

18/12/2023.

CS-322-SC

Monday

Software Construction

Chp#20 (Mc Connell)

→ Characteristics of Software Quality.

- Software has both external and internal quality characteristic. ↓
customer/client can see them.

+ External (User care about them)

- Correctness
- Usability
- Efficiency
- Reliability → It should not fail.
- Integrity → Security (related to Data)
- Adaptability
- Accuracy
- Robustness → Remain functional even if invalid input is given.

+ Internal (Programmer care about them, & also about external).

- Maintainability → 0 — 100% > 80%
- Flexibility → Good Documentation & Design
- Portability → Corrective Maintenance (Bugs)
- Reusability → Adaptive
- Readability → Preventive - improving design/ code before error is encountered
- Testability → Perfective - When you are enhancing the functionality.
- Understandability

Freedom is just a state of mind.

→ Techniques for improving Software Quality.

* Software-Quality Objectives

↳ setting the quality objective (goal)

↳ which external and internal characteristic we are focusing.

* Testing Strategy

* Software Engineering Guidelines

* Informal/Formal Technical Reviews.

QA.

* Development process.

→ Change control procedures

→ Measurement of Results.

→ Cost of finding Defects

• Testing is more expensive than inspection

* Rework reduces productivity and increases the cost.

Task: How do we measure productivity?

→ Verification and Validation:

V and V

Validation: Requirements k mukabiq hai ya nahi?

↳ Are we building the right product.

Verification:

↳ Are we building the product right.

→ Goal:

Goal is to establish confidence that software is fit for purpose

Depends upon:

Software functions

User expectations

Marketing Environment.

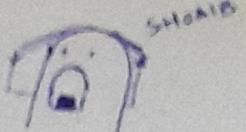
→ Approaches to V & V.

Static Technique: No running of software.

Dynamic: Involves running of software.

↳ Testing.

→ Inspections & Reviews.



- Non-functional Req such as execution time that effects performance can't be inspected. we have to do testing
- Generally dynamic approach is more expensive.
- In static approach we can inspect more than one errors in one go. But in dynamic we can't test in one go.
- For internal quality attributes. static approach is used for Exp: comments in code can be inspected not tested.

→ Testing:

- Successful test is -the one which finds an error.
 - You should be -the first tester but not the final tester.
 - Involve 3rd Parties (Not part of Development Team).
- Error: Discrepancy b/w computed or observed value, and the true or correct value.

Q: When to Stop Testing? (Task).

→ Testing Principles:

- Pareto Principle:
80-20 Rule.

→ Quality can be tested/checked through Testing but it can't be incorporated through Testing. (Quality → Design).

- Integration:

↳ Regression:

Testing that is done after changes has been implemented is called Regression Testing

↳ Smoke Testing:

↳ Became popular after agile Methodology.

• Validation:

↳ Alpha → Developers side

↳ Beta → customers

→ Black-Box Testing Method:

- Equivalence Partitioning
- Boundary Value Analysis.

Numeric valid	Non-Numeric invalid.	<10	10	>10
<1	1 - 100	>100		valid
Invalid	Valid Class		Valid	
-1	80		100	
0	1	100	101	

15/01/2024.

Monday.

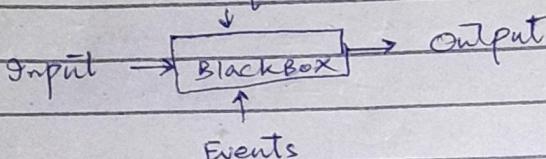
Lecture No.

→ Testing Techniques :

- Black-Box Testing

- White-Box Testing.

Requirement.



Events

↳ Interrupts / Button click etc.

→ Black Box Testing Methods.

- Equivalence Partitioning:

↳ 1st class will be valid and Invalid data classes.

Exp: Sale Quantity.

1 - 999.

Numeric

Valid class

Character

Invalid class

↳ Must check Boundary values.

Numeric

↓

≤ 1

g.v

1 - 999

Valid

> 999

g.v.

-1, 0

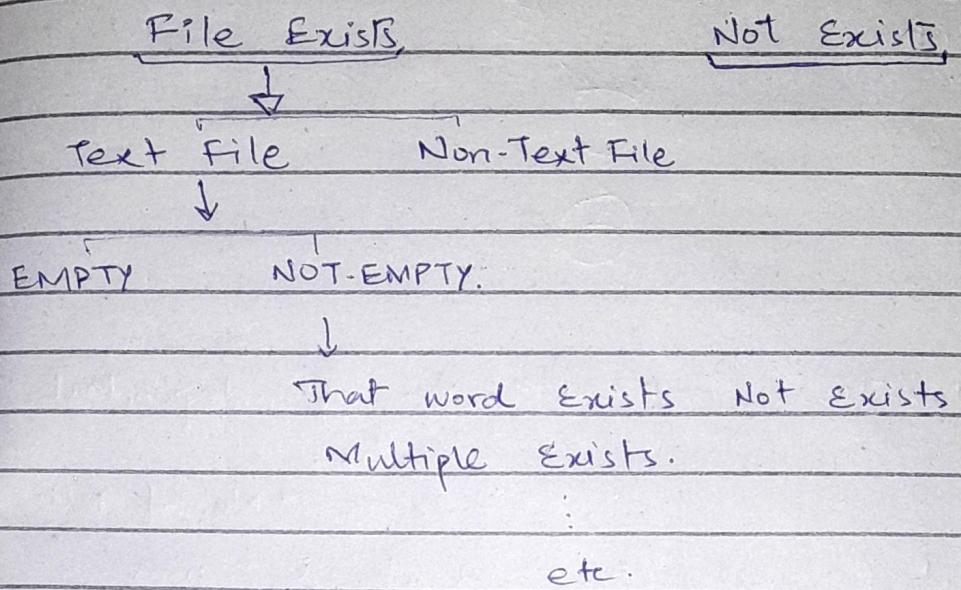
1 ↓ 999

500

1000

] Test cases.

Ex: A program that reads a Text file and a single word. It counts the occurrence of the word in the file and returns the result.



→ White-Box Testing

- We know the design and code.
- To ensure that all statements & conditions have been executed at least once. (Bases Path Technique)

* Flow Graph :

1. Number the statements. (Give composite conditions numbering separately.)

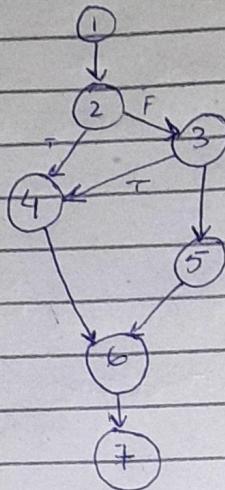
OR Problem.

Some problem

in slides

using OR

in condition.



→ Cyclomatic Complexity: No. of independent paths.

↳ Represent No. of test cases.

Total Regions : 3. - 2 closed regions
- 1 outer region.

$$V(G) = \text{number of regions}$$

$$V(G) = \text{Edges} - \text{Nodes} + 2$$

$$V(G) = \text{Predicates} - \text{nodes} + 1$$

Results should be same.

Note: cyclomatic Complexity is proportional to the number of Errors.

Path - 1 : 1-2-5-6-7

Path - 2 : 1 -

Path - 3 :

Test cases: -1, 25, 51 for x.

Test case : All the possible values that make it possible to go to that particular path.
Not one value input is a test case everytime.

- Sequential statements combined into 1 block and represented by one node in Graph.
- Regions must be closed using a node. (endif, {}, Blockended).

Exp:

Test Case TC 1 N=0

TC 2 N=1, $x(I)=1$

Draw Back:

In case of loops, if it's executed once, path is followed and on next iteration it will not give error that will not be detected because not a new path.

- Bases Path Technique.
 - Overall good for program
 - But not good for loops.