

Tuesday

SQA

23-1

- ① Error → Developer mistake
- ② Defect → Error found in testing
- ③ Bug → <sup>Defect</sup>Developer accepted by developer
- ④ Failure

Error  $\xrightarrow[\text{to}]{\text{leads}}$  Defect  $\xrightarrow[\text{to}]{\text{leads}}$  Failure

Testing → Removes bugs

- ① Black box - Tester don't known about inside code
  - ↳ Functional Testing

② White box - Tester tests inside code + functional

+ non-functional testing

→ Retesting vs Regression

T-1-

$$5^2 = 25$$

Tuesday.

S&amp;E

13-2

## Boundary Value Testing

Day 1 - 31  
Month 1 - 12

(values)

- ① Normal Boundary Value Test (Valid values with nominal only)

					Valid		Invalid
					Day	Month	day, m
worst	$x_2$	.	.	.	min	9	
	1	:	(n, min)	.	max	31	12
					nominal	15	6
					min +	2	2
					max -	30	11

- ② Robust BVT (Valid + invalid)

12	.	:		Normal BVT = nominal + <u>within</u>
M	.	.	.	Robust BVT = nominal + <u>within</u>
1	:			+ out of range
	1	D	31	
				Worst BVT = nominal + corner + <u>within range</u>

- ③ Worst Case BVT

12	.	.	.	Robust Worst BVT = All
M	.	.	.	Normal = Nominal <u>only</u> + Valid
1	:	:	:	Robust = Nominal + <u>Valid</u> + Invalid
	1	D	31	Worst = All Valid only
				Robust Worst = All valid + invalid

(49 values)

$7^2 \rightarrow$  for 2 variables

Robust Worst BVI (All valid + invalid)

N	:	:	:	:	Day	Month
M	.	.	.	.	max-min nominal mint-max	max-min nominal mint-max
I	.	.	.	.	nominal	nominal
	1	D	31			

		Input		Day		Month		Input	
Day	Month	D	M	Day	Month	D	M	D	M
min	min	1	1	min+	min	2	1		
min	max	1	12	min+	max	2	12		
min	nominal	1	6	min+	nominal	2	6		
min	mint	1	2	min+	mint	2	2		
min	max-	1	11	min+	max-	2	11		
max	min	31	1	max-	min	30	7		
max	max	31	12	max-	max	30	12		
max	nominal	31	6	max-	nominal	30	6		
max	mint	31	2	max-	min+	30	2		
max	max-	31	11	max-	max	30	11		
nominal	min	15	1						
nominal	max	15	12						
nominal	nominal	15	6						
nominal	mint	15	2						
nominal	max-	19	11						

SQE

$$\begin{array}{|c|c|c|} \hline a & b & c \\ \hline 1 & & \\ \hline \end{array}$$

$$5^3 = 125$$

HomeWork

3 variables

# ① Triangle Problem

$a, b, c$

$$1 \leq a, b, c \leq 200$$

Range

	Valid			Invalid		
	a	b	c	a	b	c
min	1	1	1	min-	0	0
max	200	200	200	max+	201	201
nominal	100	100	100			
mint	2	2	2			
max-	199	199	199			

$\hookrightarrow$

a	b	c	Input			a	b	c	Input		
			a	b	c				a	b	c
min	min	min	1	1	1	min	min	max	1	1	200
max	max	min	200	200	1	max	max	max	200	200	200
nominal	nominal	min	100	100	1	nominal	nominal	max	100	100	200
mint	mint	min	2	2	1	mint	mint	max	2	2	200
max-	max-	min	199	199	1	max-	max-	max	199	199	200
min	min	nominal	1	1	2	min	min	max-	1	1	199
max	max	mint	200	200	2	max	max	max-	200	200	199
nominal	nominal	mint	100	100	2	nominal	nominal	max-	100	100	199
mint	mint	mint	2	2	2	mint	mint	max-	2	2	199
max-	max-	mint	199	199	2	max-	max-	max-	199	199	199

a	b	c	Input		
			a	b	c
min	min	nominal	1	1	100
max	max	nominal	200	200	100
nominal	nominal	nominal	100	100	100
mint	mint	nominal	2	2	100
max-	max-	nominal	199	199	100
min	min	max			
max	min	min	200	1	1
max	nominal	nominal	200	100	100
max	mint	mint	200	2	2
max	max-	max-	200	199	199
min	max	max	1	200	200
min	nominal	nominal	1	100	100
min	max-	max-	1	199	199
min	mint	mint	1	2	2
nominal	min	min	100	1	1
nominal	max	max	100	200	200
nominal	mint	mint	100	2	2
nominal	max-	max-	100	199	199
mint	min	min	2	1	1
mint	max	max	2	200	200
mint	nominal	nominal	2	100	100
mint	max-	max-	2	199	199

			Input		
a	b	c	a	b	c
max- <del>min</del>	min	min	199	1	1
max- <del>min</del>	max	max	199	200	200
max- <del>min</del>	nominal	nominal	199	100	100
max- <del>min</del>	mint max-	mint max-	199	2	2
min	max	min	1	200	1
max-	max	max-	199	200	199
nominal	max	nominal	100	200	199
mint	max	mint	2	200	2
max- <del>min</del>	min	max	200	1	200
nominal	min	nominal	100	1	100
mint	min	mint	2	1	2
max-	min	max-	199	1	199
min	nominal	min	1	100	1
max	nominal	max	200	100	200
mint	nominal	mint	2	100	2
max- <del>min</del>	nominal	max-	199	100	199
min	mint	min	1	2	1
max	mint	max	200	2	200
nominal	mint	nominal	100	2	100
max- <del>min</del>	mint	max-	199	2	199

			Input		
min	max-	min	$i = m$	$199 = b$	$c = 1$
max	max-	max	200	199	200
nominal	max-	nominal	100	199	100
mint+	Max-	mint	2	199	2

## ② Next (Day, Month, Year) Date

	day	month	year		day	month	year
min	1	1	2000	min -	0	0	1999
max	31	12	2024	max +	32	13	2025
nominal	15	6	2012				
mint +	2	2	2009				
max -	30	11	2023				

Day	Month	Year	Input		
			day	month	year
min	min	min	1	1	2000
min	max	max	1	12	2024
min	nominal	nominal	1	6	2012
min	min +	min +	1	2	2001
min	max -	max -	1	11	2023
max	min	min	31	1	2000
max	max	max	31	12	2024
max	nominal	nominal	31	6	2012
max	mint +	mint +	31	2	2001
max	max -	max -	31	11	2023

Day	Month	Year	day	month	year
			a	b	c
nominal	min	min	15	1	2000
nominal	max	max	15	12	2024
nominal	nominal	nominal	15	6	2012
nominal	mint	mint	15	2	2001
nominal	max-	max-	15	11	2023
mint	min	min	2	1	2000
mint	max	max	2	12	2024
mint	nominal	nominal	2	6	2012
mint	mint	mint	2	2	2001
mint	max-	max-	2	11	2023
max-	min	min	30	1	2000
max-	max	max	30	12	2024
max-	nominal	nominal	30	6	2012
max-	mint	mint	30	2	2001
max-	max-	max-	30	11	2023
min	max	min	1	12	2024
max	max				
nominal	max	nominal	15	12	2012
mint	max	mint	2	12	2001
max-	max	max-	30	12	2023
max	min	max	31	1	2024

Day	Month	Year	Day	Month	Year
nominal	min	nominal	13	1	2012
mint	min	mint	2	1	2001
max-	min	max-	30	1	2023
min	nominal	min	1	6	2000
max	nominal	max	31	6	2024
mint	nominal	mint	2	6	2001
max-	nominal	max-	30	6	2023
min	mint	min.	1	2	2000
max	mint	max	31	2	2024
nominal	mint	nominal	15	2	2012
max-	mint	max-	30	2	2023
min	max-	min	1	11	2000
max	max-	max	31	11	2024
nominal	max-	nominal	15	11	2012
mint	max-	mint	2	11	2001
min	<del>min</del>	max	1	1	2024
max-	max-	<del>max</del> nominal	30	11	2024
nominal	nominal	<del>max</del>	15	6	2024
mint	min+	max	2	2	2024
max	max	min	31	12	2000
nominal	nominal	min	15	6	2000
mint	min	min	2	2	2000
max-	max-	min	30	11	2000

Day	Month	Year	Input		
			Day	month	year
min	min	nominal	1	1	2012
max	max	nominal	31	12	2012
mint	mint	nominal	2	2	2012
max-	max-	nominal	30	11	2012
min	min	mint	1	1	2001
max	max	mint	31	12	2001
nominal	nominal	mint	15	6	2001
max-	max-	mint	30	11	2001
min	min	max-	1	9	2023
max	max	max-	31	12	2023
nominal	nominal	max-	15	6	2023
mint	mint	max-	2	2	2023

Friday

SQA

16-2

Selenium - Web app black box tester

- Automatically tests the test cases
- Chrome extension

Tuesday

SQE

20-2

## Day 1 - 31 Equivalence class Testing

Month 1 - 12

$c_1$  Normal

Day Month

Robust

Black Box Testing.

Weak

=

Strong

Weak

=

Strong

$c_3$	$c_1$	$c_2$
$c_1$	$c_3$	$c_3$
$c_1$	$c_1$	$c_2$
$c_1$	min	max
$c_2$		
	$c_1$	$c_2$
	$c_3$	


Valid  
Invalid

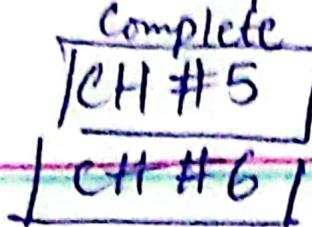
$c_1 > c_2 > c_3$




- Put up any random values inside valid range for testing

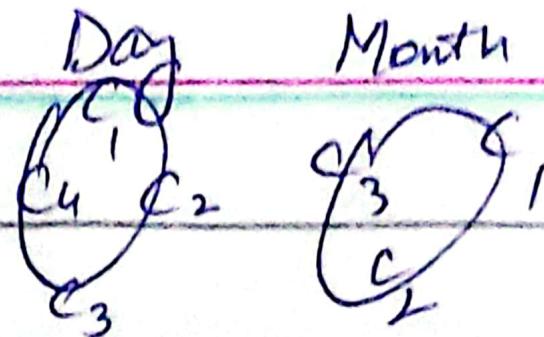
# Software Testing - A - Craft

2.4) 2.5



Day 1 - 31

Month 1 — 12



~~Step 1~~ Create classes of variable

Day

Month

$$\underline{DAYC1} = 1-28 \text{ [Valid]} \quad MC1 = 1-12 \text{ [Valid]}$$

$$\underline{DAYC2} = 29-31 \text{ [Valid]} \quad MC1 = 1 > (\text{smaller}) \text{ [Invalid]}$$

$$\underline{DAYC3} = 31 < (\text{greater}) \text{ [Invalid]} \quad MC1 = 12 < (\text{greater}) \text{ [Invalid]}$$

$$\underline{DAYC4} = 1 > (\text{smaller}) \text{ [Invalid]}$$

# SQE

## Home work

### 2.4 The Commission Problem

This problem separates into 3 parts

① Input data validation

② The sale calculation

③ The commission calculation portion

Pseudo code

lockPrice = 45.0      totalLocks = 0

stockPrice = 30.0      totalStocks = 0

barrelPrice = 25.0      totalBarrels = 0

// Input

while (locks < 0)

    input (stocks, barrels)

    totalLocks = totalLocks + locks

    totalStocks = totalStocks + stocks

    totalBarrels = totalBarrels + barrels

    input (locks)

Output (" print (' Locks sold: ', totalLocks) )

print (" Stocks sold : " totalStocks )

print (" Barrels sold : " totalBarrels )

lockSale = lockPrice \* totalLocks

stockSale = stockPrice \* totalStocks

barrelSales = barrelPrice \* totalBarrels

Sales = tankSales + stockSales + barrelSales

Output.print ("Total sales : ", sales)

if (sales > 1800)

commission = 0.10 \* sales      // 180.

commission = commission + 0.15 \* 800

commission = commission + 0.20 \* (sales - 1800)

elif (sales > 1000)

commission = 0.10 \* sales

commission = commission + 0.15 \* (sales - 1000)

else

commission = 0.10 \* sales

print ("Commission is \$", commission)

Fridays

SQE

23 - 1

- Make Test cases of Equivalent classes

D	M	Y
---	---	---

Normal =  $2 \times 2 \times 2 = 8$  cases if all then strong Normal

Robust =  $4 \times 4 \times 4 = 64$  cases if all then strong Robust

→ Test cases are must else weak —

Tuesday

SQE

27-2

FKN

## Decision Table based Testing

Condition

!Δ

Already False Combinations

Don't Care

!triangle  
 $a \neq b$

F	F	F	F	F	F	F	T	T	T	F	T	T	T	T
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

$a = c$

F	F	T	T	F	F	T	T	T	T	F	T	T	T	T
F	T	F	J	F	T	F	T	F	T	F	T	F	T	T
F	T	F	J	F	T	F	T	F	T	F	T	F	T	T

$b = c$

F	T	F	J	F	T	F	T	F	T	F	T	F	T	T
F	T	F	J	F	T	F	T	F	T	F	T	F	T	T
F	T	F	J	F	T	F	T	F	T	F	T	F	T	T

!Δ

*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

EQ

*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

ISO

*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

SE

*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

Action

Impossible

- OR -

Expected output \*

\* \*

Thursday

SGE

29-2

## Decision Table based Testing

Condition	Rules	< can be many >	< add as many as possible
Action	Expected output		

M1 : month has 30 days → c1

M2 : month has 31 days → c2

M3 : month is Feb or 2 → c3 leap

D1 :  $1 \leq \text{day} \leq 28$  → c4 | Y1 = year is leap year

D2 : day = 29 → c5 | Y2 = year is not a

D3 : day = 30 → c6 | leap year → c8

D4 : day = 31 → c7

Tuesday

SQE

19-3

## Software Metrics

Metric → combination of 2 or more than 2 quantities

Quantities

### Availability

Mean time to failure

Mean time to failure + Mean time to repair

$$\uparrow \alpha \propto \frac{1}{MTTR} \quad MTTF \downarrow$$

MTTF	MTTR
Mean time to failure	Mean time to repair

$$\uparrow \alpha \propto MTTF \uparrow$$

Maintainability  $\propto \frac{1}{MTTC}$  → Mean time to change

Defect Density → used to calculate Correctness

DD = No. of defects

Lines of code (LOC) → kilo Lines of code

Usability → Depends on various factors

Integrity → Depend on attack (can be any)

Integrity =  $\sum [1 - \text{threat} + (1 - \text{security})]$

Integrity =  $\# \text{ of successful attempts} / \# \text{ of attempts}^{\text{total}} \times 100$

### Performance

- ① Latency
- ② Throughput
- ③ Jitter

↳ Response time

1st time delay -  
2nd time delay

Friday

SQE

22-3

## Cyclomatic Complexity

Path Testing  
white Box

$$V(G_i) = E - N + 2$$

where  $E \rightarrow$  edges

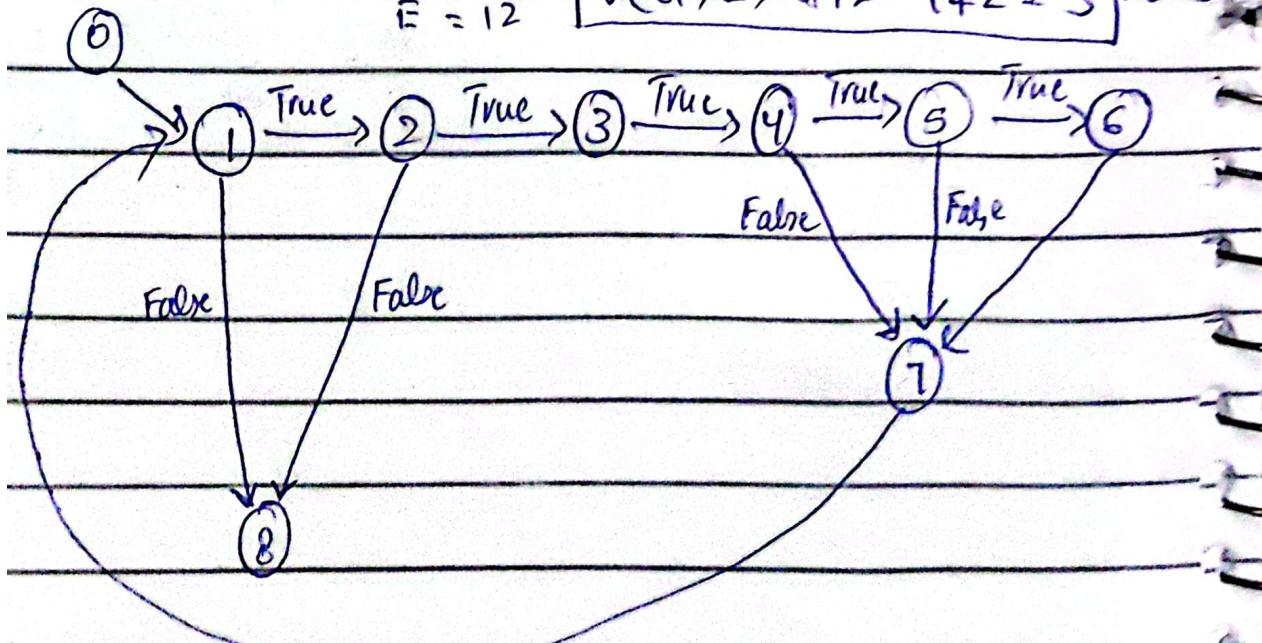
$N \rightarrow$  nodes

① // — ②  
while ( value[i] != -999 && i < 100 )  
{  
    ③  
    ④  
    if ( value[i] >= min && value[i] <= max ) {  
        ⑤  
        { tovalid++;  
            ⑥  
            sum = sum + valid[i];  
        }  
    }  
    ⑦  
}  
⑧ //

$$N = 9  
E = 12$$

$$V(G_i) = 12 - 9 + 2 = 5 \text{ Paths}$$

Paths  
0-1-2  
0-1-2-3-4-7-1  
0-1-2-3-4-5-7-1  
0-1-2-3-4-5-6-7-1



Conditional statements → True Path  
→ False Path

## Loop Testing

Single

Nested

Concatenated

input = 0, n = 5

```
for (i = input; i < n; i++) {  
    cout << *;  
}
```

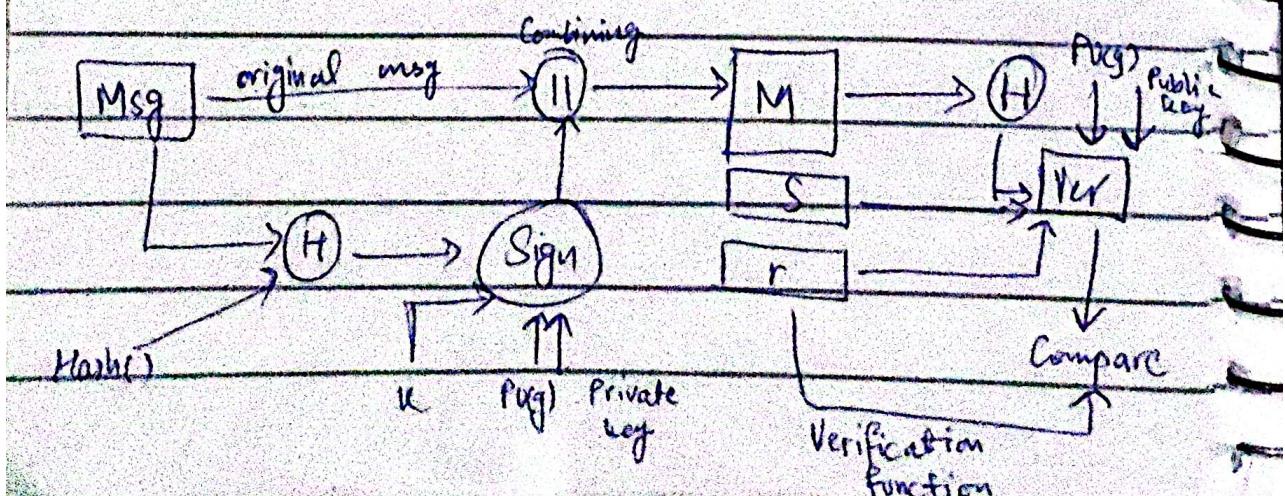
1 Skip loop by input  
2 Loop executed: 1 time  
3 Loop executed max time  
4 i = n + 1 => i = 5  
5 i = n - 1 => i = 4

for (i = input; i <= n; i++)	Output loop.	Inner Loop
{	i = 5	j = 5
for (j = input; j <= m; j++)	5	10
{	5	0
	5	6
{	5	4
}	10	5
}	10	10

0	5	10	0
0	10	10	6
0	0	10	4
0	6	6	10
0	4	6	0
6	5	6	6
		6	4

Outer	inner	Concatenated
4	3	$\{ \text{for } (i = \text{input}1; i \leq n; i++)$
4	10	$\}$
4	0	$\text{Loop1}$
4	6	$\text{cout} < "A";$
4	4	$\text{for } m++;$
		$\}$
		$\text{for } (j = \text{input}2; j \leq m; j++)$
		$\}$
		$\text{Loop2}$
		$\text{cout} < "A";$
		$\}$
index i	j	
6	5	
6	4 3	
6	3 2	
6	5 4	
6	0	

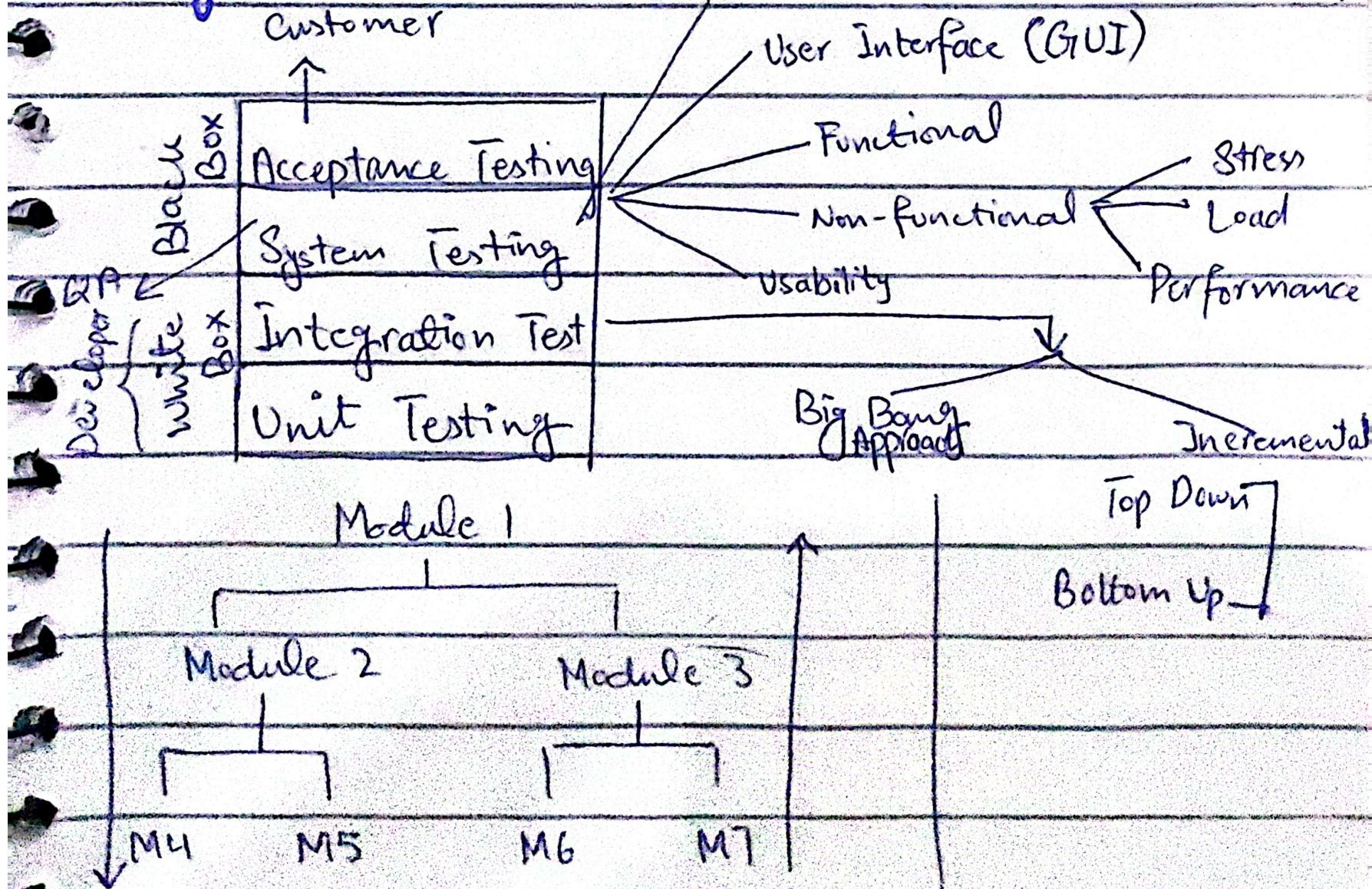
## DSA Model



Friday

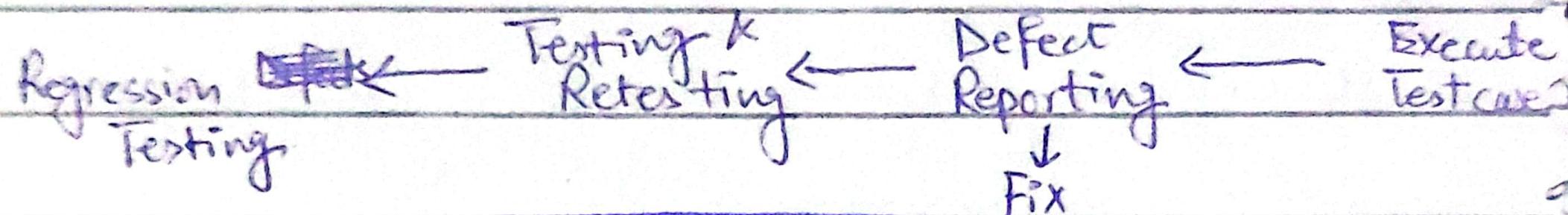
SQE Automation

19-A



## System Testing

Setup Environment → Test case generate → Testing data generate



Friday

SGE

3-5

## Software Reviews

### INFORMAL

#### Reviews

[Lunch]  
[Dinner]

- No need for presentation
- Group member/friend

CRS → SRS

### FORMAL TECHNICAL REVIEWS

(FTR) • presentation

• Schedule

• Agenda

• Walk through • checklist

• Raise issues/bugs

• Recording (minutes)

• 3-5 people

< 2 hour

Developer → Ready for review

↓  
Project Leader

↓ Make copies & distribute

Review Leader → Panel Reviewers

↓  
Reviewers → Review

↓ comments

Review Leader Advanced preparation

↓  
Project Leader

(agenda set)

Action Items

All attendees of

FTR must decide

• Accept the

product without

modification

• Reject the

product

• Accept provisionally

provisionally  
(changes)

### Summary Report

- What is reviewed
- Who reviewed
- Findings and conclusion
- Provide checklist for developers

7-5

SQE

Tuesday

## Data Flow Testing

Variable	Define $\leq$ Use		Code	
a	1	2, 3	read a, b, c	①
b	1	2, 5	if (a > b)	②
c	1	NA	int x = a + 1	③
x	3, 5	4	Print x	④
z	NA	6	else int x = b - 1	⑤
			Print z	⑥