

Class 1
15/06/19

Book :

① Software Verification & Validation for practitioners
by S. Rakitin

② Software testing principle and practices
by Naresh Acharya

— o —
V-S or Testing Model
(Lobone V&V)

Software Crisis : Resource scarcity
but quality software and Quality software
create quality software
model মডেল তৈরি করো
Browsers ব্রাউজার + Testing tools

Language PL+I

Haskell HAL/S

C++ → directly machine to C/C++ contact
NodeJS → model → Vanilla JS engine

Config compiler built
engine

V&V Model :

CASE → Computer aided sw engineering
softw. model, test case generate এসে, টেস্ট-

লাই

বোর্ড

but developper যুক্ত CASB রেজিস্ট্রেশন

তুমন একটি

মিলো, & Bd

Crisis recovered process project

① Formal profile of correctness

② Independent V & V

↳ (V&V model)

নির্বাচিত ক্ষয়ের নির্ধারণ ক্ষয়ের নির্ধারণ

cost optimize + time consuming

1+19 স্পোর্ট

এসআই লেসন

* SW crisis এর জাতীয় একটি কারন কারন কারন কারন কারন কারন

জনপ্রিয় কোম্পানি

জনপ্রিয়

0108/00/00

■ CMMI : Capability maturity model Integration

CMMI 5 } different SW organization
" 3 }

CMMI 5 → highest

↳ Industry organization কোর্টে ranking না থাবায়.

International company এবং BD এর
company কুমোরে judge করে CMMI 3 থেকে 5 পর্যন্ত

base করে, কিন্তু না

■ ISO SPICE → 14 কোর্টে follow

↳ এদের certification নির্ভুল সহ develop এবং তচন
কর্তৃত step follow কর্তৃত হয়

■ ISO 12207

object + Database → Requirements

■ Trillium → SW defects নির্ভুল করত

22/03/2019

SDLC (Software development life cycle)

① Waterfall Agile

② Planen drüren

Agile Process

Screem

Sprint (10 to 15 days)

→ 20%

Code freeze

→ यहां code ने change

पर तर तर testing
on any type करता-

-20-

Servere
(3 type)

dev → local
environment

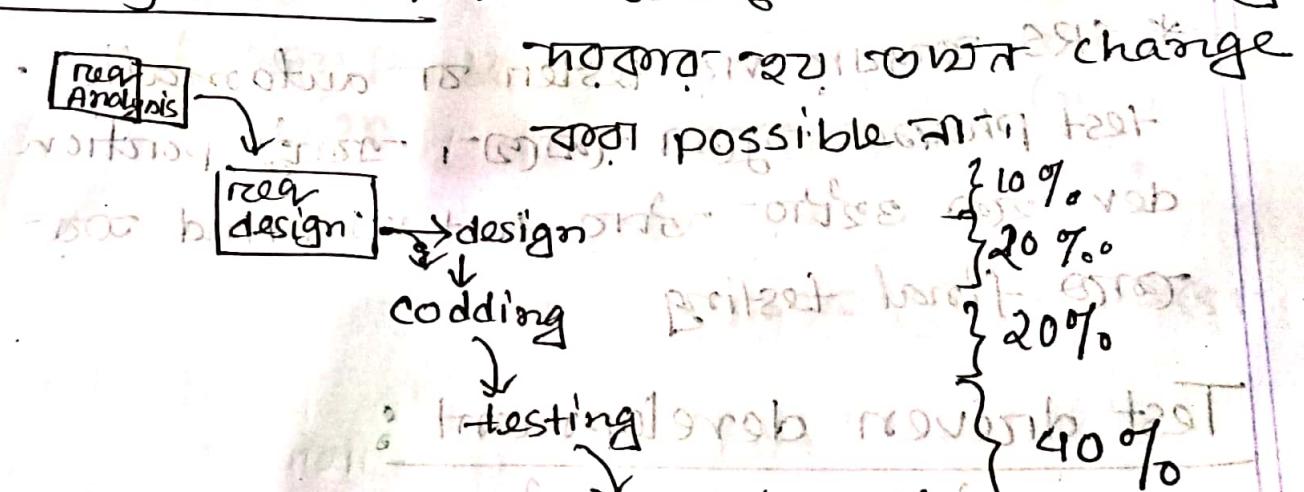
Staging → public
env.

Production

Scrum master → टीम स्क्रम मास्टर का रोल

SDLC का तर्ज → ① design phase
② requirement analysis

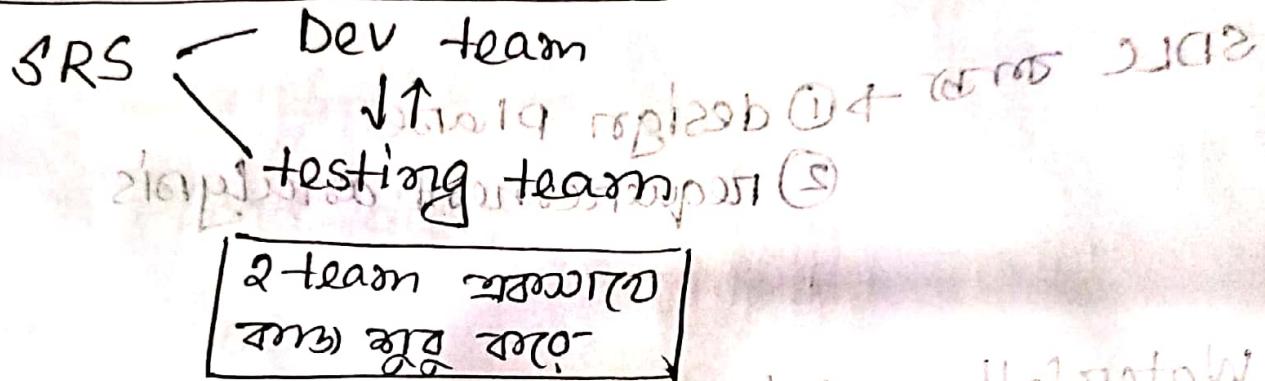
* Waterfall model → ज्ञानावाले नहीं परिवर्तन कर सकते



→ यह एक निश्चिह्नित विनियोगी प्रक्रिया है

→ ① Well documented → अच्छे प्रक्रिया plan करें

* Concurrent model



* SRS রিভিউয়ের পর testen বা automatic

test phases নিয়মো রয়েছে - প্রথম portion
dev করে ও এটো পরিসরে test করা। And তা
রেও final testing

Test driven development :

(১) অন্তর্ভুক্ত হোল্ড দেভলপেন্ট

First অ SRS দেখে test case গুলি করো

1 অস্ত, একে দেখে developer করো ①

FDP

Rapid Prototyping model

→ First 1 prototype

working
Prototype



Design Phase
Decision

then final prototype

→ തുകാ SRS മാറ്റ്

then. Prototype മാലാര്

SRS മുഴുവൻ കൂടുത്
കോർപ്പറേഷൻ (dev)

Book
222

Spiral model :

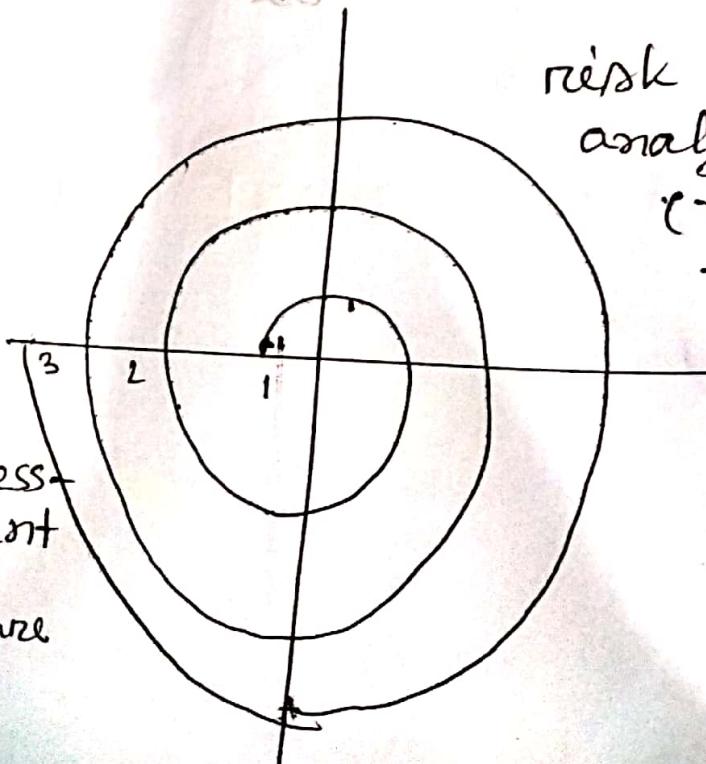
risk

analysis

(ചില - 20 വർഷ ഫീഡ്-
ബോക്സ് - 20 വർഷ - 20 വർഷ)

assess-
ment

Questionnaire



④

Hybrid model

merge ৰাস্তা

first & last
biological culture

→ app কোর্পোরেশন → agile

জিমি অবস্থা → waterfall

গুরুত্ব সংপৰ্ক কৰা

OOP model

↳ OOP language use -

↳ design planning

↳ Class, Activity, Data flow diagram

জিমি কোর্পোরেশন কৰা হোৰা হৈলো

8-09-19

08/09/2018 G. Sard

FAQ

① Why requirement must be written?

→ for client future claim or project analysis

→ CMMI certification

→ developper change

Book

The Art of SW testing

TDD (Test driven development)

Testing goal type :

① Immediate test (bug discovery) ବସନ୍ତ ମାତ୍ର

② Long term test (Product reliability
insure scalable,
respose, load balancing)

* reliability and * quality customer satisfy
* risk assurance 2 th related

③ Post Implementation : Project 1 କୁଣ୍ଡଳ କିମ୍ବା

କେତେ ଟାଙ୍କା ନେକ୍ଟା ଏହା କୁଣ୍ଡଳ କିମ୍ବା
କେତେ ଟାଙ୍କା କିମ୍ବା

* SW maintenance cost କିମ୍ବା

* trial & better

V&V

Methodology

21/08/2019

Class - 5

Git follow

Git tools

SourceTree

Gitkraken

Desktop framework

copilot tool API

(fresh labone कोड) copilot tool API

clone

SI

Condition - 1

26/08/2019

01/09 → Prototype

04/09 → IT test

Testing technique

obj. & mind

→ static
(प्रारंभिक)

→ Dynamic
(प्रक्रिया)

Q. S. I. → Q. P.

blind profile

White Box → functionality testing
fun. वित्तीय गति
कोड अवलोकन

Gray " → Security test
सुरक्षा परीक्षा

Black box

→ functionality वित्तीय - गति

कोड अवलोकन जूस टेस्ट

कोड इनपुट थ्रॉउट ऑफल आउटपुट
कोड परिवर्तन आउटपुट

contd.....

V&V

3rd chapter
validation

Requirement gathering

High level Design (Project structure →
design pattern
mvc / smvp / mvvm)
↓
Low level Design (Armed - module design)

↓ root find

validation

vali

* V&V কৈন নেওয়া

Client এর পাইল নির্কাণ, তারপর step

Chapten - 2 (2020)
nawresh book
chap (2+3)

S. Raktin
Chap - 1, 2, 3

Req analysis → ambiguity

① generic

② DFD → data flow ⇒ testing build

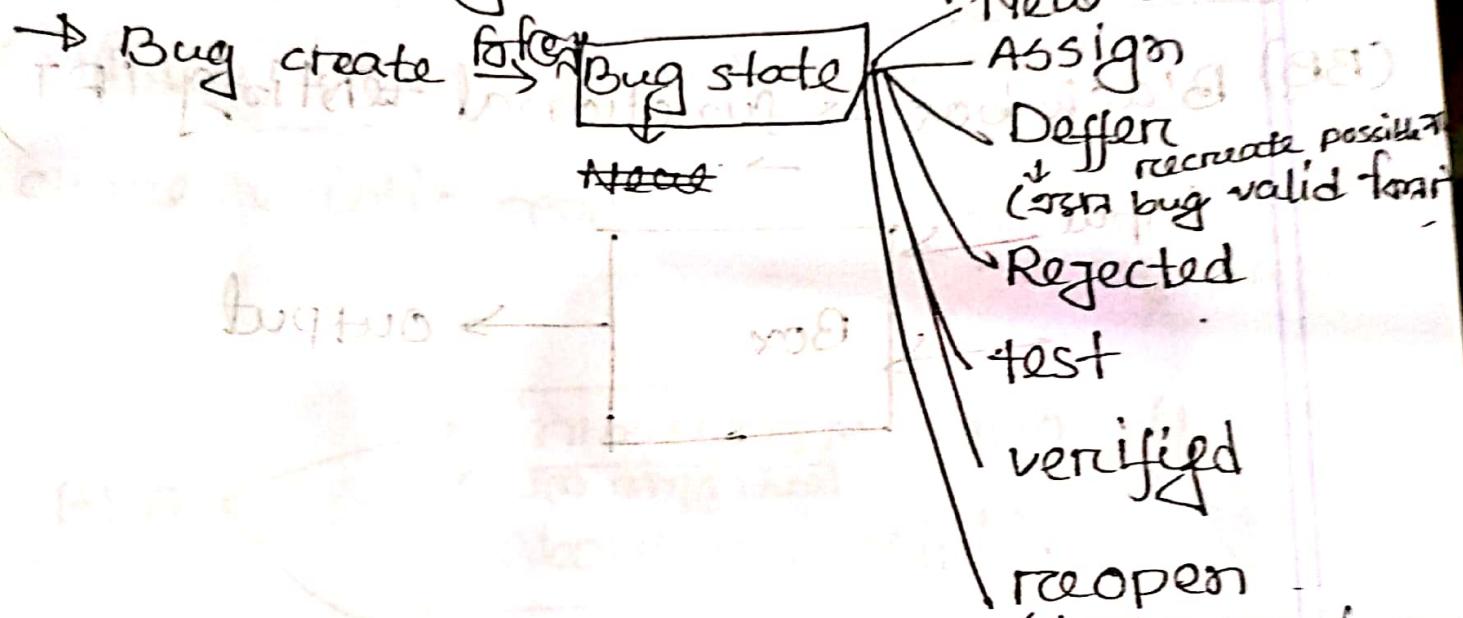
Coding
unit testing

1. security / load / responsive

integration
system

8/08/80/80

Bug repository



(goal not reached yet) right now you have to
recreate the bug for reassign

Ex Jira Atlassian

Bugzilla apache bug T4 right *

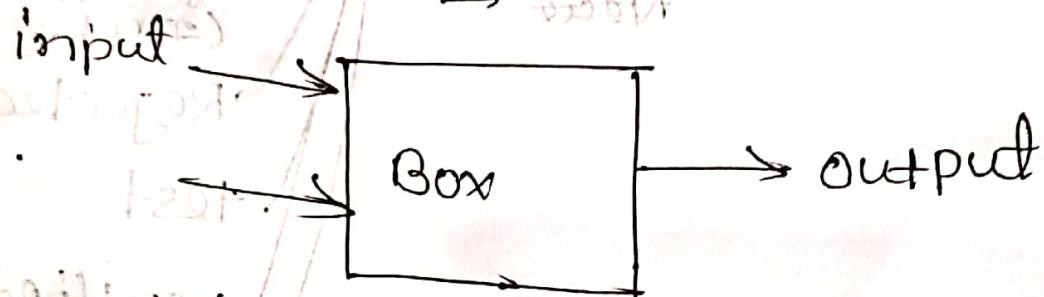
JMeter → load testing

what other tools can we use for this?

28/08/2019

Testing

(BB) Black box \rightarrow functional testing (FT)



BB න්‍යා මෙහෙයුම් (if, else, for loop)

කිහිපයේ තැබූ කදු අවස්ථා න්‍යා මෙහෙයුම් සඳහා



* එදිනී FT න්‍යා because මුළු module
න්‍යා මෙහෙයුම් කිහිපයේ තැබූ අවස්ථා න්‍යා මෙහෙයුම්
න්‍යා, but function තැබූ න්‍යා න්‍යා

→ boundary case මිලු න්‍යා

BB testing way

① (BVA) Boundary value analysis

characteristic \rightarrow define boundary value

+ for $a > b$ \rightarrow $b = a - 1$

BVA

minimum

($a < b$ or check)

above min (M_{int}^+)

maximum

below max (M_{int}^-)

for a & b 2nd - \rightarrow first fail test case

a	b
Maxa	not
Maxa	not
Min	
Min ⁺	

BVC max combination

$$2^n = 4n + 1$$

$n = \text{no of variable}$
 $= 2$

$4n + 1 = 9 \text{ no check}$
(maximum \rightarrow 2 or 3rd check
value)

② Robust Testing : maximum combination

$\frac{2}{2} \times \frac{3}{3} = 6$ + 1 = 7

RT → $\frac{2}{2} \times \frac{3}{3}$ = 6 + 1 = 7

Max + Min - Max + Max -

check $2 \times 2 = 6 \times 2 + 1 = 13$

Design of Test Cases
(Input / Output)
(Test Data Matrix)
(Test Case ID / S.NO.)

③ Worst Case Testing : boundary value testing

max no. of combinations = $\frac{n}{2}$

$n = \text{no. of variable}$

$$\frac{a}{1+ \text{Min}} \quad \frac{b}{\text{Max}}$$

$$\frac{c}{d} \quad \frac{e}{f}$$

Worst Case Testing

How to define test cases?



id	a	b	expected output
1	-1	100	invalid
2	bilov $\leftarrow 0$	100	valid
3			
4	bilov recent		

* TDD possible at unit testing case by case
* prepare test cases

book
exercises-4 for $\langle 28A \rangle$ p. 1

Equivariance Class Testing

class

- equivalent class identify করে test case define কর
- analysis করো → invalid case define (verify)
- ↓
then valid case
- 100% guarantee দ্বাৰা define

name password এবং require group (A, B, C) + type
for test case group define info

$$I_1 = \{ \langle A, B, C \rangle \mid A \neq 0 \} \text{ invalid}$$

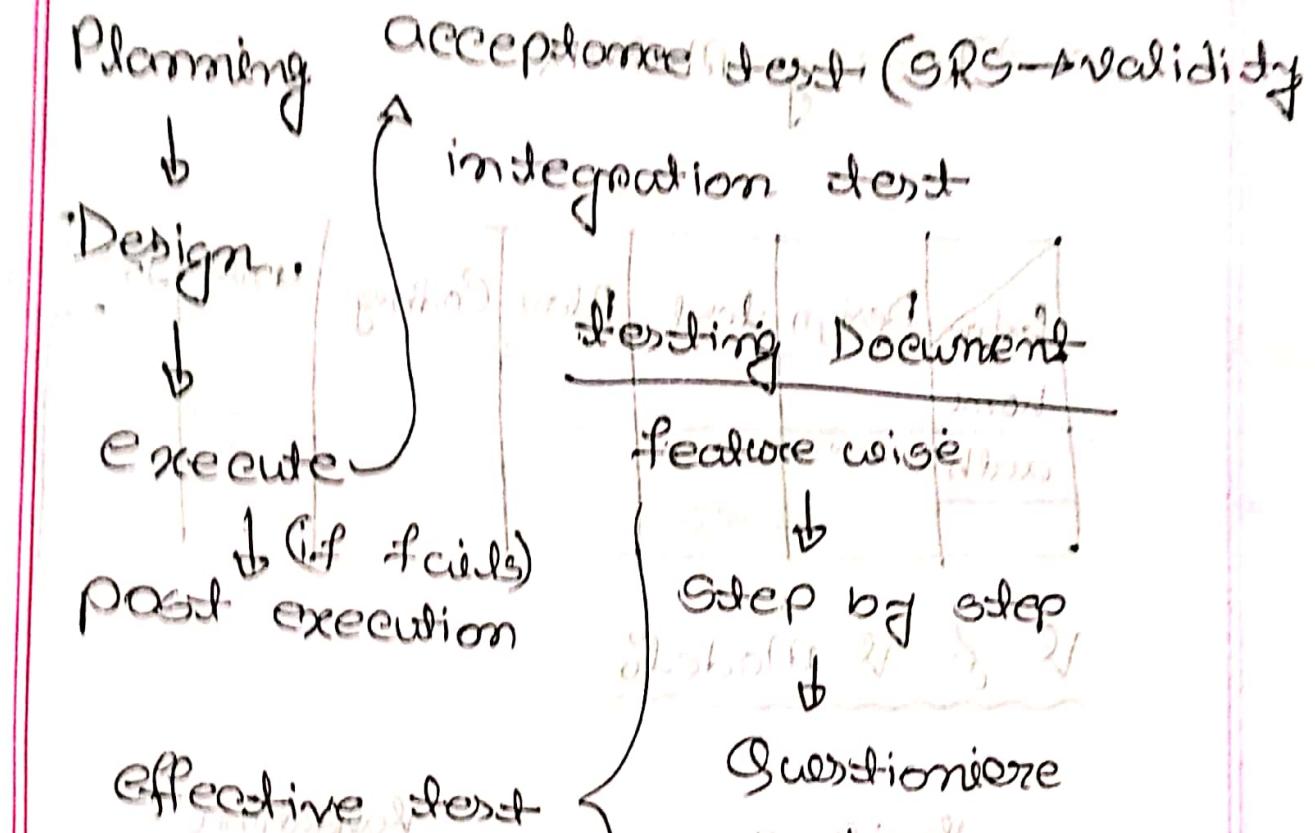
$$I_2 =$$

$$I_3$$

$$I_4 = \{ A, B, C \}$$

* 7th test case

STLC ~~Handwriting~~ ^{Handwritten}



- ① Test Planning
- ② Test case design (Redundancy ~~and~~ or ~~or~~)
→ Priority based work
- ③ Test execution
 - Report generation
 - Test log
 - Summary

④ Testing Methodology

Testing Matrix

feature	Design	Test	Plan	Coding	unit	Int.	Sys
home	-	-	-	-	-	-	-
auth	-	-	-	-	-	-	-

V & V Modelos

Verification

Req

Design

Coding

Validation

Coding Test

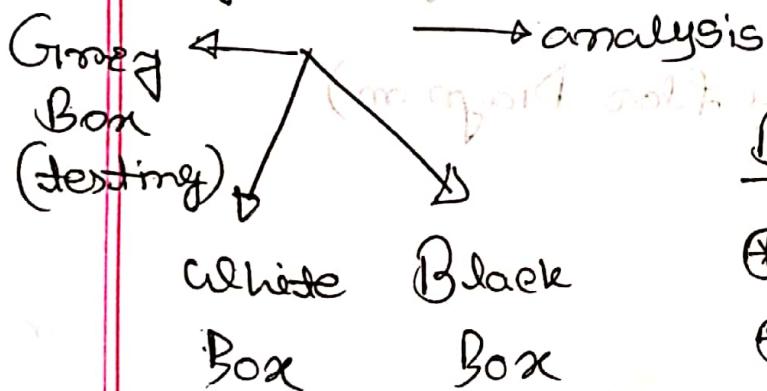
testing technique

→ Static

→ Dynamic

Static → anche ^{funzionamento} → design → Standard maintain } no code run

Dynamic → code review



Black Box → code unknown
 * input ↔ output
 * no knowledge of code inside

White Box

* transparent code

* functionality testing

* code can be seen

V & V Model

① Requirement Gathering

② Low level Design

High

③ Low " "

④ Coding

- no ambiguity
- no confusion about feature
- generic and configurable

DFD (Data Flow Diagram)

Bug Repository

Bug State

New → Open

→ Verified

→ Reopen

Assign

Deferred