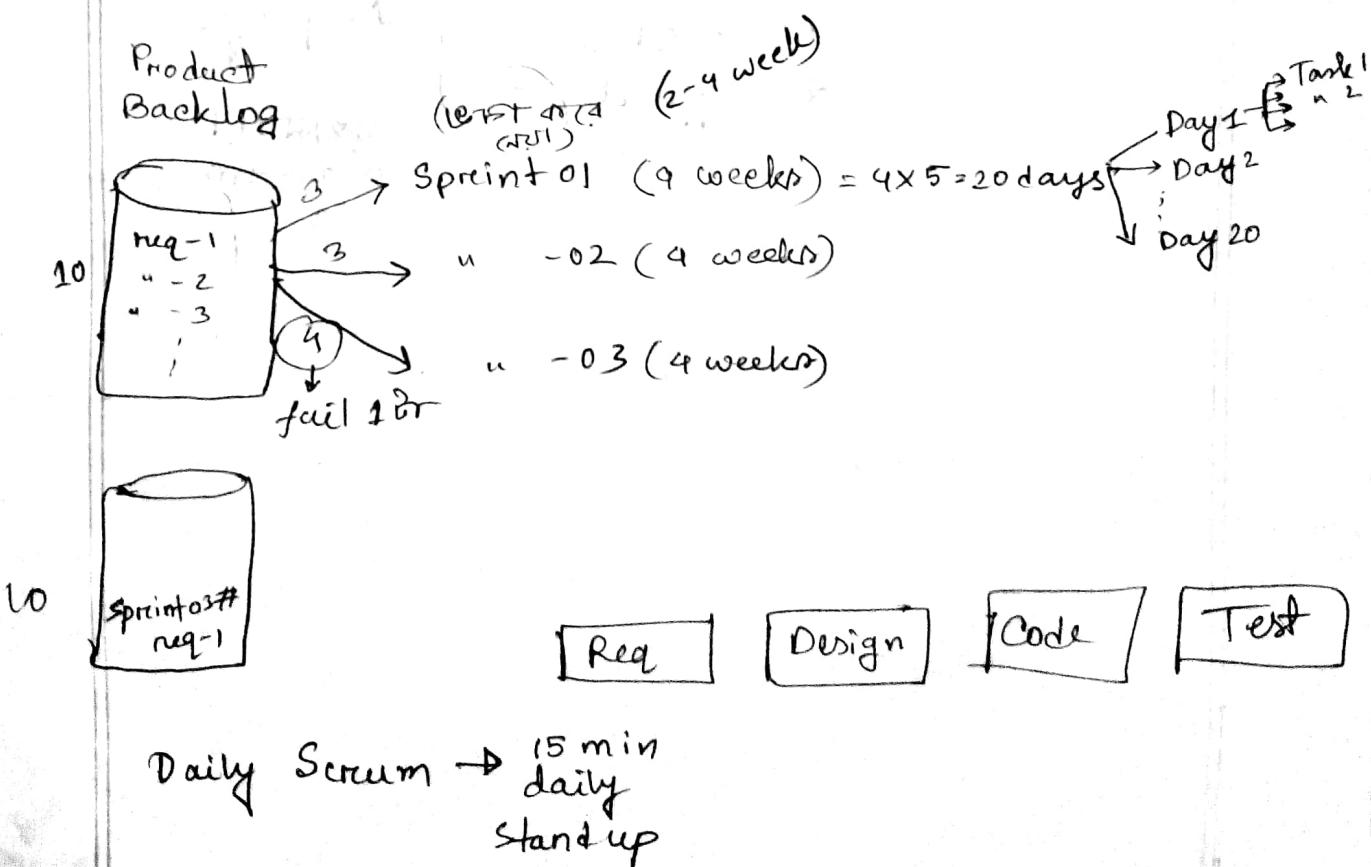
Method framework (নির্ধাৰিত rule আৰু কৰা)

Agile

Scrum

plan center people
যোৰ জোৰা দেবা
depend.

How Scrum Works?



Sample Product Backlog

| Item No | Description | Estimate | Bug |
|---------|-------------|----------|-----|
| | | | |
| | | | |

Sprint Planning Meeting :- SCR meeting, 20 tasks

1st part:

start with product owner

2nd " : यह technical एवं non technical ओर

तकनीकी

Assignment :-

- * Scrum
- * Waterfall
- * RAD
- * Spiral

when to use scrum?
when to use waterfall?

10 page

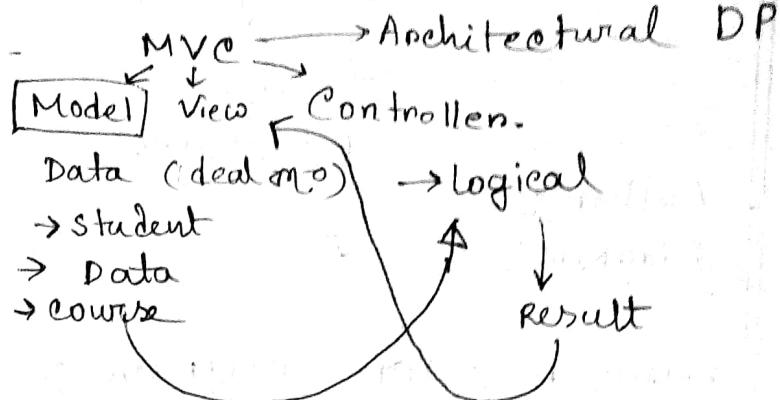
प्रक्रिया
class time

27.06.18.

view Layer :-

Design Pattern :-
(DP)

table → Model

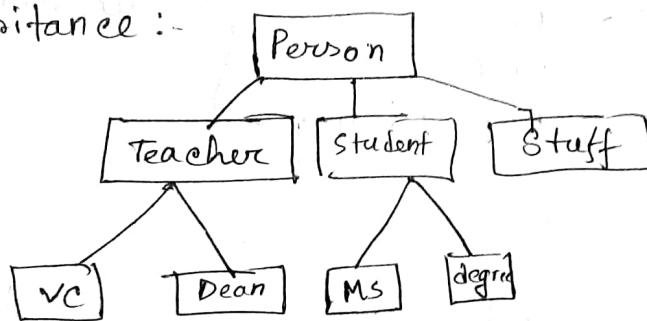
Model → Controller → View → controller → Model ...
database

Class diagram :-

Specialization
Inheritance :-

refer
specific

to AT
ATC ATC



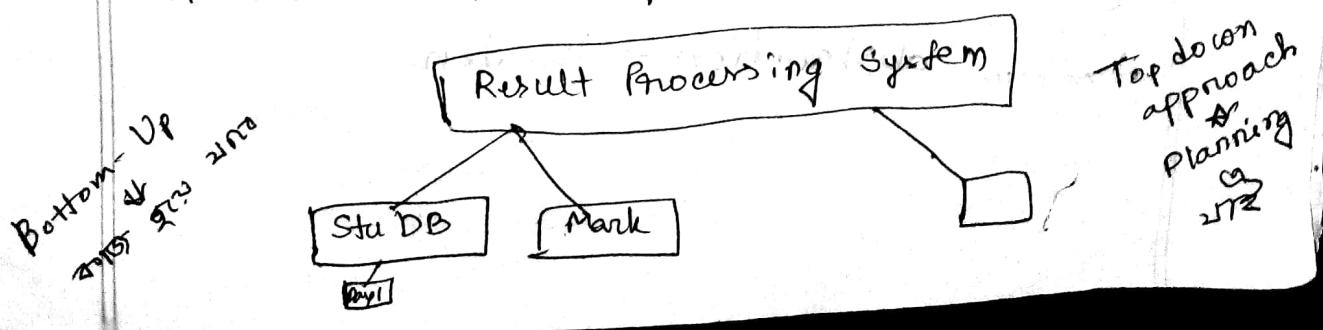
Generalization



Design Rep Schemes

H IPO : Hierarchy plus Input - Process - Output.

A tool for planning and documenting a comp



An Iterative Inventory Program

~~IPO~~

IPO Charts

Author :-

Diagram :

Warriner / Orrs Diagrams :- complex processes..

describe এর পর্যায়

use ২য়।

Main Elements : Bracket

Sequence

OR

AND

Repetition.

Question ...

— ० —

01.07.18.

Lecture - 09

Modular Software Development

কার্য (কাজ) task → module

Modular Approach: "কার্য (কাজ) scope খালির অংশ
শাখাৰ
হালুব।"

কার্য বানাবে কোথা ? :-

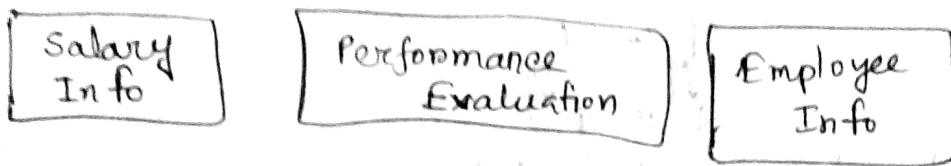
কোথা চাকু বানাবে

" abstraction → hidden

Next Lab :-

Draw

- Use case Diagram" of your project
- Presentation on "Selecting Soft. Development Model for your project". → which model?
→ why?



CEO. → All

→ Accessible (salary, performance, employee)

Line Manager

Team Lead → Performance
Employee Info

Employee → Employee Info

রাস্তা এনেন এট মডেল

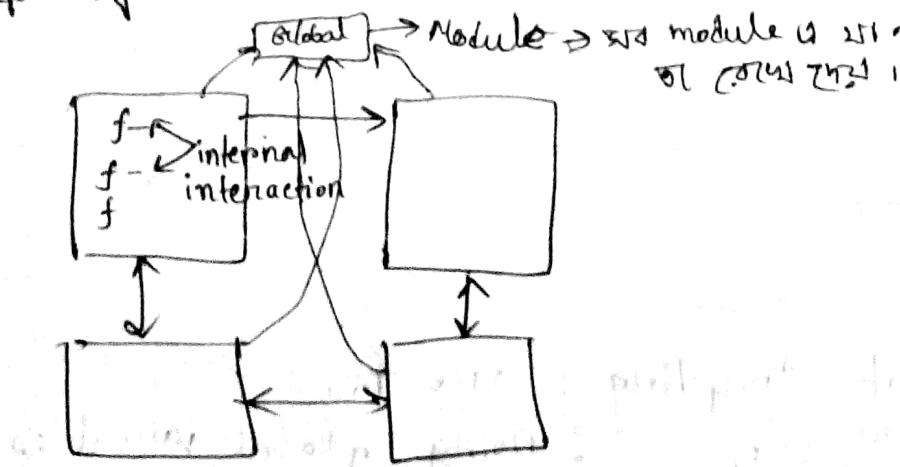
=0 =

04.07.18.

maximize

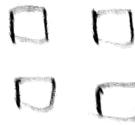
soft: cohesion :- internal iteration.

coupling :- minimize external interaction - dependency

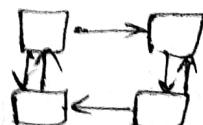


Characteristics of Good Design :-

Component :- High cohesion
low coupling.



No dependencies



Loosely couple

unnecessary for 3 statements

Content Coupling : easy approach / Bad practice

Common "

Control "

Stamp "

Data "

No

unnecessary for 3 statements

Content Coupling : easy approach / Bad practice

API

BAUST

get-use-result (use-roll, use-set)

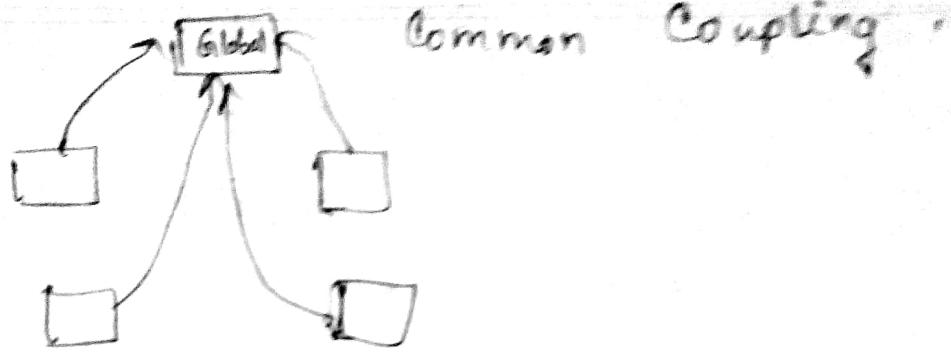
API

Education

get-use-set

}

Content Coupling : DCE for !
Common " : Using global variables .



- Disadv : i) star Global Module read write करता है।
ii) privacy तो compromise कर सकता है।
star access बढ़ा देता है।

Control Coupling :- control → code/logic share करता है।
with other module use करता है।



Scenarios के model का रूप !

Reports → ppt why choose
 → Use case Diagram
 →

Content Coupling → view level a तो क्या ?

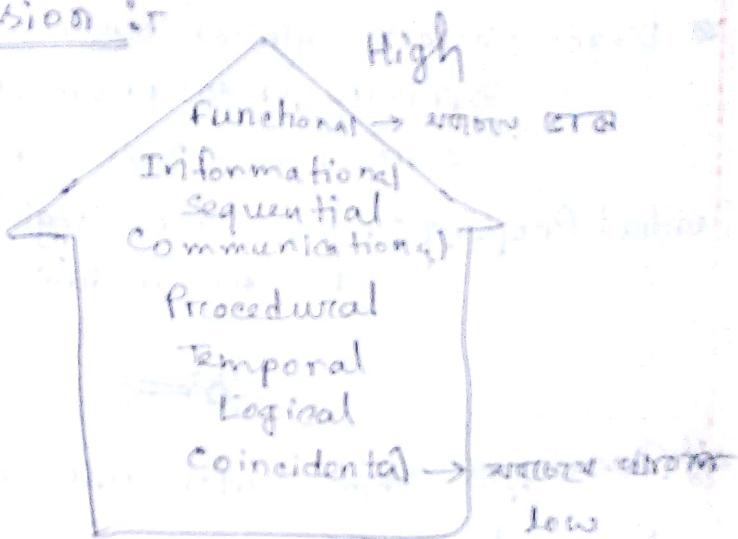
Stamp " → Data Structure a पास क्या ?
class ए शाही क्या ?

Data " → जो data का इसका API क्या ?

INTRO → Cohesion

units: Module ग्रा स्ट्रक्चर के element ग्रा ,
one : interaction वर्त एवं रेस ;
cohesion → glue up रहो , module निर्माण ग्लू
एवं ग्रा रहो ;

Range of Cohesion :-



Coincidental :- (No usability ..).

→ फैला सिर (पर अस्ति रहते हैं एवं module रहते हैं)

logical Cohesion :- logically core fu core, module रहते हैं

temporal :- वर्त उपयोग इनमें रहते हैं

Temporal cohesion :- Elements of a component are related by timing.

Ex - Bank a component/module रहते हैं time की विधि

Open or shutdown रहते हैं ;

~~Procedural~~: actions are weakly connected & and unlikely to be reusable.

Order maintain ~~area~~

Communicational Cohesion :- actions based on the ordering of steps on all the same data.

Actions are related but still not completely separated.

Sequential cohesion: one's output is the input for

Informational :- entry point $\xrightarrow{\text{C-O}}$ main funct.
no cycle between exit $\xrightarrow{\text{I-O}}$ function call

Functional :- item { are for common goal
achieve same }

Tokenizing a string of XML

Serialization :- convert object

predefined format (text or binary)

de-serialization = convert

object from text

root node content

child node for each parent

attribute for each element

value for each attribute

etc. (recursively)

de-serialization = convert

object from text

root node content

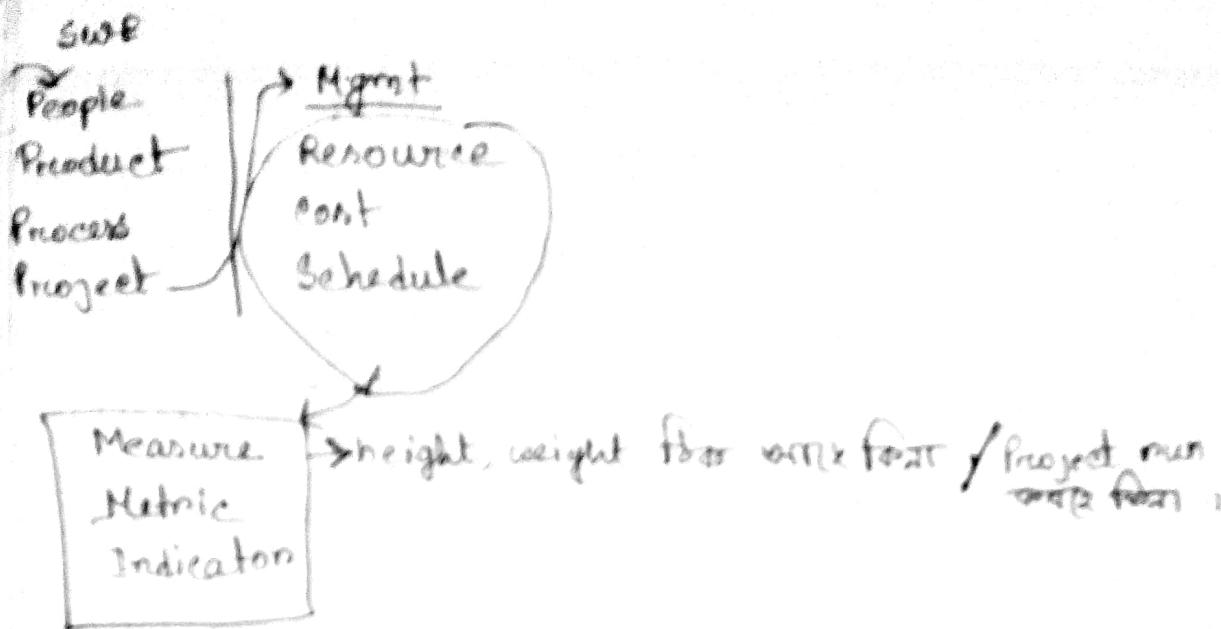
child node for each parent

attribute for each element

value for each attribute

etc. (recursively)

June 18

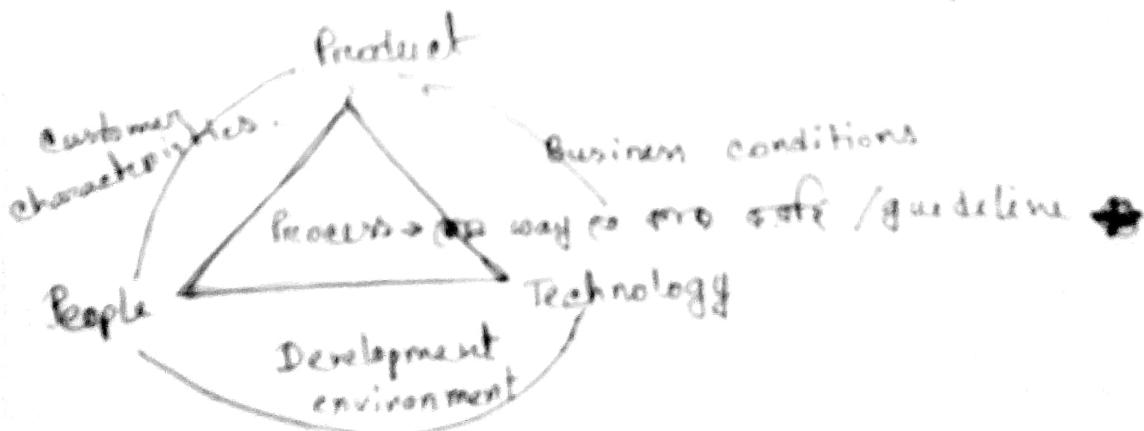


Metric for focus and projects:-

Measurement : Provides a mechanism for objective evaluation.

Assists in

- Estimation
- Quality Control
- Productivity assessment
- Project Control
- Tactical decision making .



Continuous process Improvement:-

SSPI → Statistical Software Process Improvement.

Error → Defect

defect →

Project Metrics :-

Software Measurements :-
1. Direct
2. Indirect : (Error Rate / LOC)

Size - Oriented Metrics :-

LOC (Line of Code) :-

Raihan - 1 kLOC

Rafid - 950 LOC

Faisal - 1 kLOC

Nahid - 2 kLOC

Errors per KLOC

Defects u kLOC

£ per KLOC

Pages of documentation per kLOC

Person - month → 1 person works with 1 month effort

10 months for 1000 validating 1 set of format .

LOC % programming language dependent.

function - oriented metric is the function point (FP)

proponent
FPCB

opponents
FPR

→ Based on subjective rather than

Object-oriented Metrics :-

UML : Unified Modeling Language

Use Case Diagram :- Scenario

key class: highly indpt. class

Supporting ":

Use case Oriented Metrics:

independent of prog. language,

Web Engg Project Metrics

N_{sp} = no of static web pages

N_{dp} = " dynamic "

Customization index, $C = \frac{N_{dp}}{N_{dp} + N_{sp}} = \frac{6}{10+6} = 0.5$

Metric for softw quality:

Quality →

Mean-time-to-change (MTTC)

integrity = $\sum [1 - (\text{threat} \times (\text{1 - security}))]$

threat is the probability that an attack of a specific type will occur during a given time

Security = the probability that the attack of a specific type will be repelled.

DRE: Defect Removal Efficiency

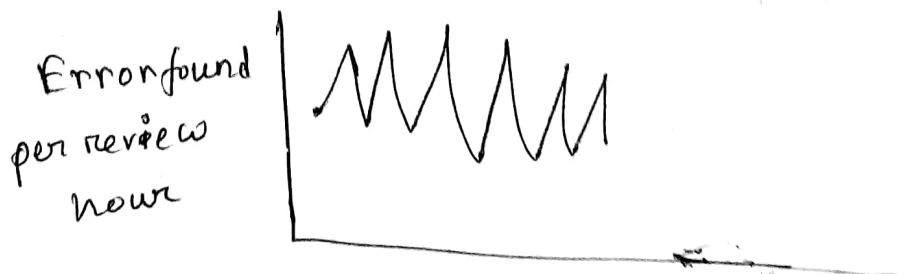
$DRE = D / (C + D)$

$DRE_i = E_i / (E_i + E_{it})$ (ith activity)

Fig: soft metrics collection process

Managing variation:

Moving avg : Control chart



mRbar →

Multiply the mean by 3.268.

UNPL → Upper natural process limit

LNPL → lower

4 br condition meet હા એવું - stable

એંફોર્મ 1 br " " " - unstable.

BMI :-

External output / Input
logical files

value Adjustment Factors :-

SafeHome software :-

$$AP = \text{count total} * [0.65 + 0.01 * \text{sum}(f_i)]$$

Metrics for the Design Model :-



fan out → કાઢ
શકીય
func. call
કાર્યર

Shape complexity :- size = $n + a$

↑ no of node
↓ no of arcs

coupling ~~positive~~
cohesion ~~negative~~

Population :

Volume :

Length :

Functionality :

Weighted methods per class (WMC) :-

$n_1 \rightarrow$ no of distinct operators that appear in a program

$n_2 \rightarrow$ " " " operands " " " "

$N_1 \rightarrow$ total " " operators " " " "

$N_2 \rightarrow$ " " " operands " " " "

Project Scheduling :- project is ~~to work~~ ~~work~~.

10% of projects successful betⁿ 1998 → 2009

Software project scheduling is an action that distributes estimated effort across



Basic principles:

Compartmentalization:

Milestone →

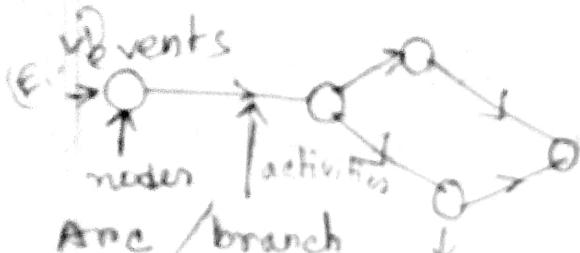
Output →

Effort
cost

→ Development timer

Project: Goal define task.

Activity Network :- A task network, is a graphical rep. of the task flow for a project.



Scheduling 2 methods : CPM : Critical path Method
PERT : Program evaluation and review technique
(PERT) (Probabilistic)

Time-Line Charts / Gantt Chart :-

Project table :

Tools help :-

- Microsoft Project
- Open Proj.org
- Open Workbench.org

Critical Path Analysis : is a special method for time analysis to determine following :

- i. Total duration for project completion
- ii. Categorize the activities of the project in two types : critical and Non-critical

As soon as the network is drawn for the given project, the time analysis is required for planning of various activities of the project. The main aim of time analysis is to prepare a planning schedule for the project.

Critical Activity :- An activity in network diagram, whose delay in beginning will further delay the project completion time.

Non-critical : An activity which allows some scheduling slack, so that the start time of the activity i may be delayed or advanced within some range without affecting the completion time of entire project:

E_i = Earliest occurrence time of event i

L_j = Latest " " " " " " j

(i, j) = Activity with head event j and tail event i .

t_{ij} = Duration of activity (i, j)

ES_{ij} = Earliest starting time of activity (i, j)

LS_{ij} = Latest " " " " " " (i, j)

EF_i = Earliest finishing " " " " " " (i, j)

LF_{ij} = Latest " " " " " " . "

Forward pass calculation :- Start from the initial node (i) with starting time of the project as zero. We move along the nodes with increasing order

Algorithm for forward pass

Step-1: initialize $i=1$, $E_i = 0$ (first node)

*-2: Calculate earliest start time for each activity that begins at node i , as $ES_{ij} = E_i$ for all (i, j) with starting node j .

*-3: Compute earliest finishing time for each activity that begins at node i , as

$$EF_{ij} = ES_{ij} + t_{ij} = E_i + t_{ij}$$

where, t_{ij} = duration of (i, j) .

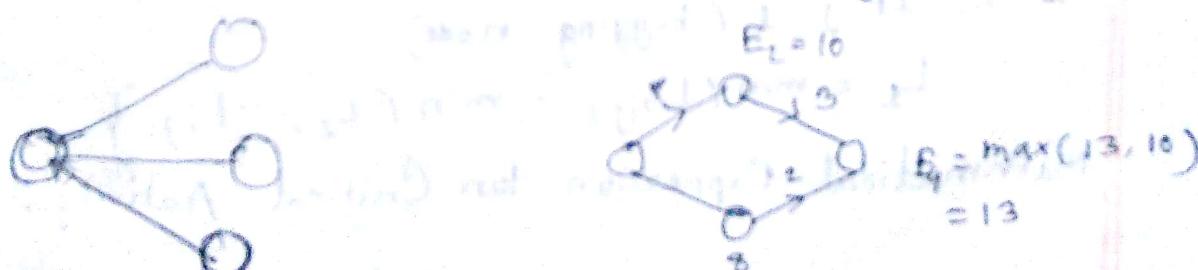
*-4: Proceed to next node, say node j ($j > i$) and compute the earliest occurrences for node j using

$$E_j = \max(EF_{ij}) = \max(E_i + t_{ij})$$

for all immediate predecessors i

*-5: If $j=n$ (final node), then the earliest finished time for the project is given by:

$$E_n = \max\{EF_{ij}\} = \max\{E_i + t_{ij}\}$$



forward \rightarrow max
backward \rightarrow min

Backward pass calculation:

Here, we begin from terminal (Last) node of the net^w proceed through the net^w visiting nodes in the decreasing order of node numbers and end at the initial node. At each node we calculate the least finish time for each activity.

Step 1: Initialize $E_n = E_n$ for $j=n$

" 2: Set the latest finishing time for each activity (i,j) that end at node j .

$$LF_{ij} = L_j$$

" 3: Calculate latest starting time for each (i,j)

$$LS_{ij} = LF_{ij} - t_{ij}$$

" 4: Proceed backward to the node in the sequence that decrease j by 1. Latest occurrence time of node i ($i < j$)

$$L_i = \min \{ LS_{ij} \} = \min_j \{ L_j - t_{ij} \}$$

" 5: If $j=1$ (beginning node)

$$L_1 = \min (LS_{ij}) = \min (L_i, -t_{ij})$$

Mathematical Expression for Critical Activity:

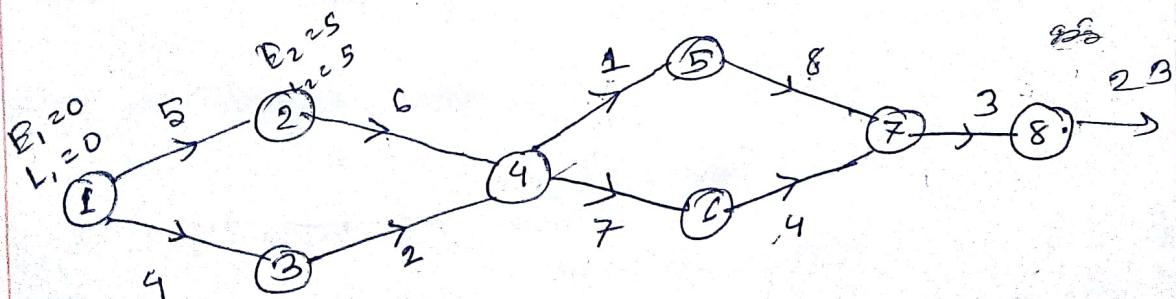
i. $E_j = L_j$ and $E_i = L_j = L_i$

ii. $E_j - E_i = L_j - L_i = t_{ij}$

Critical path: The sequence of critical activities in a network is called critical path. It is the longest path in the net^w. from starting event to ending event and determines the min time for project completion.

Float and Slack time:

| Activity | Immediate Predecessors | Time |
|----------|------------------------|------|
| 1-2 | | 5 |
| 1-3 | | 4 |
| 2-4 | 1-2 | 6 |
| 3-4 | 1-3 | 2 |
| 4-5 | 2-4, 3-4 | 1 |
| 4-6 | 2-4, 3-4 | 7 |
| 5-7 | 4-5 | 8 |
| 6-7 | 4-6 | 4 |
| 7-8 | 5-7, 6-7 | 3 |



Critical path:
1-2, 2-4, 4-6, 6-7, 7-8



a) An activity with zero total float is called "critical activity"

float and slack times:

↓ events

used for active activities \rightarrow float

" " events \rightarrow slack

slack of an event: Event float

$$\text{Event slack} = L_i^o - E_i$$

Float of Activities:

i) Total float, TF_{ij}

ii) Free

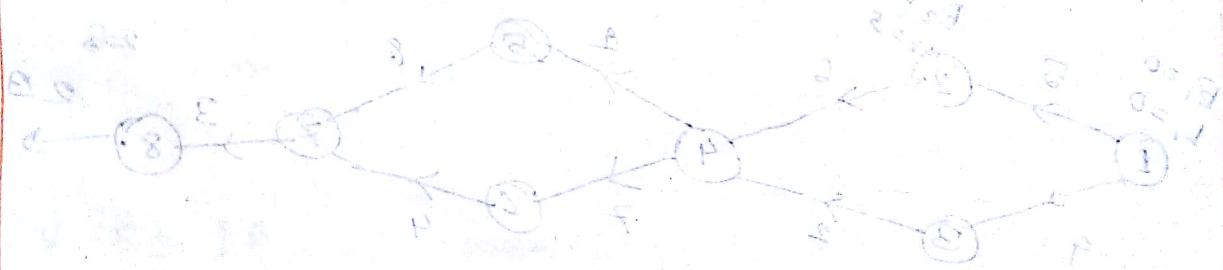
iii) Independent

iv) Inference

$$i) TF_{ij} = LS_{ij} - ES_{if}$$

$$ii) FF_{ij} = (E_j - E_i) - t_{ij}$$

iii)



8 - ES₁ + EA₁ - ES₂ + EA₂ - ... - ES_n + EA_n = AF

8.07.18

Design Pattern \hookrightarrow Obj oriented prog.

Design patterns are reusable solutions to the problems that we encounter in the day today programming approach \rightarrow solve prob \leftrightarrow solve prob.

In SE, a dp is a general repeatable solution to a commonly occurring problem in soft. design.

DPS are used to represent some of the best practices adapted by experienced obj-oriented soft. developers.

Standard \rightarrow follow \rightarrow www.12factor.net

Evaluation of design patterns \rightarrow www.12factor.net four writers

Usage of Design Pattern :-

1. Common platform for developers :- www.12factor.net
2. Best practices :- www.12factor.net

MVC \rightarrow Design pattern :

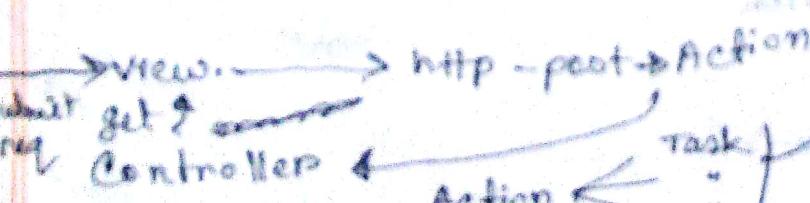
Architectural n

MVC (full project \rightarrow design pattern www.12factor.net)

Submit

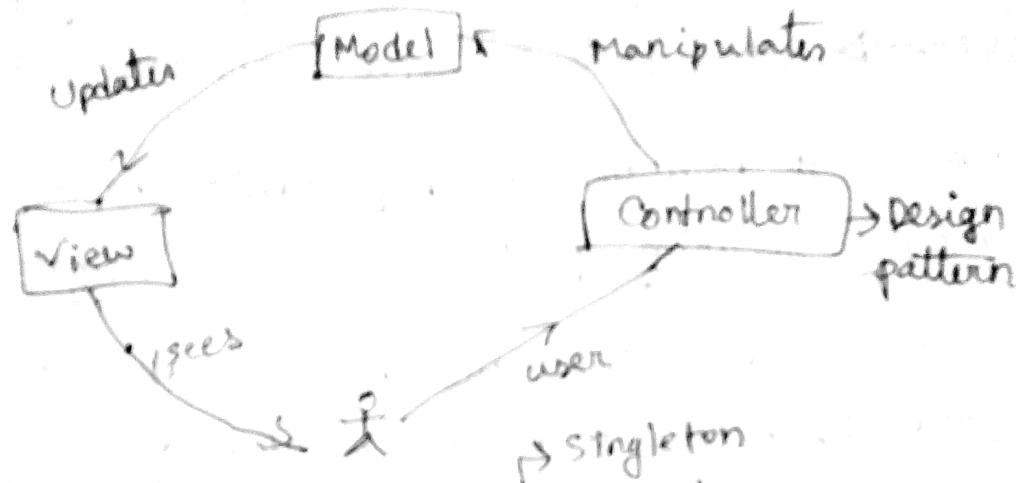
Model A

User



get \rightarrow
task \rightarrow
Action

user → view → Controller → Model → View



Types of DP :
1. Creational → class and obj.
2. Structural and → Coupling.
3. Behavioral →
 class and obj →
 communication →
 prob
(chain relation)

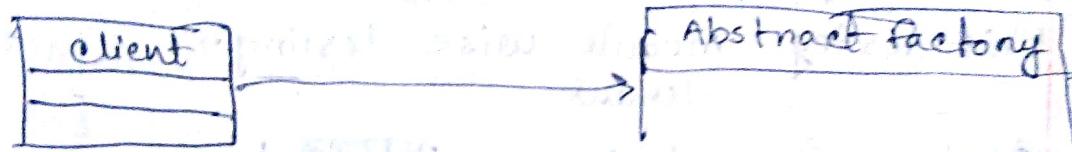
Pre-requisite Concept:

- Abstraction
- Inheritance
- Polymorphism
- Encapsulation
- {- Interface
- Class
- Abstract class

16.07.18

Implementation type 2^o solution Design Pattern

Abstract Factory Pattern :- Class ~~has~~ ক্লাসের field
↳ class কে কোর্ট implement এবং override attribute
Class diagram :-



Abstract Factory :- declares an interface for operation that creates abs. prod.

Concrete " :- implements the operations to create concrete prod. objects.

Abs: Prod :

product;

Client:

interface IDumb { string Name(); }
c# class IDumb { }
interface IDumb { }

CT-02

Lab Report

C++ Abstract Factory.

Code

→ Modular
Soft Develop

18.07.18

Chp - 14

- Logic
- Problem Solving skill

Soft. Testing Technique ..

Software Testing Techniques:-

PM

Soft. Developer

Unit testing : module wise testing of element

Developer নিজের test এর মাধ্যমে।

Local system এর soft config database metadata.

live এর উপর গুণগত PM।

deploy করি। It দ্বারা পরিষ্কার করা হয়।

CLR → .net এর virtual machine। DB 3 readers
C# এর func. library.

data type টাক্সি করা থাকে।

Layer approach →

live server → photocopy
strate " test

⇒ Algorithm → linked list, palindrome,
→ DB, Stack, Queue, SQL (Joining)
↓
Problem.

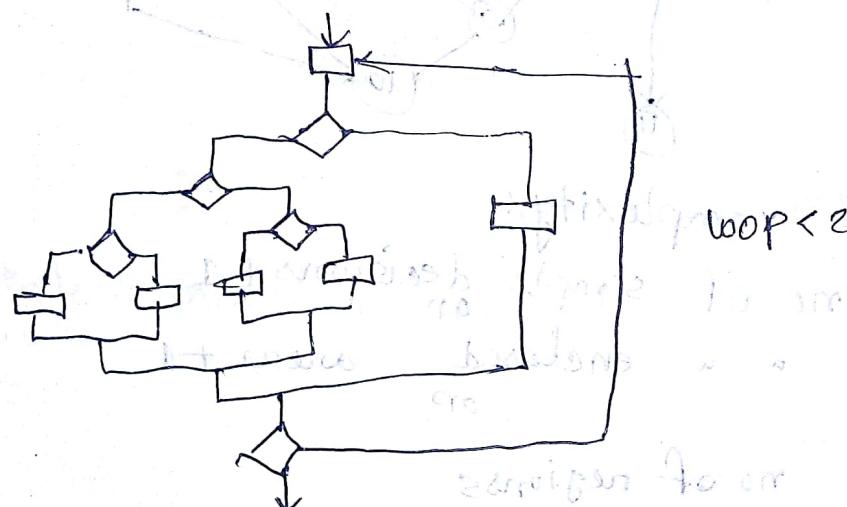
2.07.18.

A good test has a high probability of finding an error.

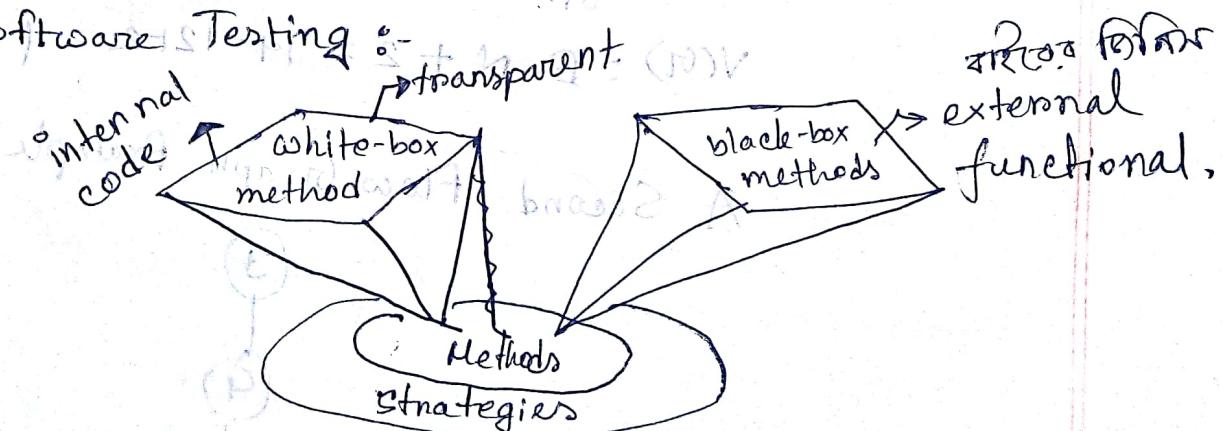
Internal and External Views :-

- functionality फैला करो, code नहीं,
- ☒ Code फैला test करो,

Exhaustive Testing :-



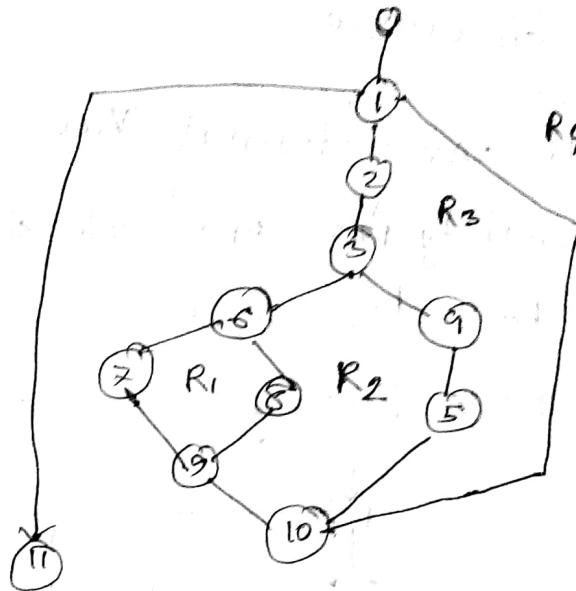
Software Testing :-



BPT : Basis Path Testing.

Flow Chart

Flow bipraph



WBTG

cyclomatic complexity:

no of simple decisions + 1
on

" " enclosed areas + 1
or

no of regions
on

$$V(G) = E - N + 2 = 14 - 12 + 2 = 4$$

A Second flow Graph Example



public class B { } ↳

public static void main() { }

B ob = new B();

B. ob1 = new B();

ob.a;

ob1.a;

}

factory Design Pattern

interconnection betn class and obj

F

Factory design pattern

Factory design pattern

has a relationship with C → factory

has a relationship with D → factory

has a relationship with E → factory

has a relationship with F → factory

has a relationship with G → factory

has a relationship with H → factory

has a relationship with I → factory

has a relationship with J → factory

has a relationship with K → factory

has a relationship with L → factory

has a relationship with M → factory

{ Factory Desn Pattern: Factory method is a creation desⁿ pattn → obj soft engg !

In FDP, we create obj without exposing the creation logic to client.

#A des without fac pat.

```
#include <iostream>
```

```
using
```

```
library clssns
```

```
class Vehicle {
```

```
public :
```

```
virtual void printVehicle() = 0;
```

```
}
```

```
class TwoWheeler : public Vehicle {
```

```
public : void printVehicle () {
```

```
cout << "I am two wheeler" << endl;
```

```
}
```

```
class FourWheeler : public Vehicle {
```

```
public :
```

```
void printVehicle () {
```

```
cout << "I am four wheeler" << endl;
```

```
}
```

Pro desn Patt is a creational des pat

Inheritance (E) class

client → class animal → model copy add

Adapter Patter: दोयन त्रुप्ति द्वितीय class

एक बिल नहीं, अपने द्वितीय
class के merge करो

class diagram :-
pseudo code :-

1. Singleton DP :- इतना एक टैक्सि करि बाहेर
मिळावा.
2. Factory DP :-
3. Prototype DP :-
4. Adaptor :-
5. Chain of Responsibility :-
6. Strategic :-

Normal

Pattern

Design

Model

View

Controller

Business

Entity

Utility

29.07.18

SEIS

UML \Rightarrow Unified Modeling Language.

Black Box Testing :-

User name

Password

User name

Test Case - 1 : Should not full numeric

" " - 2 : 1st char should not be numeric

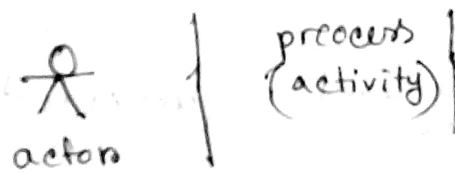
" " - 3 : Should not contain space.

" " - 4 : Min length & char. password:

1 Capital; 1 numeric, 1 special case.

Min length 8

UML \rightarrow 1. Use Case Diagram. (अवार्द्ध त्रिकोण मानदण्ड)



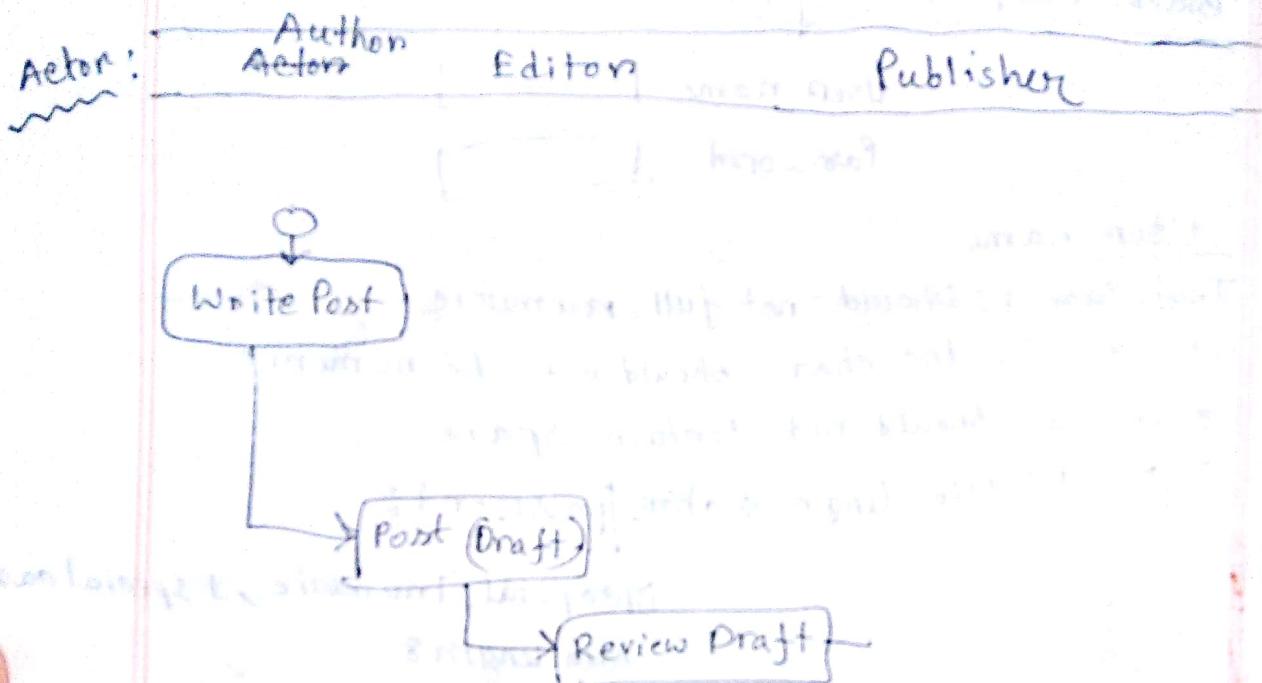
Design pattern
technical
pattern
in 2. Class Diagram (class का लिए activity)
factory pattern

What for UML का नाम क्या है?

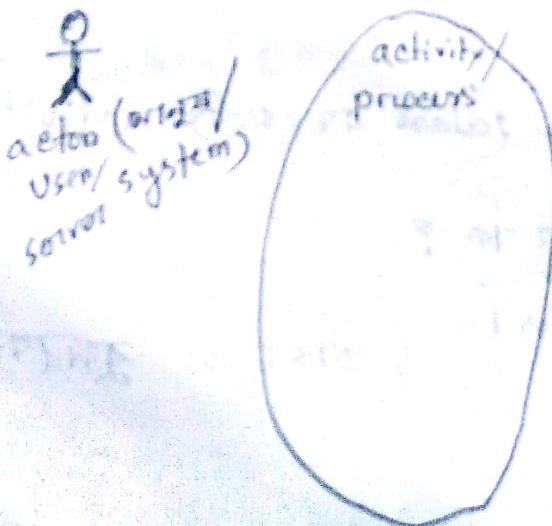
Normal client or class for { user or उपयोगी ।
obj " }
func " }

(~~natural~~
activity flow
~~etc etc~~) Behavioral UML Diag → Activity Use Case Sequence

UML → Activity diagram :- Activity flows & state transitions



Use case diagram: (high level requirement)



Sequence diagram :- শুধু computer science community
না business application develope.
মাত্র কাঠে বস্তুত পাইল (technical
knowledge ETC)

Course registration system

কাঠের sequence সম্পর্ক করব।

Register

= o =

activity diagram: Lab

30.07.18

actor কাঠের মধ্যে use case করব।

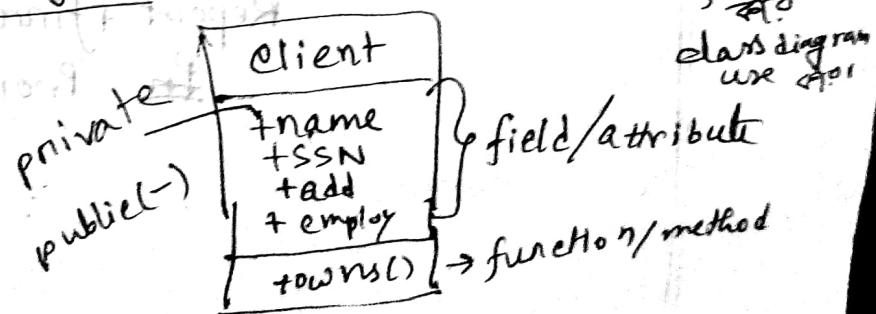
sequence diagram: function related কাঠের
(normal people কৃত্যের) itperson

কাঠের
কৃত্যের
use case

- logical approach
- function → bold line
- back n -- → dashed

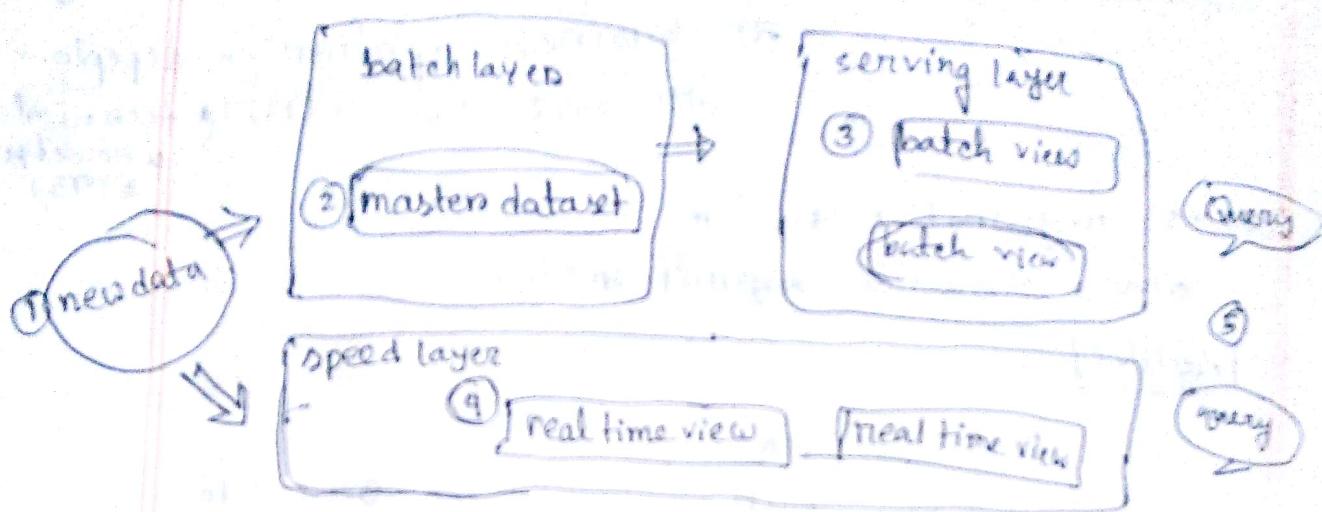
Register

Structural UML Diagram: class diagram → OOP
follow কাঠের
class diagram
use case



Hadoop

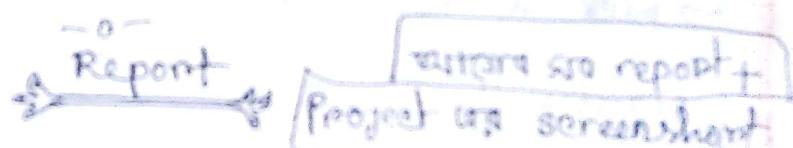
Object Component Diagram :-



Deployment Diagram :- To deploy a live system,

Nodes → application server / database server

Artifacts → " client fury schema "



Software Quality Assurance

01.08.18

SQA → tactical care for strategic ~~care~~ ^{plan} ~~care~~

↳ quality compromise ~~quality~~ ^{quality} ~~quality~~ ^{quality}

SQA consists of a means of monitoring the software engineering processes and methods used to ensure quality.

CMMI

SDLC

SQA engineering?

Q. What does a soft. quality engineer do?

→ Tester

Software engineer.

Quality Control for?

Quality Management → also called soft. quality assurance

Quality control team for ~~design~~ ^{design} ~~design~~ design pattern use

" Defined : Defined as a characteristic or attribute of something.

Compare ~~and~~ !

Quality of design : design as quality ~~as~~ ^{as} ~~as~~ ^{as}

" " conformance

user satisfaction

Quality Control : ~~work~~ ^{work} ~~review~~ ^{review} ~~review~~ ^{review}

Cost of quality :

Kinds of Quality cost?

Prevention costs

Appraisal " "

Failure " "

SQ Defined:

Purpose of review:

practices for writing good code

14, 15, 16

so =

27.08.18.

~~the following are the following principles~~

Habibullah sir:

Lecture - 11

Solid Principles

~~all~~

Design principle. the 2nd solid principles.

Obj. oriented program start face starts ORJ

from Algo start,

get your faces common prob. soft project solve ORP
solution from implementation ORJ, project will

start from Algo start ORJ solid principle.

relation to other design principles

Usage of Solid Principles:- soft project ORJ
flexible ...
loose coupling.

Soft design ORJ maintainability managing risk

S : Single responsibility principle (SRP)

O : Open closed principle (OCP)

L : Liskov substitution principle (LSP)

I : Interface segregation principle (ISP)

D : Dependency inversion principle (DIP)

SRP : A class should have only one reason to change.

↳ single responsibility

LSP : superclass \Rightarrow instance \Leftarrow subclass

OSP : Software entities should be open for extension, but closed for modification.

ISP : client-specific interface

DSP : depend on abstract class not on concrete class.
concrete class \Rightarrow dependency

V.V.S Design Pattern vs Solid Principle

1. It provides low-level solutions. It provide high-level guidelines.

A successful application development depends on

1. Architecture :

2. Design Principle :

3. " Pattern : code का रूप

03.09.18.
next: Lab 4 final (individual work)

(individual work)

next: Presentation + Report. (group work)

08.09.18.

Extra Class

Software Engineering & Information System

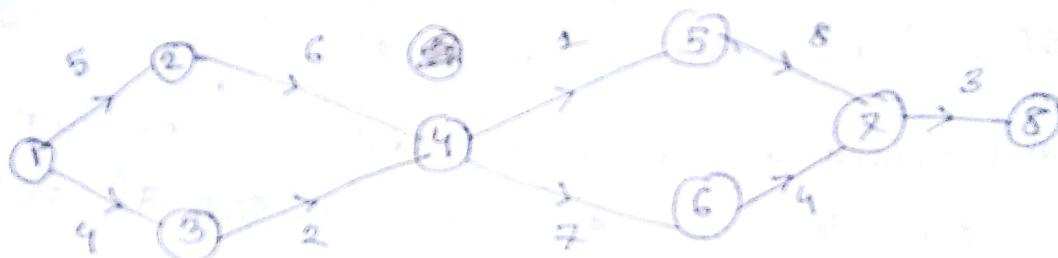
CSE - 4103

→ Engineering :-

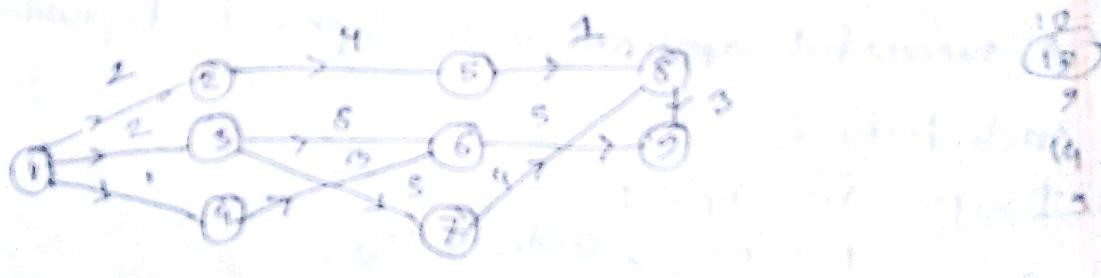
CPM

PERT

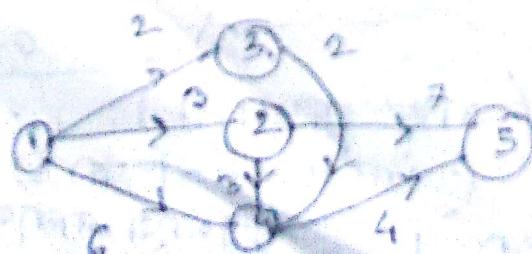
Critical Path Activity :-



1 → 2 → 4 → 6 → 7 → 8



1 → 3 → 6 → 9



1 → 2 → 5

1 → 4 → 5

1 → 2 → 4 → 5

~~Time~~ Earned Value Analysis :- is a measure of progress.

BCWS → Budgeted cost of work scheduled ..

BAC → $\sum_{i=1}^n \text{BCWS}_{\text{Actual}} \text{ from } 0 \text{ to end date}$

PERT : Prog. Evaluation and Review Technique.

* Developed by NS/Navy for Polaris Missile project.

* It can handle uncertain activity time.

PERT uses 3 cases :-

i. Most optimistic

ii. " pessimistic

iii. Average

Optimistic time (t_o) :- Time estimate of an activity when everything is assumed to go well as per plan.

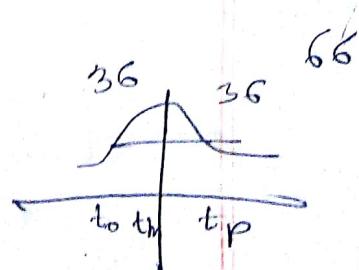
It is the

Variance = σ^2 σ = standard deviation.

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

$$\sigma^2 = \frac{(t_p - t_o)^2}{36}$$

$$t_p - t_o = 6\sigma$$



crash :- force crashing :-

The critical path is the path of least float for such kind of PERT network in which backward pass is based on the scheduled time.

PERT Algorithm

Step 1 : For each activity estimates of optimistic time, pessimistic time and most likely time.
(These estimates are supplied by project manager or someone who is familiar with similar projects.)

u2 : Using these estimates, compute the mean and variance of each activity times by the formulae.

u3 : Based on mean activity, determine the critical.

Time-Cost Trade-off in PERT :-

→ Project Managers sometimes may have the option of crashing activities, spending extra money to compress an activity's duration by using

Crashing strategies :-

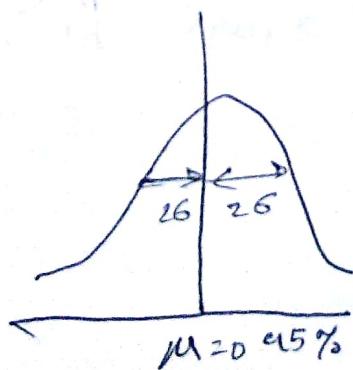
→ Don't crash Non-Critical activities. (This will not reduce cost in general).

→ Crash only critical activities, activities on the critical path

→ Crash activities with lowest crashing cost per unit time until desired project duration will

crashed cost C_c % The

$$\text{cost slope} = \frac{\text{crash cost}}{6}$$



$$\begin{aligned} \mu &= 0 \\ \sigma &= 1 \end{aligned}$$

$$z = \frac{x - \mu}{\sigma}$$