

C.W  
03/08/2025  
Final

## Topic: 6

### Software Testing

- Tester - SQA engineer
- Purpose - কীভাবে কোর্ট করা (Software কোর্ট করা কীভাবে করা)
- Validation is a type of testing
- ↓  
    real life environmental testing
- Debugging করে Developer
- Verification - Mathematically, testing to prove correctness
- Whitebox test → Developer tests → code available, test from code  
Blackbox test → Tester tests → SRS analysis and test from SRS
- Unit testing → Developer tests
  - method, class, component,  
(loop, function, block etc.)
  - granularity
- Integration Testing → Developer tests
  - Top-down Integration Test
  - Bottom-up

$$\begin{aligned} 80.0 &= (180.0 - 1) \leftarrow 180.0 = 80.0 \times 2.0 \\ 20.0 &= (208.0 - 1) \leftarrow 208.0 = 20.0 \times 10.0 \\ 208.0 &= (211.0 \times 2.0) - 1 \end{aligned}$$

disadvantage  
required lots of stub  
and mocks objects

$$180.0 = (80.0 \times 2.0)$$

$$20.0 = 180.0 - 1$$

$$1.0 = 20.0 - 1$$

$$20.0 = (1.0 \times 20.0)$$

$$180.0 = 20.0 - 1$$

$$189 = 180.0$$

2025

14/08

Sefalo.

CiCD  
DevOps  
GitHub actions

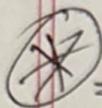
APR (Automated Program Repair)  
Software Testing Domain

Function Testing: অব Req functionality requirement test কৰিব

Req

major function

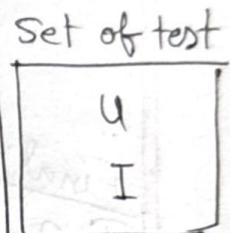
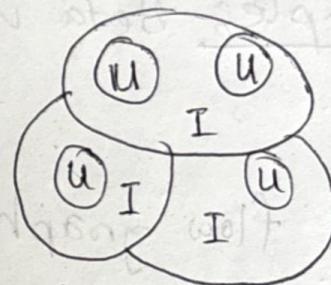
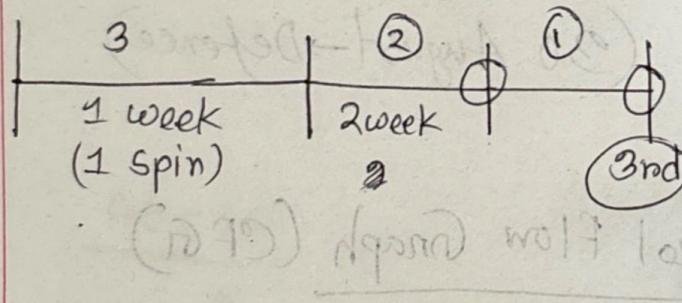
minor function



Regression Testing: নতুন unit add কৰলে বা modify কৰলে যদি সাধের unit bug, prob হয় তাহলে মিটিংয় Test হচ্ছে Regression test (for dependency on else)

U = Unit test case

I = Integration test case



(New Test case)  
(Progressive Phase)  
[plus added new one]  
testing

Regressive test  
(Bucket)  
আগ্রহসমূহ  
testing

Smoke Testing: • test to expose errors

- main service একো প্রকল্পে বা useless হওলে smoke testing কৰা হয়।

System Testing: Testing before handing over to User.

black box

- Security perspective
- Load testing
- Recovery testing

Validation Testing: Tester tests being a real user.

black box

- Tests based on SRS

## □ Acceptance Testing :- User Tests / Client Tests.

- Alpha Testing : User tests and developer is behind
- Beta Testing : ~~client tests~~ Only user and product

Example :- Beta version of games

Final

\* Control flow graph

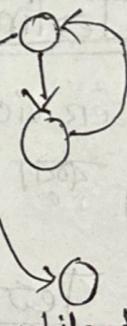
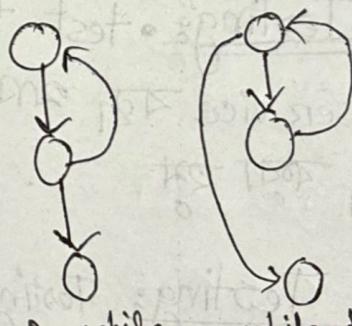
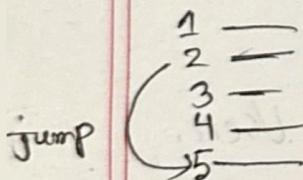
\* Testing

(30 August → Defence)

14/8/25

Control Flow Graph (CFG)

Code → graph



Node → এক/কয়েক স্টেটমেন্ট

Edge/link → এক স্টেটমেন্ট থেকে আরেক স্টেটমেন্ট যাওয়া

Decision node → for, If, Else If

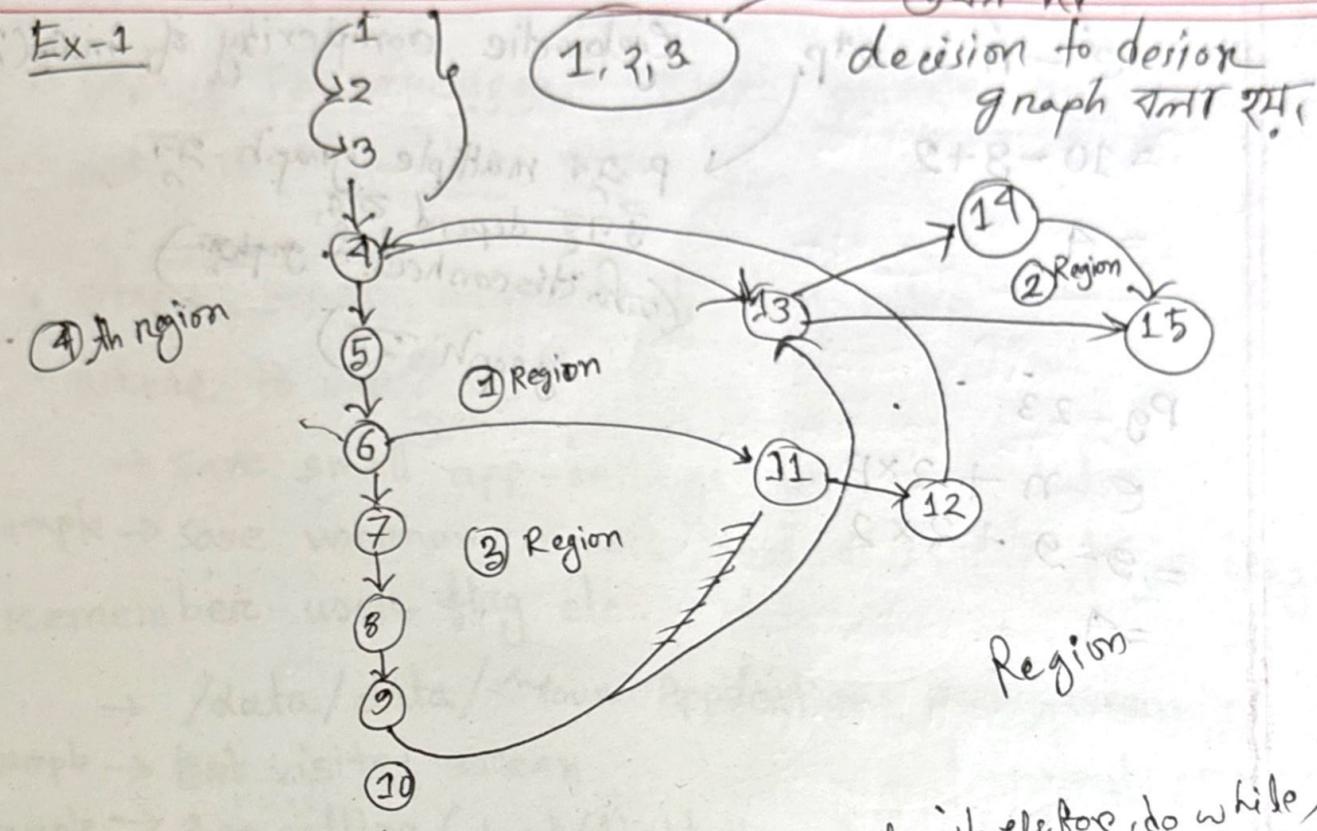
Junction node → 

```
graph LR; N1(( )) --> N2(( )); N1 --> N3(( ));
```

Region →

We can merge sequential node  
to one node

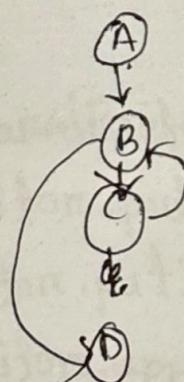
Ex-1



decision flow graph यात्रा ग्राफ़

ABCBD

for



while

if

if

T F

+ F

T F

DABFH

2) ABFGH

w=f, if=t, if=f

3) ABCEBFH

w=f, if=t, if=f

4) ABCDFH

w=t, if=f, if=f

w=t, if=t, if=f

w=f, if=f, if=f

Condition + 1 = Ind. path

Region = Independent path

E-n+2p = Independent path

w=t, if=f, if=f

w=t, if=t, if=f

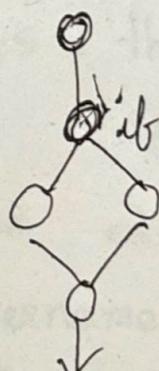
w=t, if=f, if=t

conditional if, else, do while,  
while

\* Independent path

# Region = 4.  
Ind. path = 3 = 1.  
(Pg-11) CFG

~~ABCBD~~



Component  
Class  
function

edge  
node

$$e - n + 2 * p$$
$$= 10 - 8 + 2$$

$$= 4$$

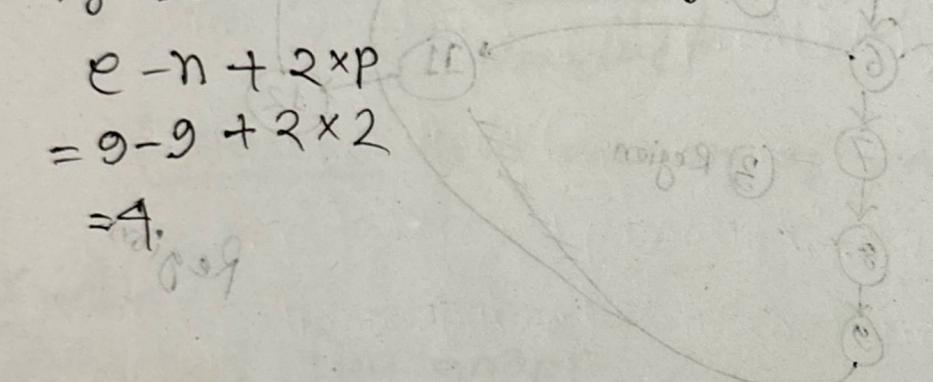
Pg - 23

$$e - n + 2 * p$$
$$= 9 - 9 + 2 * 2$$

$$= 4.$$

Cyclomatic complexity of main( )

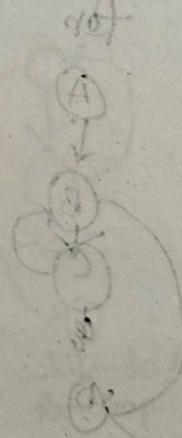
p এর multiple graph এর  
টিগুরু depend রয়ে,  
(যদি disconnected graph)  
graph হল



multiple components  
multiple functions

multiple inheritance

multiple inheritance  
multiple inheritance  
multiple inheritance



T = di, T = di, T = w

multiple inheritance = 1 + no of func

multiple inheritance = no of f

multiple inheritance = no of f - 1

T = di, T = di, T = w

HFBAD

H@IBAC

H@IBAEC

H@IBAEC

H@IBAEC

T T

T T

T T

T T

T T

Sunday  
17.08.2025

DevOps ← Brainstation  
QA ←

## Calculating CSC

### Halstead's complexity:-

$n$  = distinct (multiple = 1)      fixed across

$N$  = total (multiple = multiple)      languages

1 = operator (যেটা operate করে) → action

2 = operand (যাকে operate করা হয় হচ্ছে) → data

→ variable, constant (change কৰা যাবে)

Program Length ( $N$ ) =  $n_1 + n_2$

↓      ↓  
Total      Total  
no. of      no. of  
operator      operand.

Vocabulary length ( $n$ ) =  $n_1 + n_2$

Program Volume ( $V$ ) =  $N \times \log_2(n)$

Program Level ( $L$ ) =  $\frac{n_1}{n_2}$  → complexity ratio মাপা এবং

Program Difficulty  $D = \frac{N_2}{2} \times L$  ~~( $n_1$ ,  $n_2$ ,  $L$ )~~

$$= \frac{N_2}{2} \times \frac{n_1}{n_2} \rightarrow \text{constant}$$

Program Effort,  $E = D \times V$

Time,  $T \propto E = T = \frac{E}{18}$

20.08.25

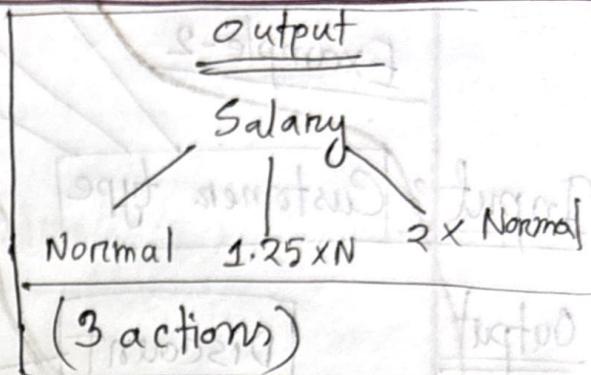
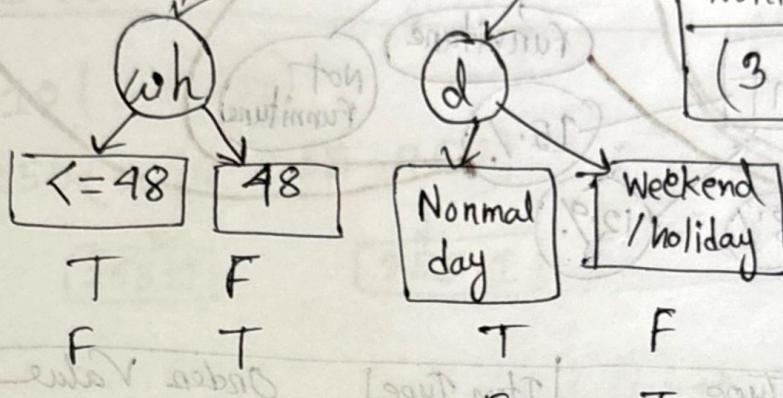
## Data coverage Analysis

Input over condition আসবে  
output " action "

## Decision Testing

Example : 1

Input : ① working hours  
                  ② working days



Condition-1 :  $wh > 48$  I F T T

R1	R2	R3
F	T	1

Condition-2 : ~~w/h~~ d = w/h (Weekend / holiday) F F T

Action-1 : Salary = Normal ✓

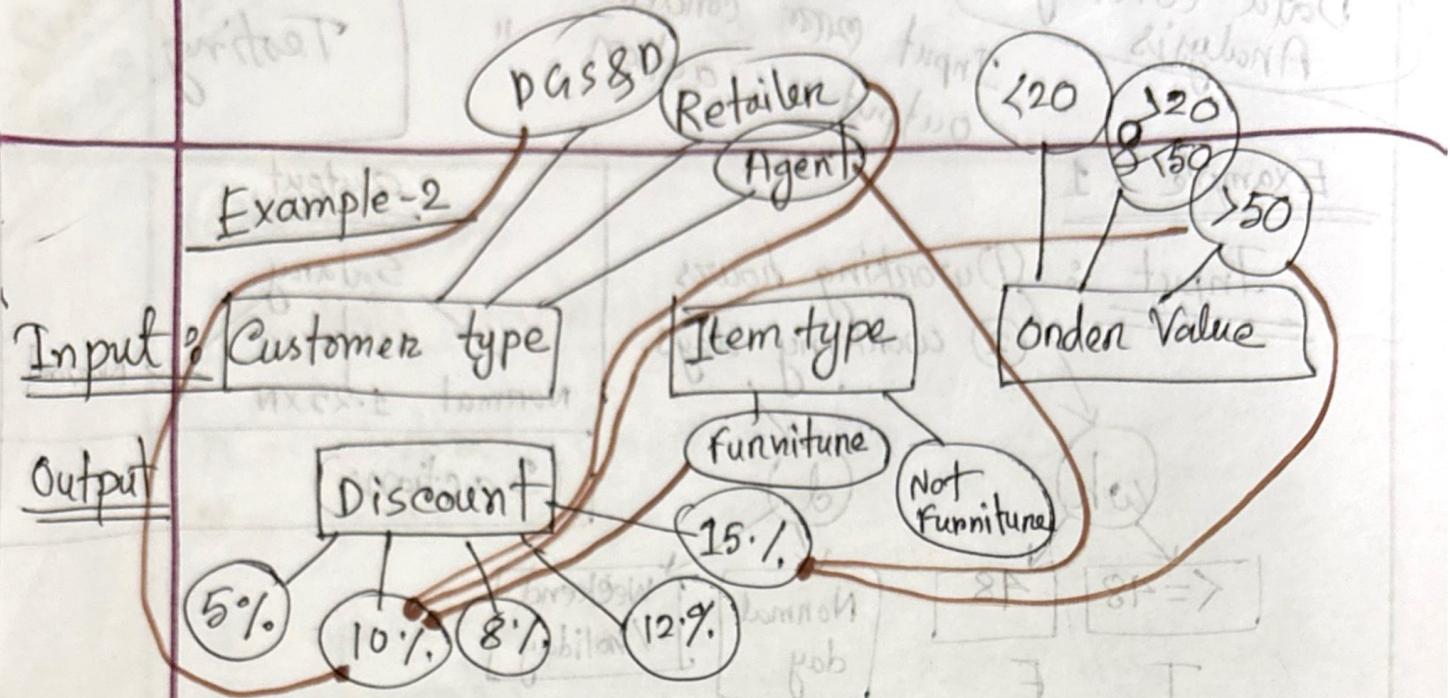
Action-2 : Salary =  $1.25 \times N$  ✓

Action-3 : Salary =  $2 \times N$  ✓

Test case আসবে Rule দ্বারা

Maximum test case  $2^n$ . (for all possible combination)

Minimum test case : No. of rules.



Customer Type	Item Type	Order Value
Retailer   DG58D   Agent	F   NF	<20   >20 & <50   >50
① F   T	I   I	I
② F   NF	I   I	I

I = Immaterial

Test case table of input and output

### Example 3 (Practice)

#### BVA (Boundary Value Analysis)

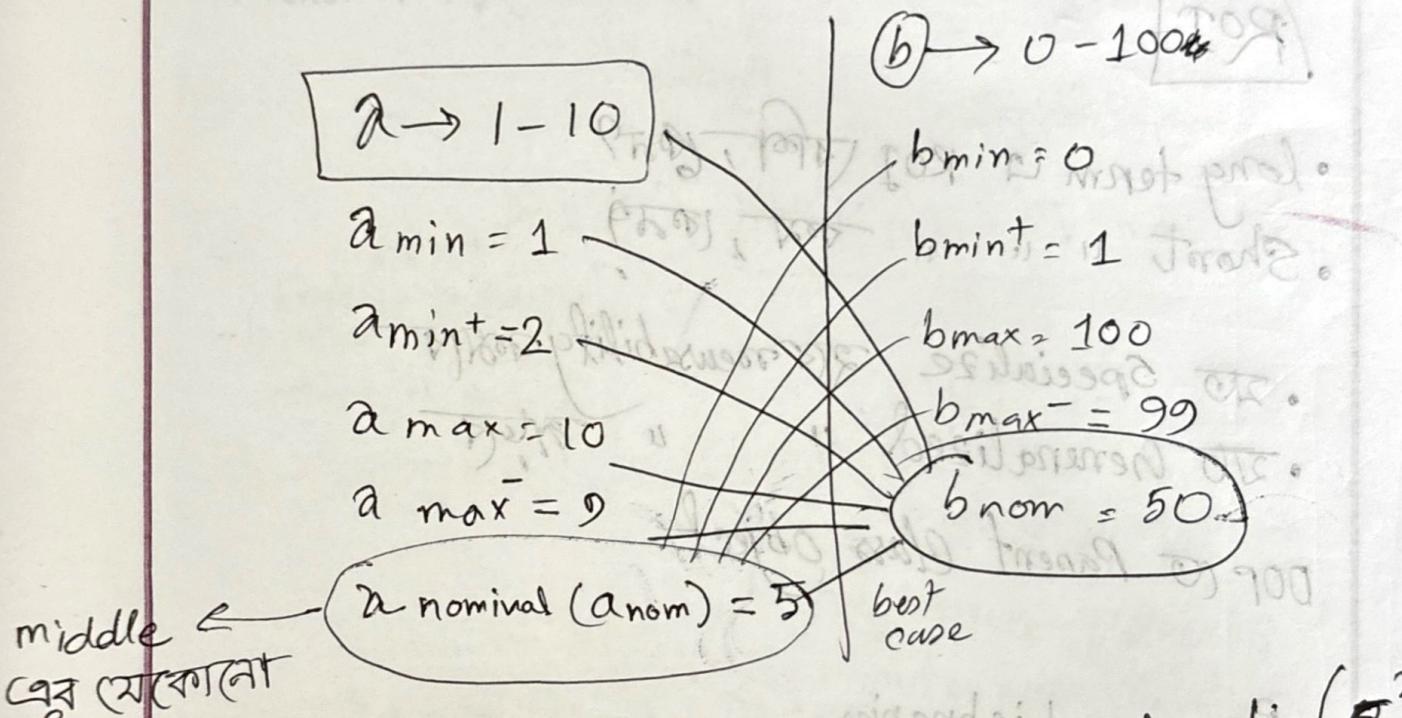
$10(9, 10, 11)$

$255(256, 255, 254)$

$\downarrow$   
 $255+1$

$\downarrow$   
 $255-1$

Inversion



worst case  $\rightarrow$  all possible combination ( $5^n$ )

best case  $\rightarrow$  nominal fix মেঝের মুকোনো  
combination

basic boundary value checking

$4n+1$

(including invalid)  $\rightarrow$  Robustness method ( $\max+$ ,  $\min-$ )  $\rightarrow 6n+1$

Refactoring green  
microservice architecture  
monolithic

## Example-1

test case → Expected output

Quiz 2  
6, 6.1, 6.2

## Software Reuse

### ROI

- long term ROI एक्षि, बोन?
- Short नु त्रिमि रास्ता, कैसे?
- यह specialize और reusability
- यह generalized "

### POD (Parent Class Object)

### Program Libraries

### Design Pattern

Legacy System Wrapping ← two fold.  
ERP System

printed paper preferred print.

item ← (new), (new) bottom count down ← (old), (old)

Springboot

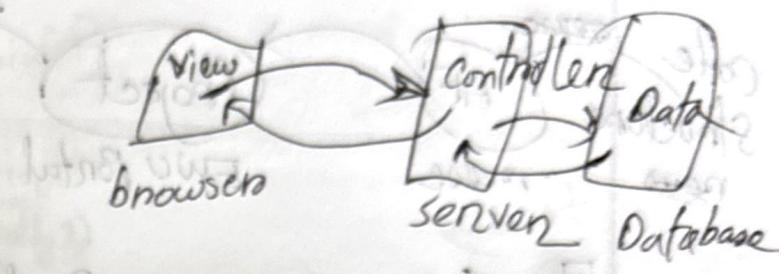
HelloWorld

AJAX

## Application Framework:-

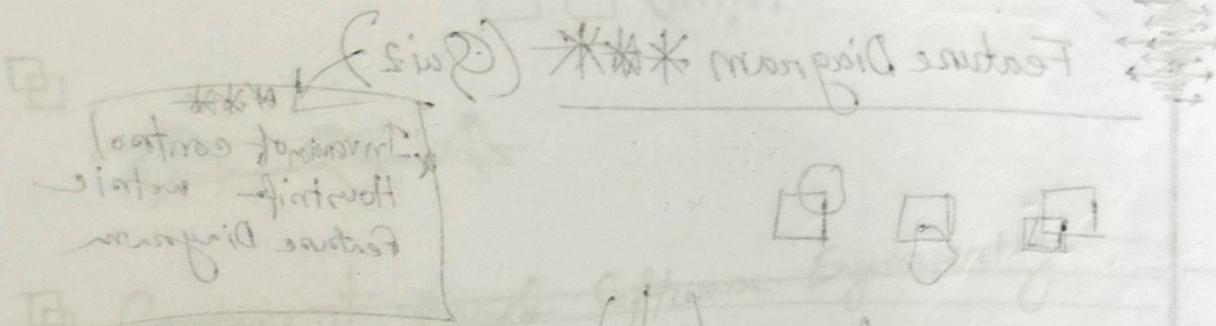
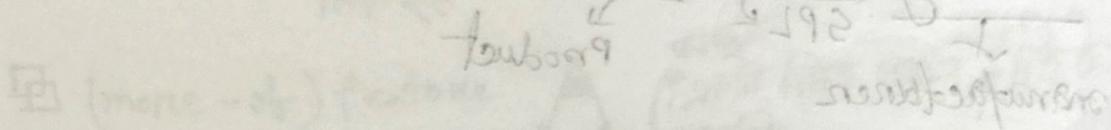
Flutter, Django, etc.

MVC pattern



\* IoC

Inversion of Control

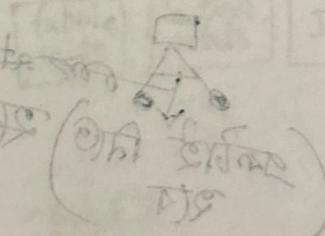


(\*) interface protobuff

(\*) erlang language

interface erlang

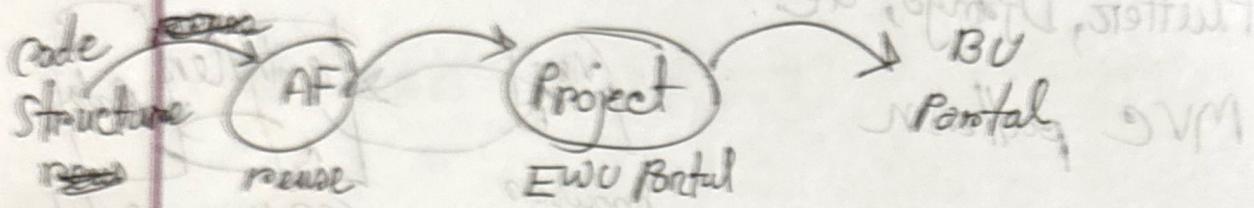
(\*) erlang for erlang



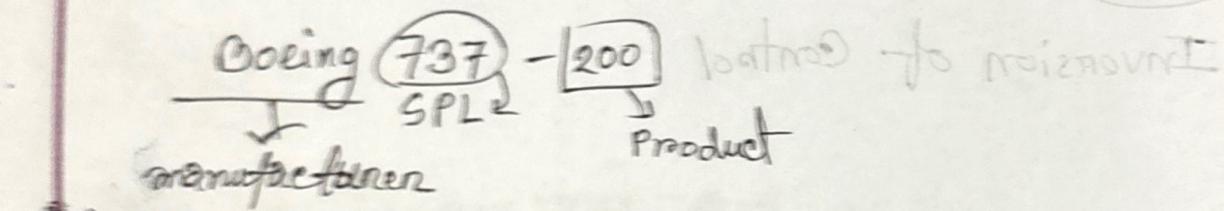
6/10  
24/08/25

1st W breadth first search  
Then, DFS

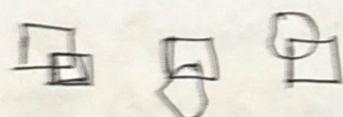
## Frameworks & Software Product Lines



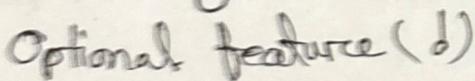
■ SPL = Software Product line



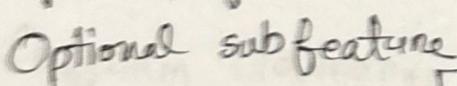
■ Feature Diagram (Quiz)



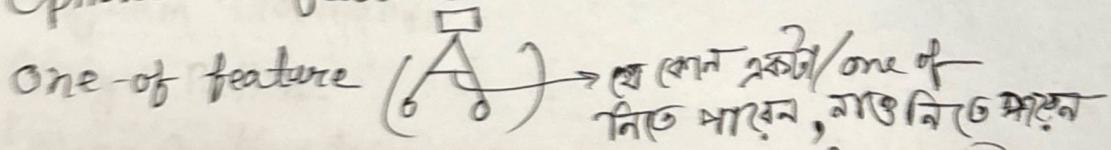
Mandatory feature (!)



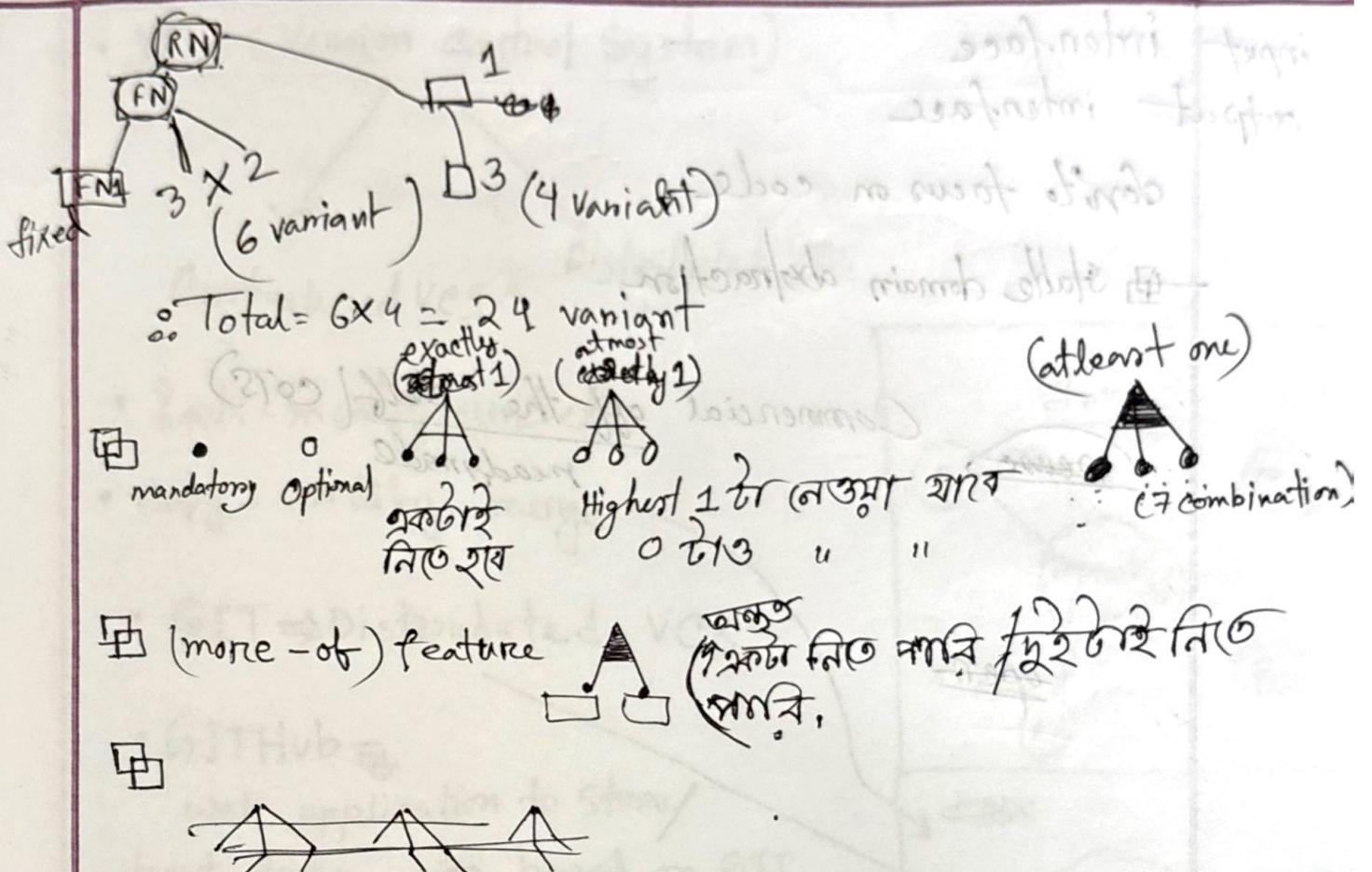
Optional feature (?)



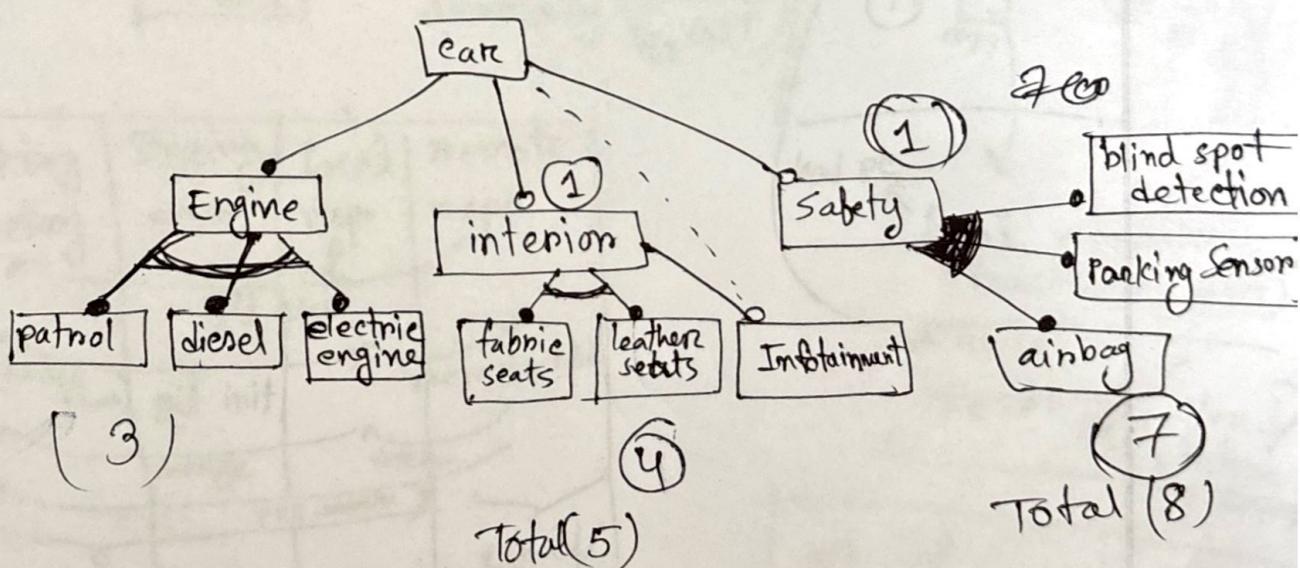
Optional subfeature



One-of feature (A) → একটি/one of  
নিচের ধরণ, একটি নিচের  
ধরণ  
~~(কোটাই নিচে)  
হবে~~ at least one will be  
~~হবে, কিন্তু একটি নিচের  
ধরণ~~



### Component Based Software Engineering :-



$$3 \times 5 \times 8 = 120 \text{ combinations}$$

# Component interfaces:

input interface

output interface

don't focus on code

stable domain abstraction

(in terms)

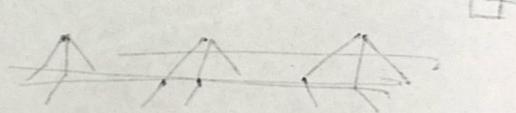
~~reuse~~

Commercial off the shelf (COTS)

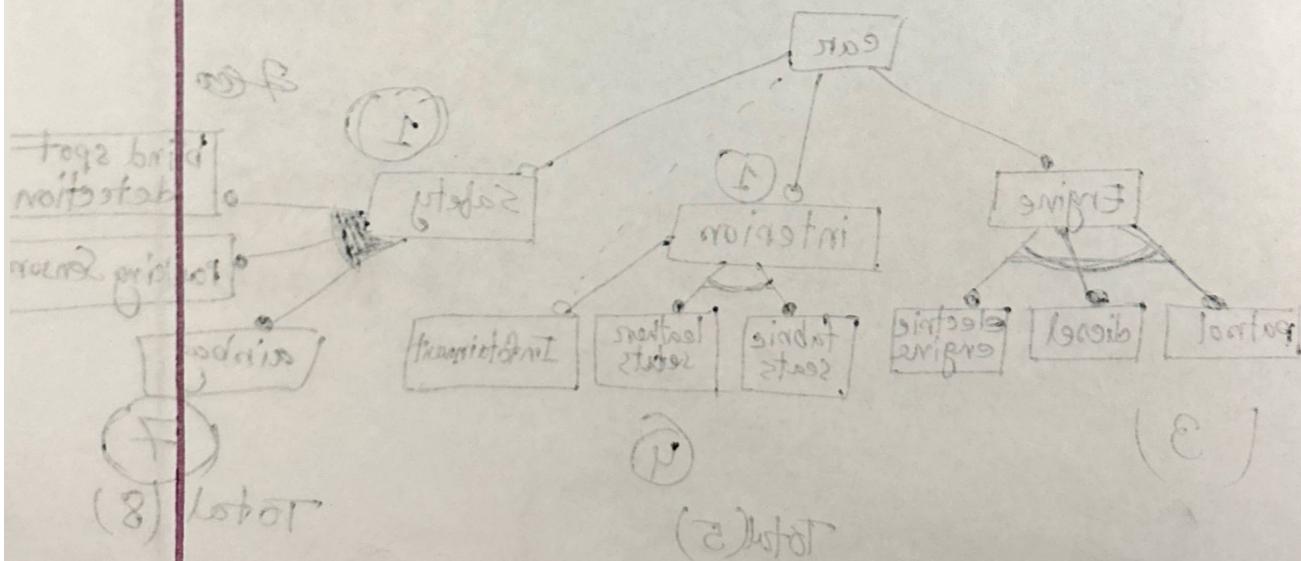
readymade

(commercially available)  $P_S = N \times d = 10 \times 3 = 30$

open source (do-it-yourself)  $N \times d = 10 \times 3 = 30$



principles method based framework



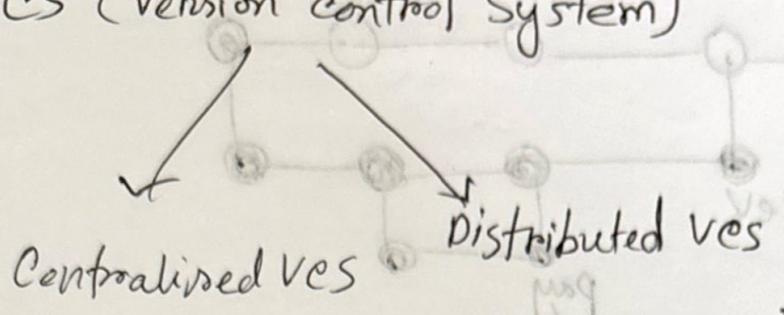
$$\text{number of ways} = 8 \times 2 \times 8$$

CW  
27/08/2025

Open Source software  
commercial "  
Enterprise "

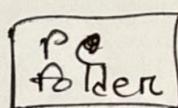
SOA  
microservice-monoly  
Distributed system

## • VCS (Version control System)



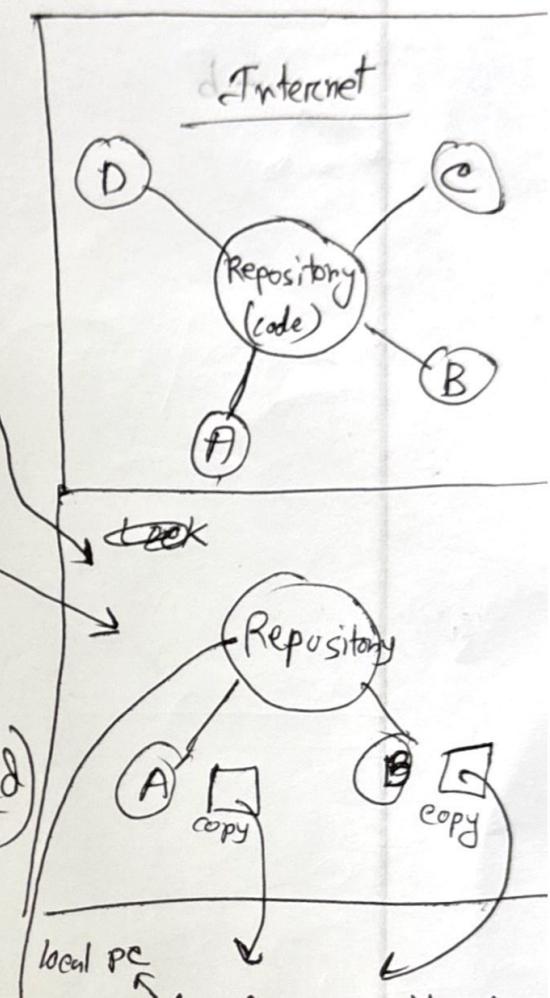
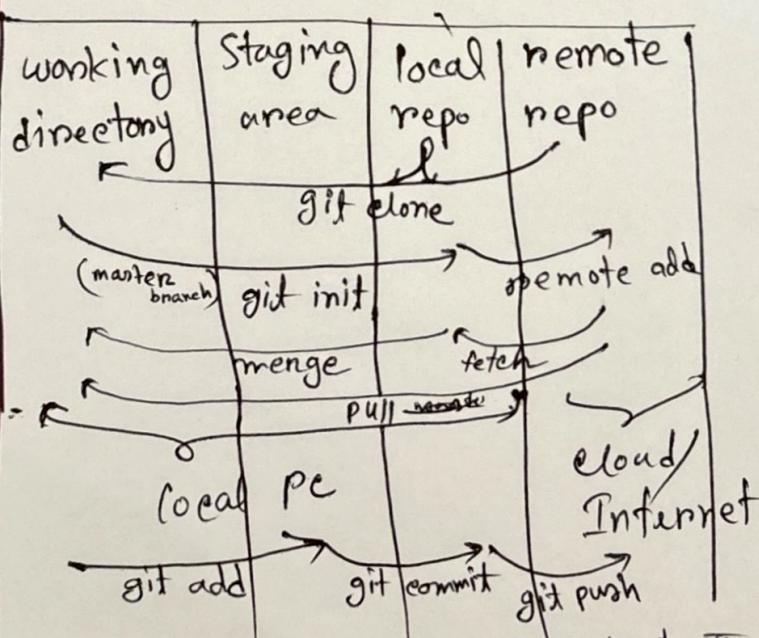
- Lock - modify - unlock
- copy - modify - merge
- GIT  $\Rightarrow$  Distributed VCS
- GITHUB  $\Rightarrow$  web application to store / host code ~~not~~ based on GIT

→ 2nd part

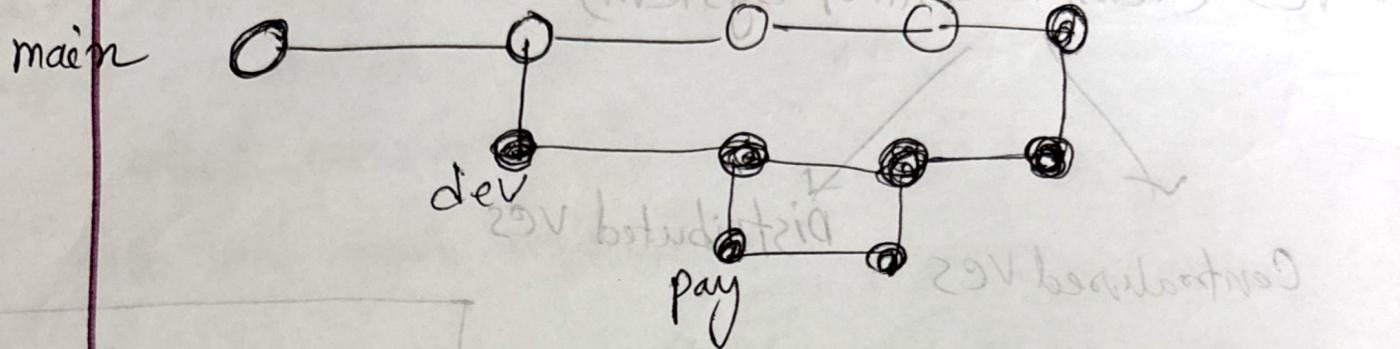


working directory

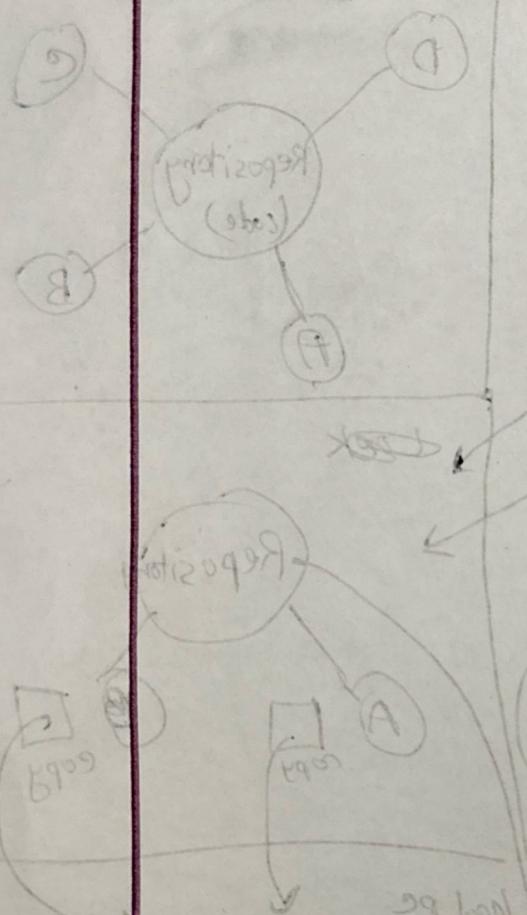
local pc  
(not tracked by GIT)



git init  
git remoteadd origin link  
git status  
git pull origin.  
git branch -r master main



terrorist - b



प्रतिसर्व लोक  
(मैंने यह देखा)  
निम्नलिखि  
(Notes) प्रतिसर्व

tiny tip