Sir Conrad
Walter D'Silva
conrad.dsilva@dsu.edu.pk.
30-10-23
SE-225

Introduction To Software Engineering

Text Book: Software Engineering By Ian Somerville 9th Edition.

Ch # 1: Introduction

Software Metrics.
" Quality
" Control
" Architecture
" Testing
" Project Management

Industry		
	caler	system analyst
min	15k	25k
max	25k	35k
	≈ 70k	≈ 90k

estimated figures.

Software Engineering:

- Economies are dependent on software
- Systems are software controlled.
- SE is concerned with theories, methods & tools for professional software development.

MIS
Management Information System

Software costs:

Software engineering is concerned with cost-effectiveness.
Software cost often > hardware costs.

Software Products.

→ Generic Products: Stand-alone systems, a general system which any customer can buy. (Developers)
MS OPM
CAD.

→ Customized Products: for a specific customer fulfilling their needs. (customer)
Embedded control system, traffic monitoring;

80% 20% Pareto Principle

Documentation coding

P.M. Tools
MS office
Project Management
CAD
Computer Aided Drafting
" Design
" Drawing

PAPERWORK

Software: Computer programs & associated documentation.

Attributes of a Good Software.

→ Maintainability: Changing or updating your system will not have any effect on its previous state. it is maintainable.

→ Dependability ex: Reliability, security & safety
Security

→ Efficiency : optimized use of responsiveness, processing-time & memory utilizations

→ Acceptability: Must be understandable, usable & compatible with other systems that they use.

Software Engineering.

- engineering discipline.
- software production from early stages to System specification. through maintaining the system in the its use.

SDLC (System Software Development Life Cycle)

UML Unified Modeling language

→ feasibility (technical, operational, economical)
→ analysis (requirements gathering).
→ design (frontend, backend, UML Diagram).
→ coding (programming)
→ testing (unit, integration, system)
→ implementation (~~unit, integration, system~~)
→ maintainence & operation.

31-10-23

Software Process Activities.

SRS => System Requirement Specification
SDS => System Design Specification.

V & V

Validation & Verification.

Validation is ~~a~~ testing applied on one test case

Verification is a testing applied for all test cases.

Unit testing.

Integration testing.

System testing.

White box testing:. applied on a programs, a particular piece of code.

Black box testing:. applied on a form, a GUI.

Generic Issues that affect most software.

① Heterogenity: compatibility issues.

② Business & social change:.

③ Security & trust:.

Application types *

① Stand-alone applications: Computer, Laptops. Assistant PDAS runs on a local computer, no need to be connected to a network

② Interactive transaction-based applications.

E-commerce: Buying & selling of goods over the internet

E-business: Day to Day transactions in an organization is termed as e-business. → CRUDS

Palm

Personal Digital Assistant PDAS

PAPERWORK

C R U D | S | - create, retrieve, update, delete, search

(5) Embedded control systems: software control sytms that control & manage hardware devices.

(6) Batch processing systems: process large number of individuals inputs to create corresponding outputs.

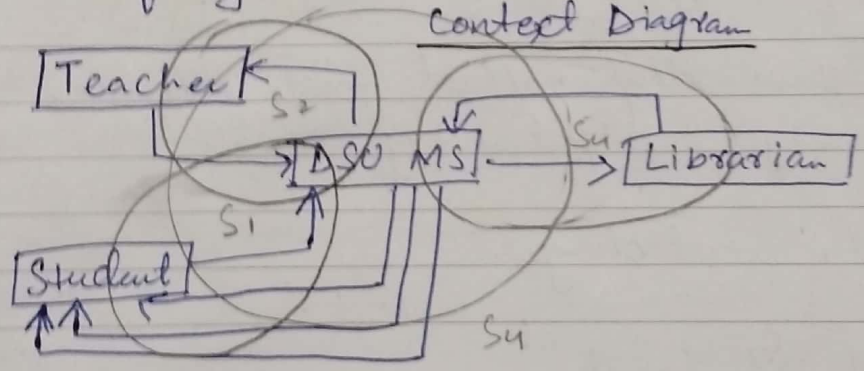
(5) Entertainment systems: personal use, entertain the user.

(6) Systems of modeling & simulation: developed by scientists & engineers to model physical processes & situations

(7) Dat collection systems: set of sensors. collecting data from environment. & send to other systems for processing.

(8) System of systems.

Context Diagram



Software Engineering Fundamentals.

(1) Analyzing the requirements.
(2) Dependability.
(3) Management of Schedule
- Budget — on / under / over
- Schedule — on / behind / ahead

(iv) Understanding software to reuse or build a new one.

Software Engineering and the web 6-11-2023

Chapter 2 - Software Process Models

The software process:

① Specification : SRS SD S. => Documents.
② Design & Implementation. (organization & implementation of the syste
③ Validation. (checking that the system as it is needed).
④ Evolution : (changing the system in response to customer needs)
→ An abstract representation.

Plan-driven, Document-driven, & Agile processes:

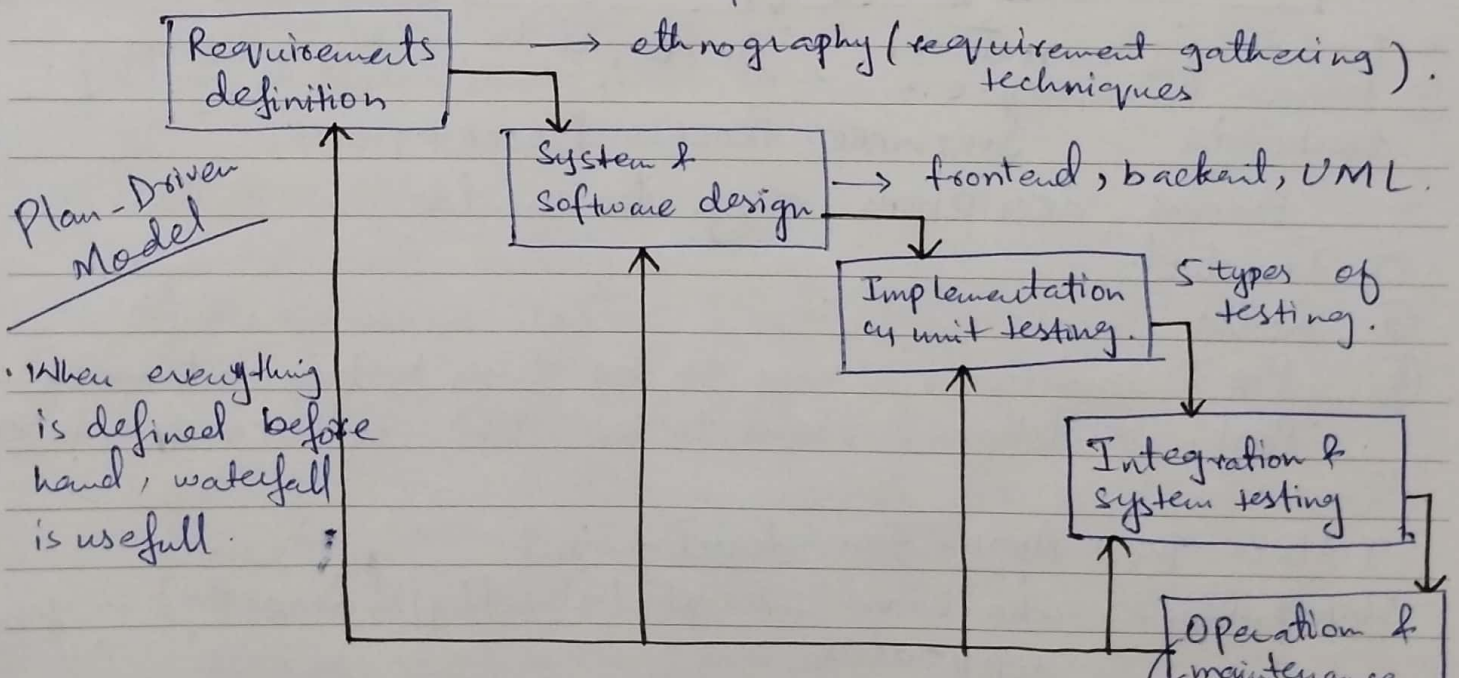
Planned activities in advance.

• No right / wrong software processes.

user-scenario driven planning is increasingly easier to change with customer demands

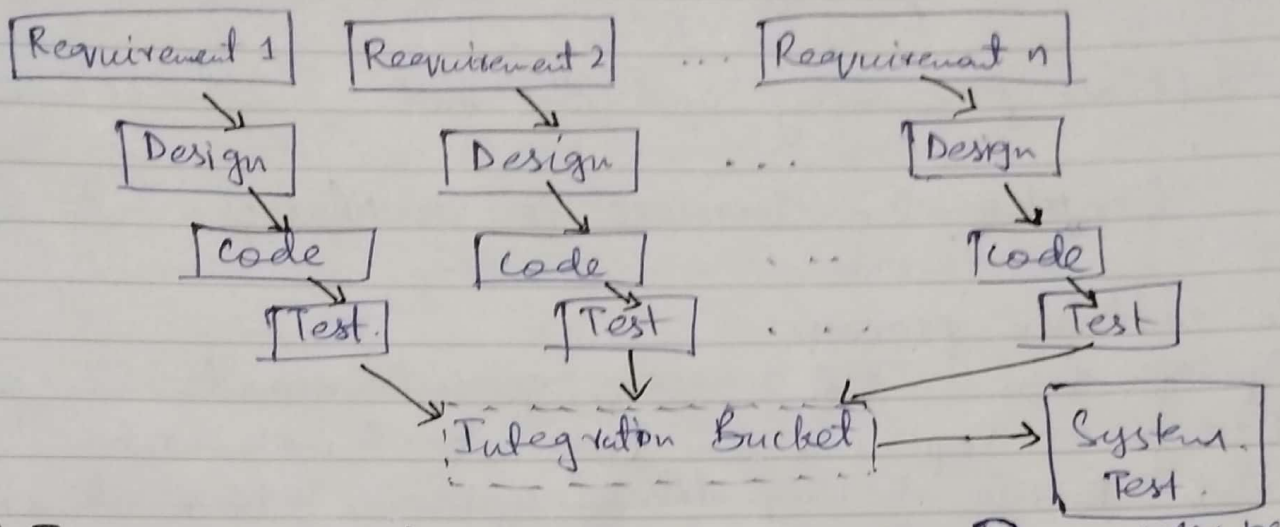
Tool: MS Office Visio.

① The waterfall model: top to bottom.

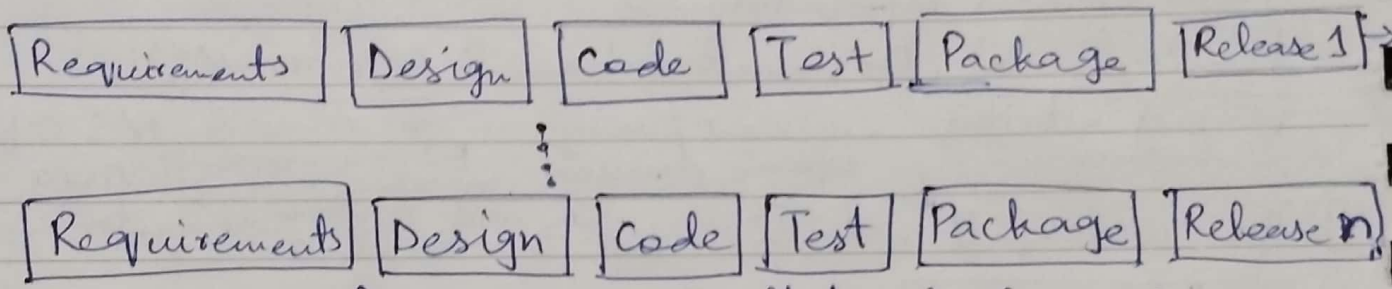


If we get an error in any phase, we have to complete all phases & then we can get back to the errored phase.

(2) ITERATIVE PROCESS MODEL. R D C T



(3) INCREMENTAL MODEL R D C T P. (May be plan-driven or agile)



① Evolutionary in nature ③ difficult to identify overlapping.
② Divide & conque.

overheads:

SOFTWARE PROCESS DESCRIPTIONS.

Process decriptions may also include.

① Product.
② Roles.
③ Pre - conditions : have to be true before a transaction.
Post - condition : have to be true after a transaction.

JAD => Joint Application development.
Stakeholder => who have interest (directly/indirectly) in your application.

7-11-23

INCREMENTAL DEVELOPMENT BENEFITS:

1. cost of accomodating change to customer needs is reduced.
2. easier to get customer feedback on the development.
3. Rapid delivery & deployment of useful software to customer.

COTS
Commercial-off-the-shelf

OOP
OOAD
object oriented Analysis & Design
or
SAD
System Analysis & Design.

(4) REUSE-ORIENTED S.E (Model).

COTS => Commercial-off-the-shelf
↳ reusing existing software.

Process stages.

- The system is assembled from existing components.

SOFTWARE SPECIFICATION

Process of establishing what services are required & what constraints on the system.

Testing.

6) Component Testing
7) Acceptance Testing (Beta Testing).

(5) SOFTWARE PROTOTYPE (MODEL).

prototype is a dummy version of your application.

Benefits.

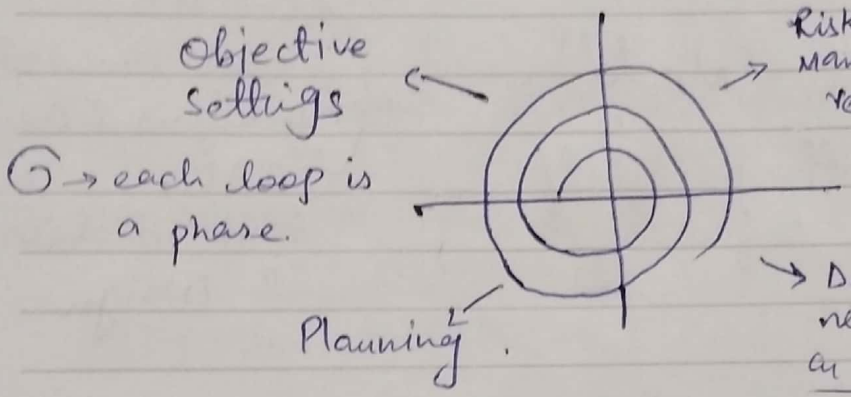
System is more usable.
Design quality is improved.
Maintainable.
Less development effort.
closer match to user's need.

If you fail to plan, then you plan to fail.

THROW-AWAY PROTOTYPE:

You make a prototype, use it then throw it away.
That prototype has no documentation, no coding.

BOEHM'S SPIRAL MODEL:



6 → each loop is a phase.

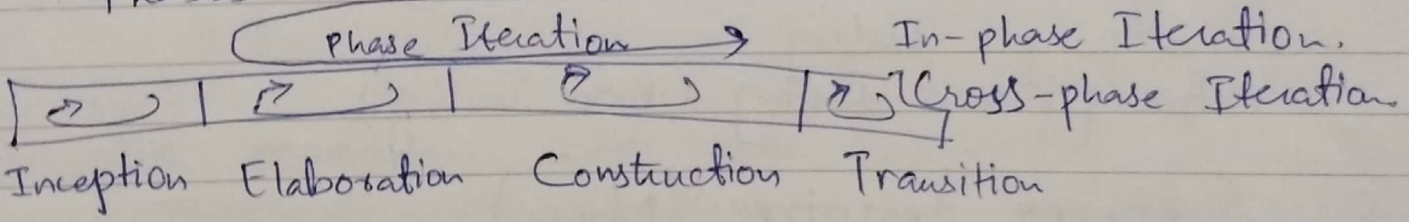
① waterfall
② incremental.
③ iterative.
④ reuse ose.
⑤ COTS.
⑥ Prototype.
⑦ Throwaway Prototype
⑧ Spiral.
⑨ RUP
⑩ Agile

Advantages & Disadvantages of Spiral.

→ customer is involved.
→ Risk management.

THE RATIONAL UNIFIED PROCESS RUP MODEL:

Phases.



Individual phase is in-phase iteration. 4
Overall is cross-phase iteration 1.

CHAPTER 3.

13-11-23

"AGILE SOFTWARE PROCESS MODEL".

Rapid Software Development.

→ Rapid development & delivery is an important requirement for software system

RAD Rapid Application Development

JAD Joint Application Development.

Stakeholder: Anyone who has interest in your project, effect directly/indirectly.

interleaving means alternating b/w two entities

Agile Methods

- focus on code rather than design
- based on iterative approach.
- intended to deliver working software quickly & ensure evolv e
- Agile method is used to reduce overheads (e.g. documentation).

Principles of Agile Method.

① Customer involvement: customer is involved throughout the system development.

② Incremental delivery: software is developed in increments with customer specifying the requirements in each incre waterfall-scrum

③ People not process.

Real time Systems.

↳ Hard RTS (more chances of severeties)

↳ Soft RTS (less ")

XP Practices.

Extreme Programming Practices

Let's eat, children
Let's eat children.

14-11-23

SEMESTER GROUP PROJECT (PART - I)

total 5 parts

① Introduction of the Project: (5 paragraphs) 8-10 lines each paragraphs
Format of Report:
Font Style = TNR Font Size: 12
Line spacing = 1.5 Justified.

② Background.
① Gap Analysis Table (comparison b/w different versions of a project)
② Abstract of Research Papers.
↳ IEEE Xplore
↳ ACM
↳ Google Scholar

③ Work-Flow diagram.
Flow chart of project

④ Aim & Statement of the problem
W6 H Questions.
What When Why How (what model are you using)
Where Who Which

⑤ Methods, Assumptions and Procedures.
One paragraph of each => Structured, Information Engineering or Object Oriented.
4th paragraph => which one are you using.
5th paragraph => why are you using
↳ advantages & disadvantages (which you are using)

⑥ Available Relevant Solutions & Evaluation:
a system's screenshot to write its flaws. (5-8 systems)

⑦ Context Diagram:
→ arrows
* business functions
* transactions

min 10 arrows
Gliffy, Visual Paradigm, draw.io, MS office Visio

Ch. 3 20-Nov-23

Extreme Programming:

Best known & widely used agile method.

Extreme Programming (XP).

extreme approach to iterative development.

↳ more versions of a product

↳ many test cases (an exhaustive process) can be generalized.

XP and Agile principles:

- pair programming.
- refactoring the code.

Extreme Programming Practices:

1. Incremental Planning.
2. Small releases.
3. Simple design.
4. Test-first development
5. Refactoring
6. Pair Programming
7. Collective ownership.
8. Continuous Integration.
9. User - Exceptance Testing.
10. On-site customer.

Requirements Scenarios:

User requirements are gathered as scenarios / user-stories.

customer is a part of XP.

SDLC
System Development Life cycle.

XP and Change:

Refactoring:

improvements in software made.

dll → dynamic link libraries.

Examples of Refactoring:

Reorganization of a class hierarchy to remove duplicate code.

Testing in XP.

Test-first development
Incremental test development.
User involvement throughout testing & validation.

Test-first development:

Writing tests before code clarifies requirements

Customer Involvement:.

User-Exceptance Testing: Testing with dummy data.
maintaining customer involvement is a tedious task.

Junit ⇒ Java unit testing.

Quiz 01 on 23-11-2023.

from Chapter 1.

Time limit = 1 hr , Window = 12 hr.
# of Questions = 01. (Text box).

cocomo khao full marks lao.

21-11-23.

Constructive Cost Model COCOMO.

Adapted from Allan Caine.

COCOMO N Models

MODES	Basic	Intermediate	Detail.
Organic			
Semi-detached			
embedded			

LOC = Lines of Code.
KLOC = Kilo Lines of Code.
= 1000.

	ai	a	b	c	d
Organic	3.2	2.4	1.05	2.5	0.38
Semi-detached	3.0	3.0	1.12	2.5	0.35
embedded.	2.8	3.6	1.20	2.5	0.32.
	-0.2 ↓	+0.6 ↓	+0.08/+0.02 ↓	constant	+0.03 ↓

The Basic Model Of Cocomo.

1. $E = a(KLOC)^b$, LOC will be given in question & KLOC will be determined by ÷ by 100.

Where, E is effort in Staff months.
a & b are coefficients to be determined.

2. $TDEV = c(E)^d$ Project Duration.

Where, T is time
DEV is development } Time for Development.
and E is effort
c & d are constants to be determined.

Average Staff Size.

3. $ASS = \frac{E}{TDEV}$

Productivity.

4. $P = \frac{Size}{E}$, Size will be given in question.

Units of All formulas.

1. Effort = PM / Person-month
   SM / Staff-month
2. TDEV = months
3. ASS = staff.
4. Productivity = $\frac{Size}{E} = \frac{KLOC}{PM}$.

→ formula
→ correct substitution
→ working
→ correct ans.
→ unit

THE INTERMEDIATE MODEL OF COCOMO

① $E = a_i \cdot (KLOC)^b \times C$.

C is the number of attributes multiplied together.

C. { Effort adjustment factor.
Error adjustment factor.
cost driver

② $TDEV = c(E)^d$.

③ $ASS = \frac{E}{TDEV}$.

④ $P = \frac{Size}{E}$

TDEV.

④ $P = \frac{Size}{E}$

Chapter 5 28-11-23

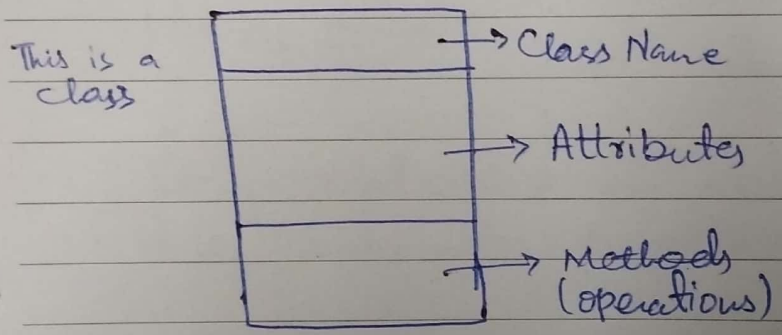
UML Diagrams from the book Kendall & Kendall
Semester Group Project Report (Part 3)

For every arrow in context diagram there will be 10 UML Diagrams.

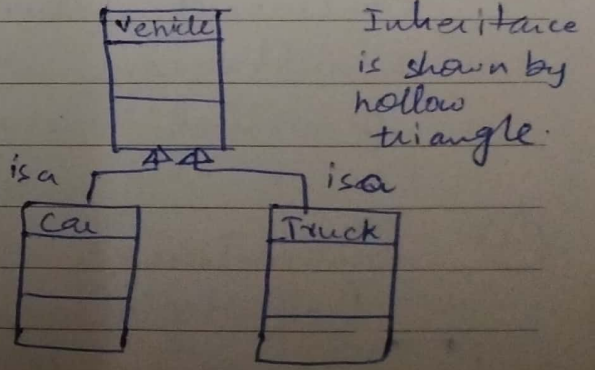
UML : Unified Modeling Language.

① Class Diagram:

Attributes/characteristics/Properties
Methods/Operations/Functions.



Class Diagram:



Class	Class
Parent	Child
generalize	specialize
base	derived
super	sub

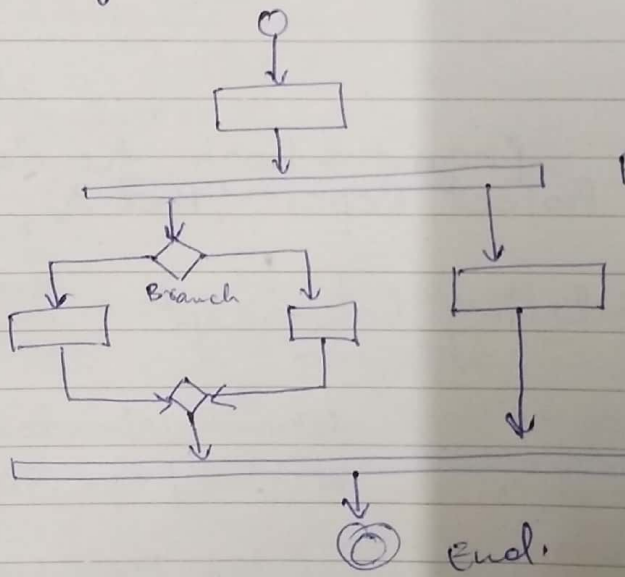
PAPERWORK

② CRC : Class Responsibility Collaborator.

Classname :			
Superclasses :			
Subclasses :			
Responsibility	Collaborators	Object Think	Property
All methods	All associated classes • Inheritance, Association, Composition	Write a sentence of parameters. Ex: I know my integer a.	parameters of a method.

③ Activity Diagram.

A flow chart



Fork. (whenever two things are being splitted)

Join (whenever two things are being joined)

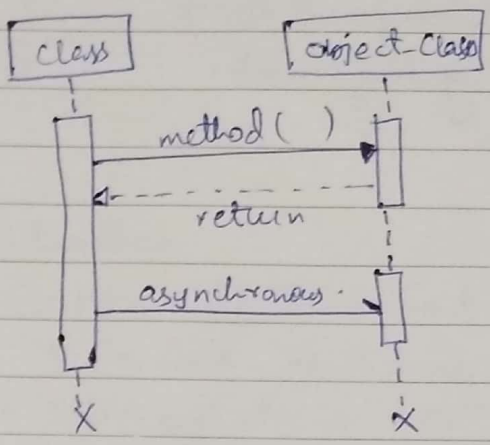
End.

concurrent activities
simultaneous "
parallel "

④ Swimlanes

Activity diagram is divided by colors into compartments (highlighted).

(5) Sequence Diagrams



→ synchronous.

⇢ asynchronous.

two classes are interacting.

organization of objects
time ordering.

(6) Communication Diagram.

- We don't know the order in which activities are performed
- organization of objects.
- methods are numbered.

Extension to Class Diagram.

◆— composition dependent on one another

◇— aggregation independent.

— association.

Every relationship of a class diagram is an association rule but not composition vice versa.

Cardinality : One to Many
Many to One.

@ State Chart Diagram:

State is changing with respect to time.
Object's behavior changing wrt.

MS office Visio
Gliffy
Draw. I.O.
Visual Paradigm.

MID TERM EXAM PAPER PATTERN: 04-12-23

Q1. (a) (b) (c) (d) ~~(e)~~ (20 Marks)
Q2. (a) (b) (c) (d) ~~(e)~~ " Quiz 2.
Q3. " " Ch # 2.
Q4. " " 8-12-23 (Fri
Q5. " " 1 hr
window 12pm

Ch2 : Process Models.

→ Advantages
→ Disadvantages.
→ Diagram
→ Descriptions.

no diagram for spiral model.
for every topic, 4 points ~~marks~~ each.

80% Theory
20% Numerical.

Q5. Cocomo Mode Numerical.

correct formula.
" substitution
" working
" answer
" unit

5-Dec-23

CHAPTER. 4

SRS : System Requirement Specification
SDS : System Design Specification.

SRS.

1. Introduction
   - 1.1 : Purpose of Document : main aim, for whom you're making. (why ?)
   - 1.2 : Intended Audience : layman, anyoneers
   - 1.3 : Abbreviations. DIP → Digital Image Processing. make abbreviations / of big words..
2. Overall System Description.
   - 2.1. Project Background : To what domain your project belongs to
   - 2.2 Project Scope : In-scope : those thing included in system. Out-scope : those thing wont included in system.
   - 2.3 Not of in scope :
   - 2.4. Project objectives : To do this, To do that, To achieve this etc are objectives of the project.
   - 2.5 Stakeholder : Anyone who has an interes, effect, impact in your project directly or indirectly.
   - 2.6 Operating environment : system has to be operational in any operating environment, it should be compatible
   - 2.7 System Constraints : All the issues, hindrances are constraints.
   - 2.8 Assumptions or Dependencies : Assuming that certain requirement our user have . Any task dependent on any ~~previous~~ task.

     containment
     dependancy.
     include relationship.

3. External Interface Requirements.
   3.1 Hardware Interfaces: How are hardware components interacting with one another
   3.2 Software Interfaces: " " software " " "
   3.3 Communication Interfaces: Interactions of Software & Hardware
4. Functional Requirements.
   4.1 Functional Hierarchy:
   4.2 Use cases:.
5. Non-functional Requirements:
   5.1. Performance Requirements: secondary: speed, storage, cost-
   5.2. Safety Requirements: credentials. tangible.
   5.3. Security Requirements: credentials.
   5.4 User Documentation. user manual.
6. Reference: adding reference of your used sources.
   eg. Google scholar, IEEE, ACM.

SDS: ~~Software~~ system. Design Specifications: designer (developer).

1. Introduction.
   1.1 Purpose of Doc. who'll read it, main reason → designer, coder
   1.2. Intended Audience,
   1.3. Project Overvies: abstract of the project.
   1.4. Scope: In scope & out scop-.
2. Design Considerations.
   2.1. Assumptions & Dependencies.
   2.2. Risk & Volatile Areas. RMMM Plan
   Risk Mitigation, Monitoring & Management Plan.

	Risk type	Portability	Effect
1.	Hardware Risk	high	catastrophic
2.	Software "	medium	severe
3.	Employee "	low	insignificant
4.	Technical "		tolerable
5.	Organization "		moderate.

Hardware Risk: 3 examples for each | probability of each example. | effect of each

Software Risk.

Employee Risk.

Technical Risk.

Organization Risk

VIBGYOR.

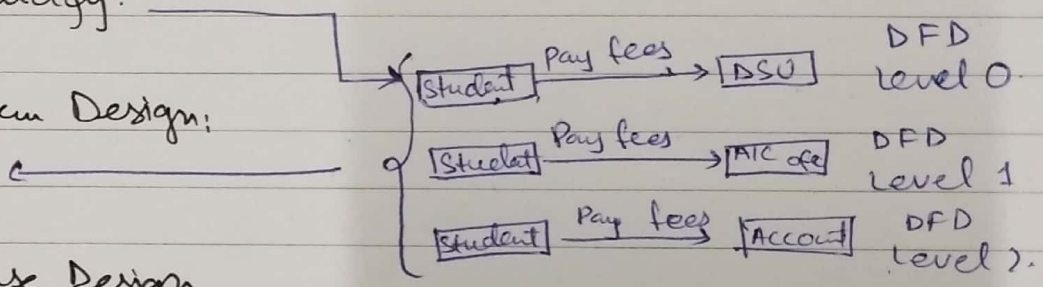
3. System Architecture:

3.1 System Level Architecture: A pictorial diagram showing how things are interacting withe each other -

3.2 Software Architecture: Broken down into different layers e.g OSI (7 layers) model, TCP IP (4 layers) model.

4. Design Strategy:

5. Detail System Design:



6. 5.1. Database Design

5.1.1. ER Diagram.

5.1.2. Data Dictionary: Metadata (Data about data.

§ 5.1.2.1 Data 1

5.1.2.2 Data 2

5.1.2.3 Data n

3
4.8
78.9
77
6.5

where x.y is rounded to the nearest decimal value.

5.2. Application Design Data

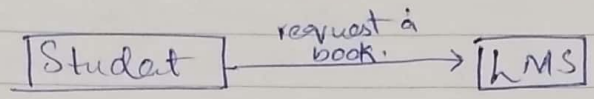
5.2.1 Sequence Diagram.

5.2.2 State-chart Diagram. } UML Diagram.

FINALS 18-12-23

Semester Group Project Report.
(Assignment 02)

Context Diagram.



Filling a table for every arrow of context diagram.

2. Unique Id: 4 digit number 0001 0002 ⋮ 0010
~~Area~~
3. Area: to what domain the area belongs to: arrow coming out from the entity.
4. Actor(s): Entity manupulating with your system. / User.
5. Stakeholder: Anyone who has an |interest| affed |impact on your project directly or indirectly.
1. Usecase name: the name on the arrow, one-line.
6. Desciption: Describe the usecase name.
7. Triggering event: Everything that is thrown at the user.
8. Steps Reaformed. Information for steps function peiforming. requirements for the functions
9. Precondition: Those conditions that have to be true before a transaction occurs.
10. Post condition: Those conditions that have to be true after a transaction occurs.
11. Assumptions: The things you're assuming for an arrow.
12. Success Guarantee: % value of a particular transaction.

13. Requirements Met: - Transactions / Business functions / CRUDS operations

14. Outstanding Issues. Cater on later

15. Priority: High / Medium / Low.

16. Risks: Negative effect. Software, hardware, Technical, employee, organizational risks.

FUNCTION POINT (FP) Nomerical # 02.

19-12-23

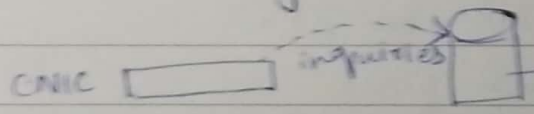
~~Internal~~ Logical Files
~~External~~ Interface Files
~~External~~ Inputs
~~External~~ Outputs
~~External~~ Inquiries.

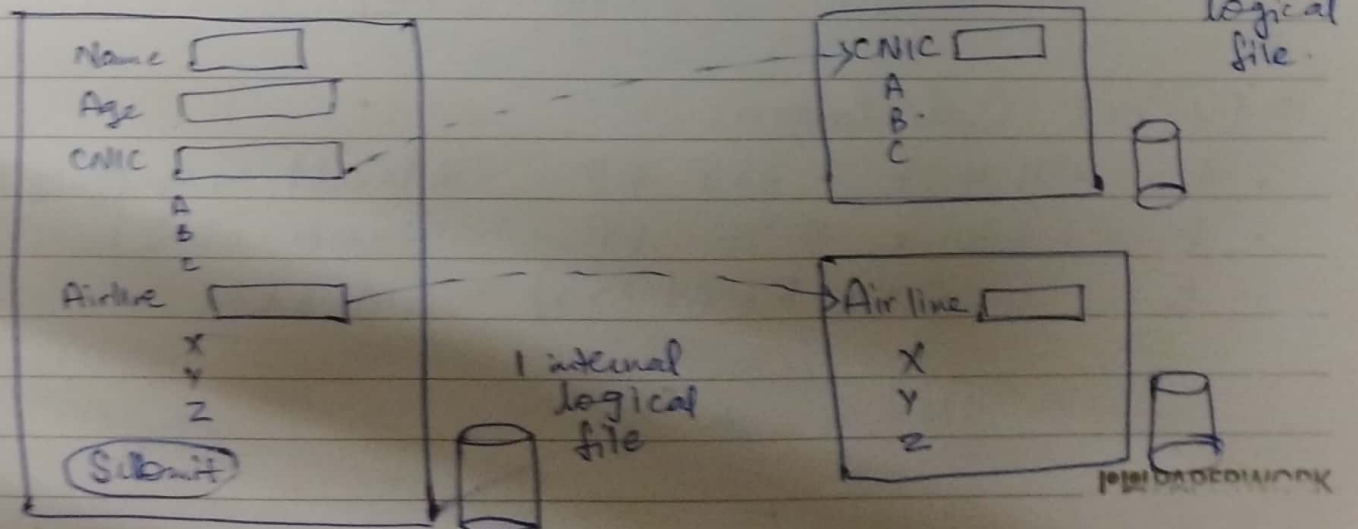
Inputs: 45-50 inputs in exam

Outputs: pop-ups. error messages alerts notifications warnings exceptions message boxes alert boxes help display

45-50

Inquiries
All inputs which are online inputs, are called inquiries
Shown by dotted arrows in diagram.





Submit is an input as well.

Weighing Factor.

Information Domain Value	Count		Simple	Average	Complex	=	
External Inputs	[ ]	x	3	4	6	=	[ ]
External Outputs	[ ]	x	4	5	7	=	[ ]
External Inquiries	[ ]	x	3	4	6	=	[ ]
Internal Logical Files	[ ]	x	7	10	15	=	[ ]
Output logical Files	[ ]	x	5	7	10	=	[ ]
Count total →							[ ]

Inputs:
Simple → entering name
Average → entering CGPA
Complex → entering prediction

Outputs.
Simple → Output in same tab
Average → " " different tab
Complex → " " different window

FORMULAS:

→ Computation:

$$FP = \text{count total} \times [0.65 \times 0.01 \times \sum(F_i)]$$

$\sum F_i$ → There are 14 points, We will rate from 0–5 for each point, then sum it, it will be $\sum F_i$

For exam, take average for each point.

$0 15/2 \Rightarrow 2.5 \Rightarrow 2.5 \times 14 =$ [35] ✓.

→ Effort Calculation:

- $E = -91.4 + 0.355\ FP$ — Albrecht & Gaffney Model
- $E = -37 + 0.96\ FP$ — Kemerer Model.
- $E = 0.054 \times FP(1.353)$ — SMPEEM.

SM is unit of E.

OOAD

Object Oriented ~~Design~~ Analysis and Design.

=> Design Patterns

=> www.dofactory.com ✓

www.webopedia.com ✓

NUMERICAL #03

26-12-23.

PERT

Performance Evolution and Review Technique.

→ Earliest Start Time (EST)
→ Earliest Finish Time (EFT)
→ Latest Start Time (LST)
→ Latest Finish Time (LFT)

Nodes, Edges — 1 start & 1 End node always.
CPM: Critical Point Method: Total of longest path (duration).

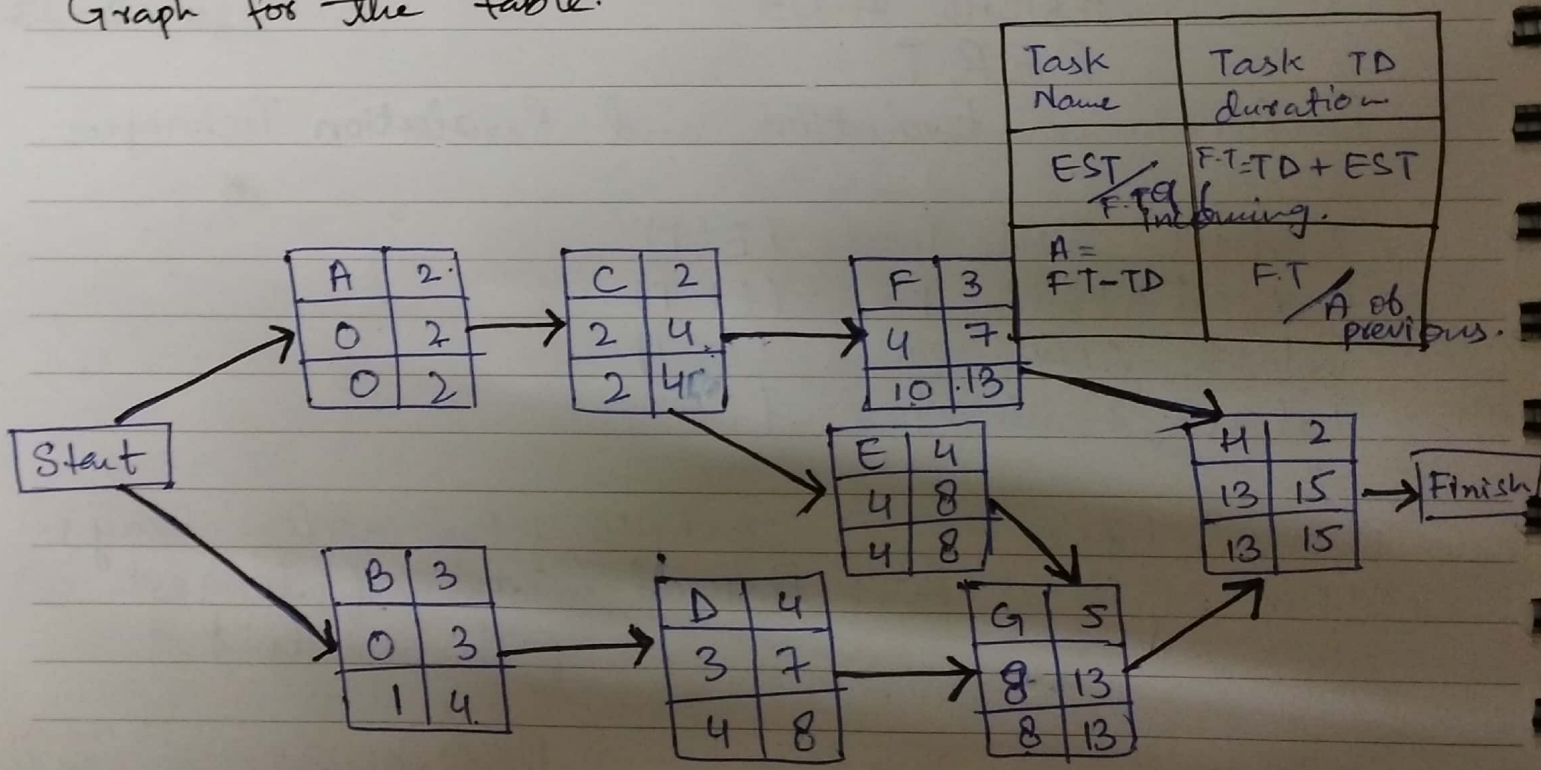
Task	Duration	Predecessors.
A	2	–
B	3	–
C	2	A
D	4	B
E	4	C
F	3	C
G	5	D, E
H	2	F, G.

26 rows, in exam.

EST → Left to right → add EST in Weight
if two arrows, add EST in weight, then select max value.

LFT → Right to left → subtract LFT from weight.
if two arrows, subtract LFT in weight, then select min. value.

Graph for the table.



Task Name	Duation.
EST	EFT
LST	LFT

Key Symbol.

Forward pass direction
max value
add values
2nd row

Backward pass direction.
min value.
Subtract values
3rd row.

Slack = LST - EST
or Slack = LFT - EFT

If Slack = 0, On Critical Path => yes, else => No.

Shade the node which has yes for on critical path.

- The activities on critical path, if they've delayed there will be effect on the whole project's duration
- The activities not on critical path, check its Slack's value, if the delay value of that activity is almost the slack value then there will no effect on the whole project's duration. But if the activity is delayed & it is more than the slack value, then there will be effect on the whole projects duration

2-1-24

SEMESTER GROUP PROJECT (PART-4)

5 → USER MANUAL.

~~10~~ Forms → Screenshot of Forms (GUI's) and give description of each form. (1 paragraph description)

PART-5 Black-box Testing

For every form of assignment 4, make tables.

6 tables, ~~5~~ form = ~~60~~ 30 tables.

Table has 2 columns, 1st column has questions, answer them in the second column, according to your form. → This is black-box testing.

Quiz no. 3, 3rd chapter, 5th January.

CHAPTER: 8 SOFTWARE ENGINEERING. 8-1-24

① Program testing:
- Testing is to ensure, the program is working according to requirements and to discover program defects.
- Testing is an exhaustive process.
- Artificial data is used in testing.
- insertion, deletion & updation anomalies are checked during program testing.
- Presence of error can be identified, not presence (all)
- Verification & Validation.

② Validation & defect testing: ③ ④ verification.
- validation is testing for individual test cases. Making of Product process is going right.
- Verification is a whole. Product is correct.
- Defect testing: one correction of error can correct number of errors.

⑤ Software Inspection:
static, constant

⑥ Software Testing:
dynamic, runtime, changeable.

⑦ Development testing.
Unit testing: testing individual program units.
System testing: all components are integrated & tested as a whole.
Component testing: several individual units are integrated & tested.

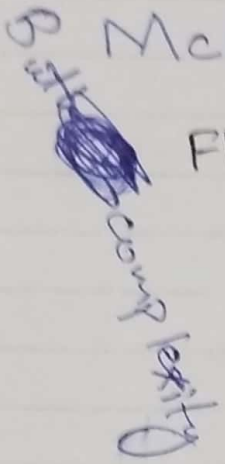
White Box Testing

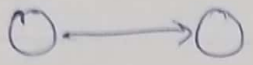
Software Metrics

9-1-24.

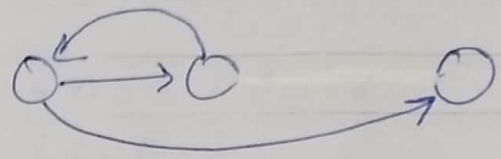
McCabe Cyclomatic Complexity.

Flow graph Notation.

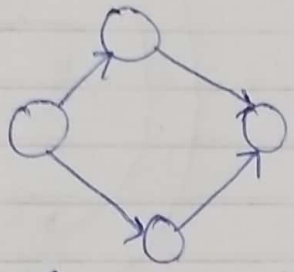




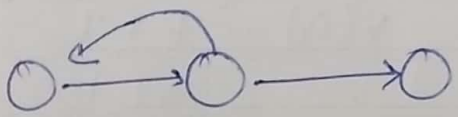
Sequence.
for Sequential Statement.



while / for

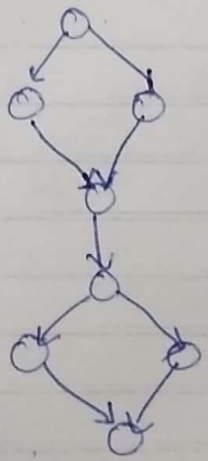


if-then-else.

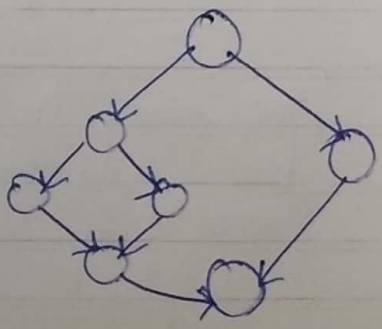


Until / do-while.

```
if ()
{
}.
else
{
}.
if()
{
}
else
{
}
```



```
if()
{ if()
  {
  }
  else
  {
  }
}
else {
}
```



maximum 3 nested if-else ~~with~~ can be give

Cyclomatic Complexity:

- $V(G) = E - N + 2$.

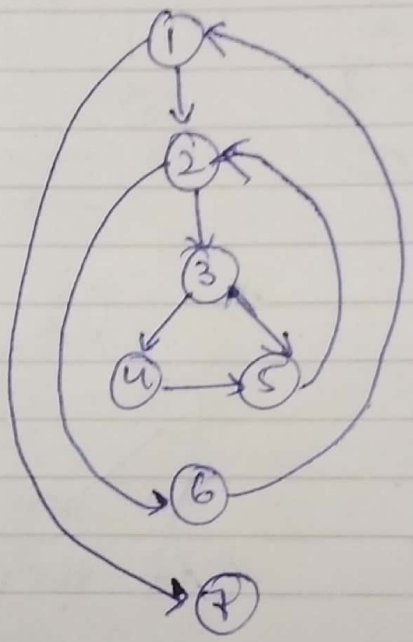
  V → vertix G → Graph V(G) → Cyclomatic Complexity.

  E → Edges N → nodes.

$V(G) = E - N + 2$.

$= 9 - 7 + 2$

$= \boxed{4}$.



- $V(G) = P + 1$

  P → Predicate (conditions).

$V(G) = P + 1$

$= 3 + 1$

$= \boxed{4}$.

- $V(G) = R + 1$

  R → Closed Region.

$V(G) = R + 1$

$= 3 + 1$

$= \boxed{4}$.

The answer 4 tells us that there are 4 possible paths from starting node to ending node.

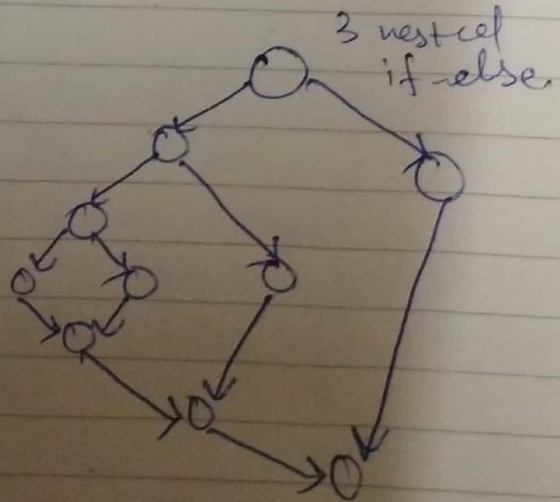
Basis Set

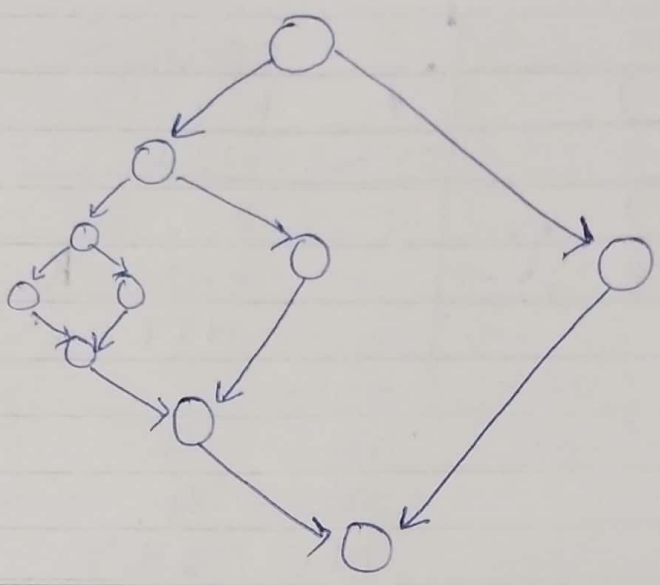
→ 1, 7

→ 1, 2, 6, 1, 7

→ 1, 2, 3, 4, 5, 2, 6, 1, 7

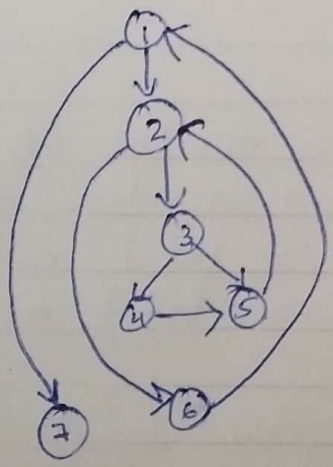
→ 1, 2, 3, 5, 2, 6, 1, 7.





```
if( ) {
    if( ) {
        if( ){
        } else {
        }
    else {
    }
else {
}.
```

```
i = 0;
while(i<n-1) do.
    j = i+1;
    while (j<n) do
        if A[i] <A[j] then
            swap(A[i], A[j]);
    end do;
    i = i+1;
end do;
```



HALSTEAD'S METRICS

7 + 6 + 7 - 3 * 2

↓ operator ↓ operand.

number of operators = 4
distinct operators = 3
number of operands = 5
distinct operands = 4

	operator	operands.
Line 1	(<)	k 2
Line 2	{	NIL
Line 3	(>)	k 3
Line 4	= * ;	x, x k
Line 5	}	NIL.

```
if (k > 2)
{
   if(k > 3)
      x = x * k;
}
```

$n_1 = 9$
$n_2 = 4$
$N_1 = 11$
$N_2 = 7$

n_1 → number of distinct operators
n_2 → number of distinct operands.
N_1 → total number of operators.
N_2 → total number of operands.

Program length : $N = N_1 + N_2$
Program vocabulary : $n = n_1 + n_2$.
Estimated length : $\hat{N} = n_1 \log_2 n_1 + n_2 \log_2 n_2$.

Purity ratio : $PR = \hat{N} / N$.

Volume : $V = N \log_2 n$.

Difficulty : $D = \frac{n_1}{2} * \frac{N_2}{n_2}$

Program Effort : $E = D * V$

$N = 11 + 7 = \boxed{18}$

$n = 9 + 4 = \boxed{13}$

$\hat{N} = 9 \log_2(9) + 4 \log_2(4) = \boxed{36.52}$

$\hat{N} = n_1 \log_2 n_1 + n_2 \log_2 n_2$.

if $n_1 = 17$ and $n_2 = 19$.

$\hat{N} = 17 \log_2 17 + 19 \log_2 19$.

$\hat{N} = 17 \left(\frac{\log 17}{\log 2}\right) + 19 \left(\frac{\log 19}{\log 2}\right)$.

$\hat{N} = 69.48 + 80.71$

$\hat{N} = 150.19$.

$PR = \frac{36.52}{18} = \boxed{2.02}$

$V = 18 \log_2(13) = \boxed{66.60}$

$D = \frac{9}{2} * \frac{7}{4} = \boxed{7.87}$

$E = 7.875 \times 66.60$

$\boxed{E = 524.475}$

Object class testing:

Complete test coverage of a class
- ↳ testing all operations on object.
- ↳ all attributes.
- ↳ testing ~~all operations on object~~ objects in all states.
- ↳ Inheritance, localised, access modifiers.

Automated Testing:

Testing done by a program, made by a human.
It also can have errors.

Partition Testing:

Testing a portion of tests but not equally divided.

Equivalence Testing:

equally partitioned.

16-1-24.

Program.
Validation
Defect
Software
Development
Component
Unit
integration
System
object
class.
automated.
Partition.
Interface
Regression
Release
Performance

Component Testing:
↳ an entity, an API

Interface testing:
① Parameter interfaces
②
③
④

System testing:
units → integration → system

Regression testing:
adding a new functionality into the system, previous functionalities should not be broken. This is regression testing.

Release Testing
small no. of people test it but not developers.

Requirements-based testing:
examining requirements & make tests for it.

Performance testing:

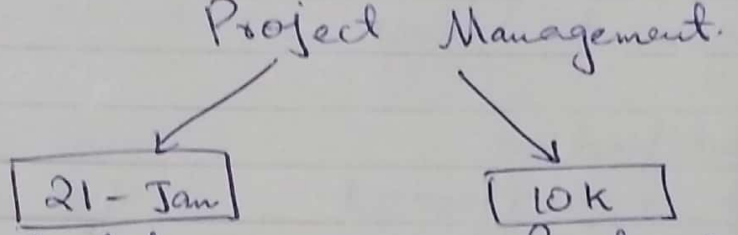
23-1-24

Chapter no. 9 : Project Management

* Risk Management
* Managing People
* Team work.

CH: 21/22 of textbook

Software Project Management:



Risk mitigation monitoring and management Plan

Success Criteria:

Risk Management:

The Risk Management Process
→ identifying the risks
→ drawing up plans to minimise the risks

Risks Mitigation.
100 flaws
↓
20 flaws

→ Risk → probability of some adverse circumstances.
- Project risks
- Product risks
- Business Risks

Risk Type	Probability	Effect
S/w R	High	Catastrophe
H/w R	Medium	Severe
Employee R	Low	Insignificant.
Technical R		Tolerable
Organizational R.		Moderate.

Risk Management Process:

1. Risk Identification:
   - Estimation risk (budget. Estimated cost / Actual cost.
   - Requirements risk. (fault in requirement gathering)
   - Organisational risks
   - People risk
   - Technology risk (some compunet get obsolete).

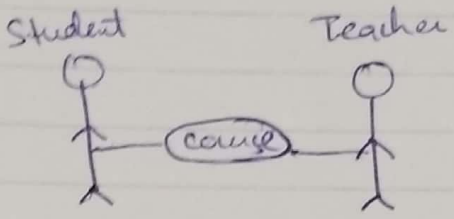
   Examples of different risk types (any 4).

2. Risk Analysis:
   - assesing probability of the risk. High, medium/moderate, low.
   - effects of risks.

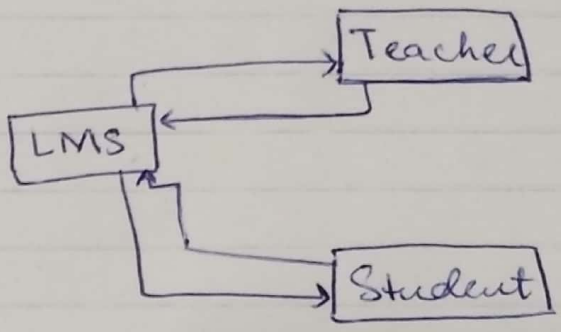
3. Risk Planning:
   Solution to mitigate the risks.
   Strategies.
   - Avoidance.
   - Minimisation
   - Contingency Plans

Strategies to help manage risks (any 4)

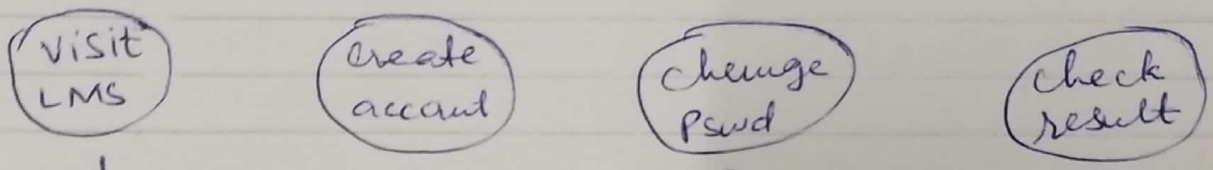
Actor Use Case Diagram



actor use-case diagram.



Context diagram



- In order to check result, we have to perform task 1, 2, 3.
  task 4 extends 1, 2, 3.
  - - - -> dotted arrow.

- Providing food is not functionality of a university. so it extends this functionality.

Chapter 9 (Last Chapter)

29-01-24.

People Management Factors:

Insert anomaly.
Update "
Delete. } Consistency.

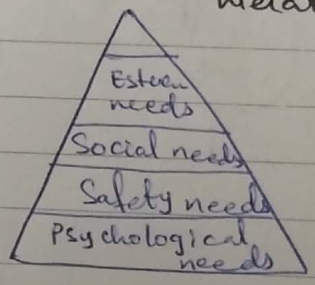
* Consistency (no discrimination).
* Respect
* Inclusion
* Honesty.

Motivating people:

- manager should motivate people.
-

- Motivation is a complex issue.
  1. Basic.
  2.
  3.

Human need hierarchy.

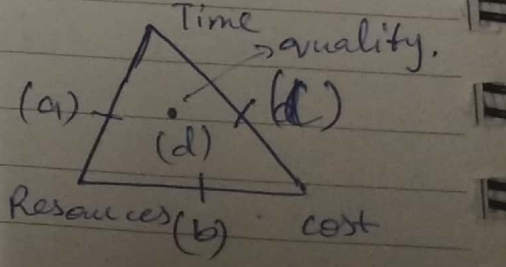


Need Satisfaction:

- Social
- Esteem
- Self-realization.

Personality Types:

* Task Oriented.
* Self oriented.
* Interaction Oriented.



Motivation balance:

Team work:

Group cohesiveness:
3Ps Product, Process, People.

The effectiveness of a team:
Horizontal communication.
Peer to Peer
communication on same level

Vertical communication.
communication b/w different level

Selecting group members:
identify which person is right for the job.

Team Work Imp
Assembling a Team:

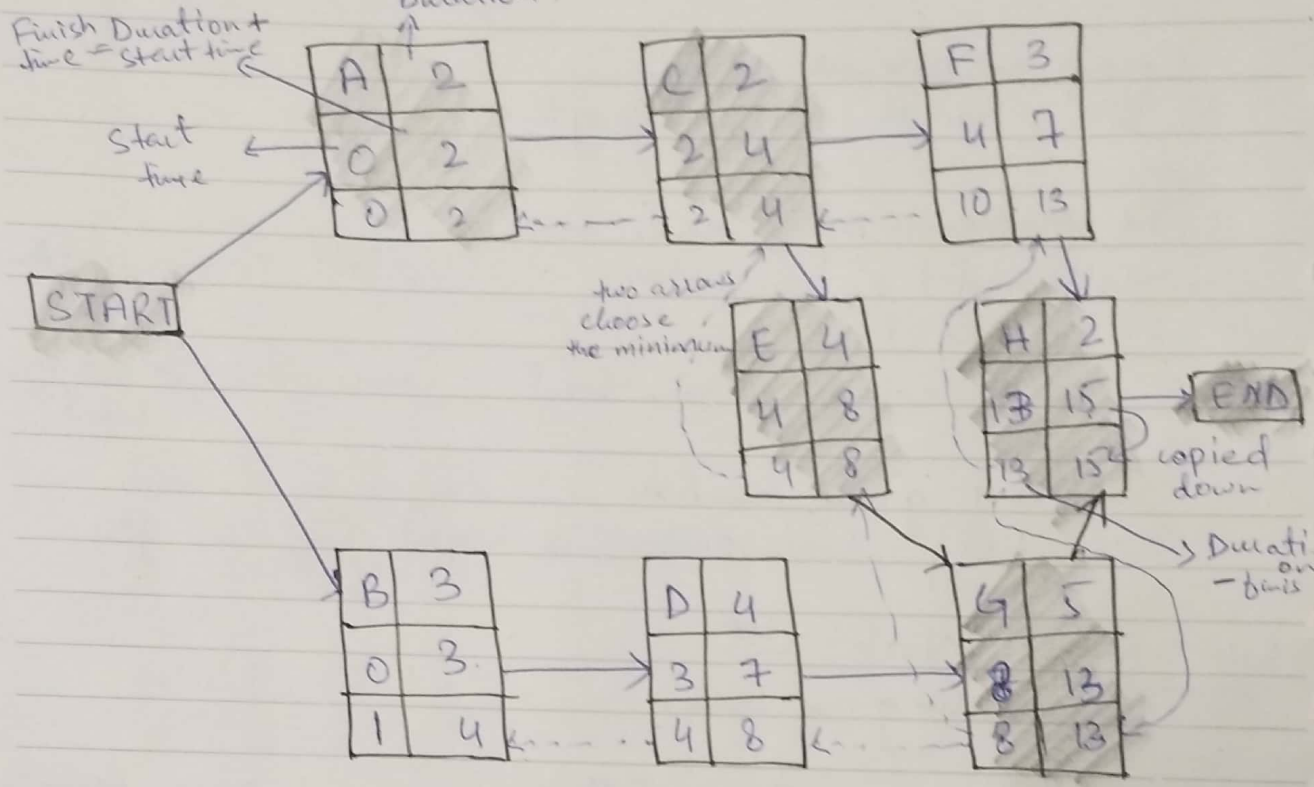
Final Exam Paper Pattern.

Q_1	(a)	(b)	(c)	(d)	(20 marks) → 2 Basic Models.
Q_2	"	"	"	"	" → 2 Intermediate Models
Q_3	"	"	"	"	"
Q_4	"	"	"	"	"
Q_5	"	"	"	"	"

8 40% 20
12 60%

30-1-24

Performance Evaluation and Review Technique.



In exam 26 rows, 26 alphabets in table.

Task	Duration	Predecessors.
A	2	—
B	3	—
C	2	A
D	4	B
E	4	C
F	3	C
G	5	D, E
H	2	F, G.

- finish time of a block will be the start time of next block
- if there are 2 predecessors of one block, tha take maximum finish time of predecessor 4 make it child's start time

Activity	EST	EFT	LST	LFT	Slack LST-EST or LFT-EFT	On Critical Path.
A	0	2	0	2	0	Yes.
B	0	3	1	4.	1	No.
C	2	4	2	4	0	Yes
D	3	7	4	8	1	No
E	4	8	4	8	0	Yes
F	4	7	10	13	6	No
G	8	13	8	13	0	Yes
H	13	15	13	15	0	Yes.

If Slack == 0
On Critical Pat Yes
Else
No.

In digram-
the blocks on critical path, shade them

Lastly, write the nodes on critical path.
Start → A → C → E → G → H → Finish.

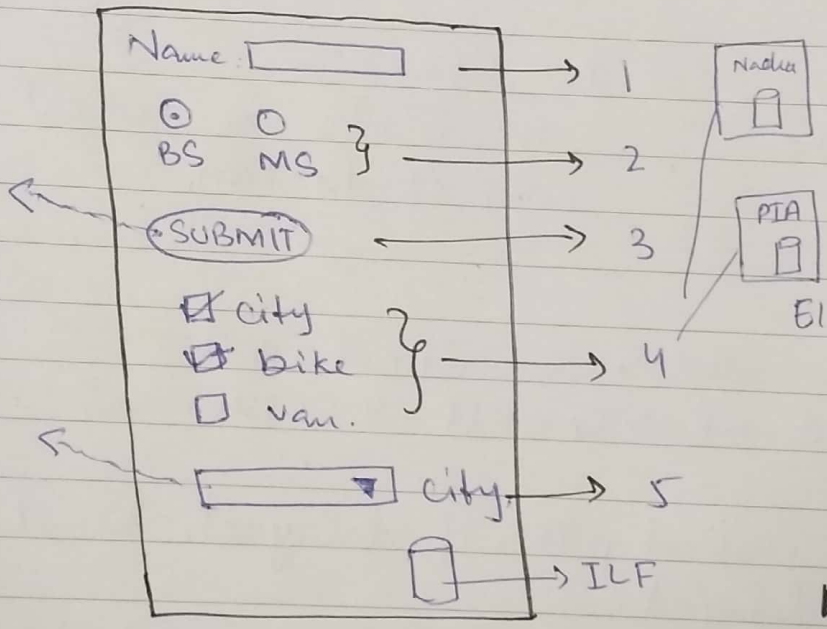
- The activities, on critical path, if delayed, the whole project will be delayed.
- The activities not on critical path, can only be delayed till the slack value calculated, after that it will effect on project

In question the table will be like this (Mazaar the)

Taskname	Pessimistic P	Most likely ML	Optimistic O	Duation	Predeca
A	1	2	3		-
B	4	5	6		A
C	7	8	9		A,B

$$\text{Time Duration} = \frac{P + 4(ML) + O}{6}$$

FUNCTIONAL POINT



Find these words in a paragraph given in question, count & write it in EOs

- error messages
- notifications
- alerts
- popups
- warnings
- exceptions
- dialouge boxes
- message boxes

In exam 40-45 External Inputs.

External Inquiries will be shown in form by dotted arrows

Weighting Factors.

Information Domain Value	Count	x	Simple	Average	Complex	=
External Inputs (EIs)	5	x	3	4	6	=
External Outputs (EOs)	6	x	4	5	7	=
External Inquiries	2	x	3	4	6	=
Internal Logical Files	1	x	7	10	15	=
External Interface Files	2	x	5	7	10	=

Count Total → =

For External Outputs.

ex: error within same window tab (Simple), error another tab (Average)
: error another window (complex).

For External files

if it is 1 → Simple.

" 2 → Average

more than 2 → Complex.

For External Inquiries

in paper it'll be given that it is simple / average / complex.
inquiries are shown by dotted arrows.

Function Point formula

F.P = Count Total × [0.65 + 0.01 × Σ(Fi)].

For ΣFi
there are 14 points
we'll rate each in
range of 0-5.
We'll get 14 values
we'll add all
& it'll be Σ(Fi)

Effort Calculation.

- E = -91.4 × 0.355 (F.P) Albrecht & Gaffney Model
- E = -37 + 0.96 (F.P) Kemerer model.
- E = 0.054 × FP (1.353) SMPEEM.