DELTA Pg No.
Date 15/10/2018

Module-4

Testing

Error - Caused by programmer.
fault - Incorrect Error
failure - Incorrect fault

Testing: Acc. to IEEE, the process of Analyzing a software item to detect the differences between existing and required condition and to evaluate the features of the software item

or

The process of Analysing a program with the intent of finding errors.

Objectives of Software Testing:

- Software quality improvement
- Verification & Validation
- Software Reliability & Estimation

Principle of Software Testing

- All test should be based on User requirements
- Test should be planned long before testing begins
- It is impossible to test everything
- Use effective resources to test
- Testing must be done independent third party.
- Testing time & resources are limited.
- Document test cases & test result.
- Provide expected test result if possible.
- Assign best personnel to the task.

DELTA Pg No.
Date 17/10/2018

Diff. b/w Black Box & White Box

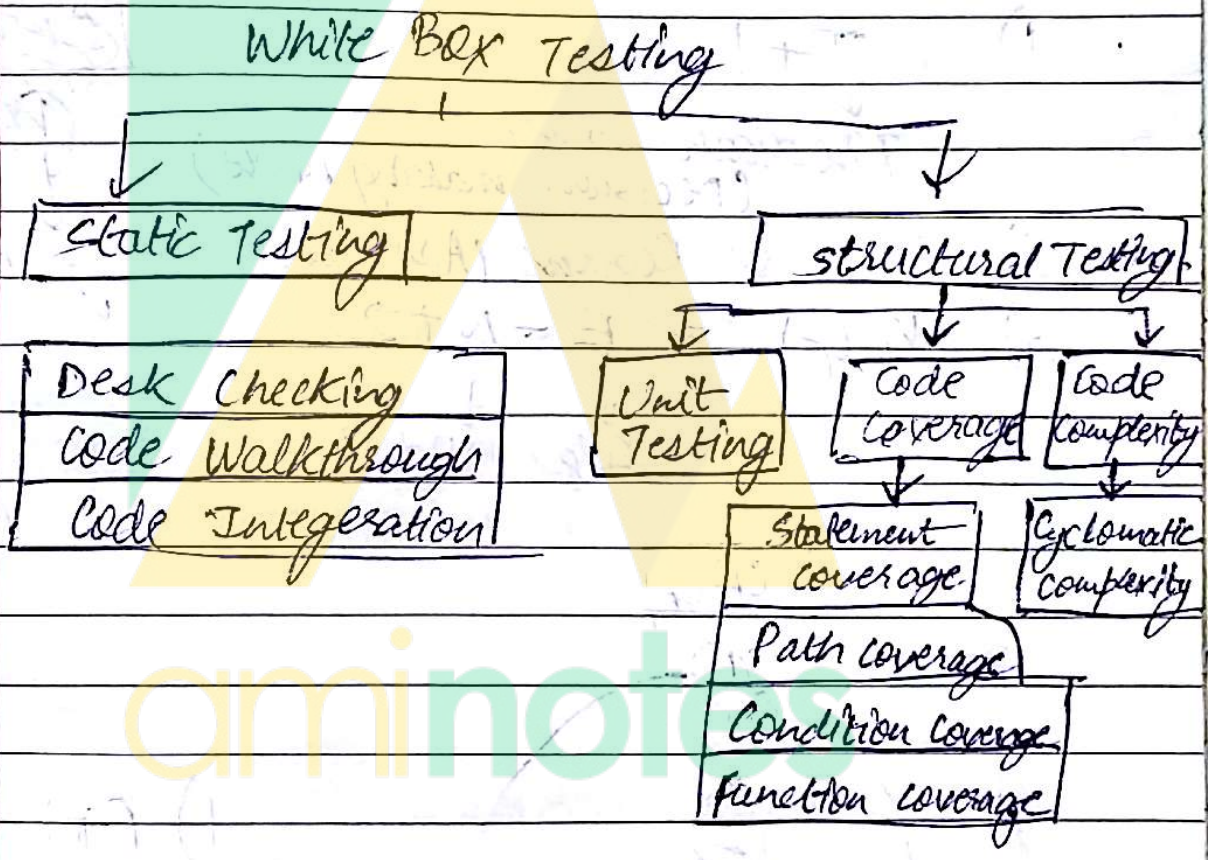
	Black Box	White Box
1.	It is performed by end users and also by testers and developers	1. Normally done by testers and developers.
2.	Black Box is a technique where no knowledge of the internal functionality & structure of the system is available	2. Testers have full knowledge of the internal working of the application.
3	This the least time consuming and exhausting	3. The most exhaustive and time-consuming type of testing.
4	This technique can be done by trial and error method	4. Data domains and internal boundaries can be better tested.
5	This type of testing is suitable for large projects.	5. This type of testing is suitable for small projects.
6.	In this testing the test designer selects valid and invalid i/p & determines the correct o/p.	6 In this testing tester chooses cases inputs to execute paths through the code and determine the appropriate outputs.
7.	In this testing program cannot be tested 100%	7. In this testing, test the program thoroughly.

DELTA | Pg No. | Date / /

White Box Testing (Structural Testing)

The techniques under white box testing are:-

1. Basis path testing
2. Structural testing
3. Logic based testing
4. Fault based testing



→ Basis Path Testing

↳ Control flow Graph

1. A CFG shows the control structure of code
2. Each node / circle represents block of code that has only 1 way through the code.
3. The edges b/w nodes represent possible flow of control.

DELTA | Pg No. | Date / /

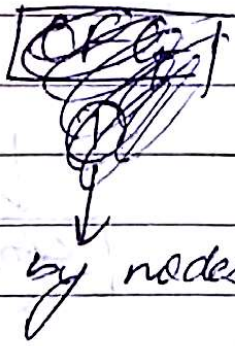
Cyclomatic Complexity

1. Its a ~~major~~ measure of the structural complexity of a procedure

2. It directly measure the no of independent paths through the program

Methods of calculating Cyclomatic Complexity:-

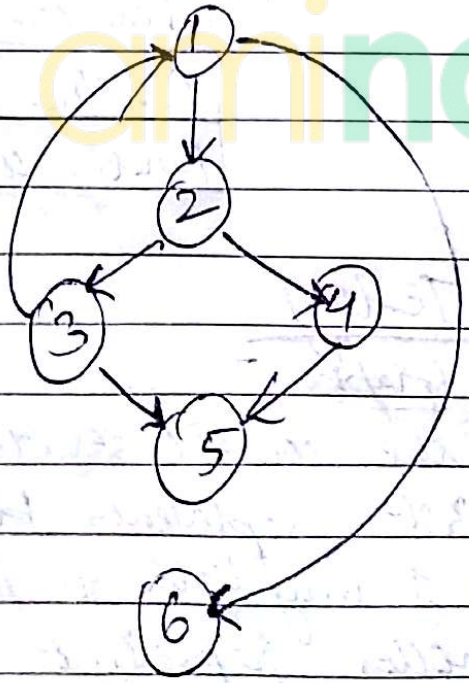
1) P+1
↓
Predicate nodes
(decision making node)



2) No. of Regions (Area enclosed by nodes & edges)

3) V(G) = E − N + 2
↓ ↓
Edges nodes

CFG



1) P+1 = 1+1
= 2

2) No of Regions = 3

3) V(G) = 7−6+2
= 3

22/10/2018

Maintainance

→ Adapting (FD)
→ Corrective (SRS)
→ Perspective (OS)
→ Preventive (Obsolete)

→ Meaning :- Software maintenance is modification of a software product after delievery to correct false, to improve performance or other attributes or to adapt the product to a modified environment.

DELTA Pg No.
Date / /

Need / Aims :—

1) Correct errors
2) Change in user requirement with time
3) Changing in hardware and software requirements
4) To modify the components. 5) To update the s/wetc.

Types of Maintainance

1. Corrective Maintainance :—

The main objective of corrective maintainence is to remove errors from the software, hardware, procedures, network, data structure & documentation.

2. Adaptive maintainance—

It means changing the program functions. This is done to adapt to external ~~sept~~ environment changes. Eg:—

The current system was designed so that it calculates taxes on profits after deducting the divident on equity shares. The govt. has issued orders now to include the divident in the profit for tax calculation. This function needs to be changed to adapt to the new system.

3. Perfective Maintainance —

It means enhancing the performance or modifying the program to respond to the user's additional or changing needs

4. Preventive Maintainance

It is a process by which we prevent our system

from being obsolete. It involves the concept of pre-engineering and reverse engineering in which an old system with old technology is re engineered using new technology.

Software Maintainance Process Modell

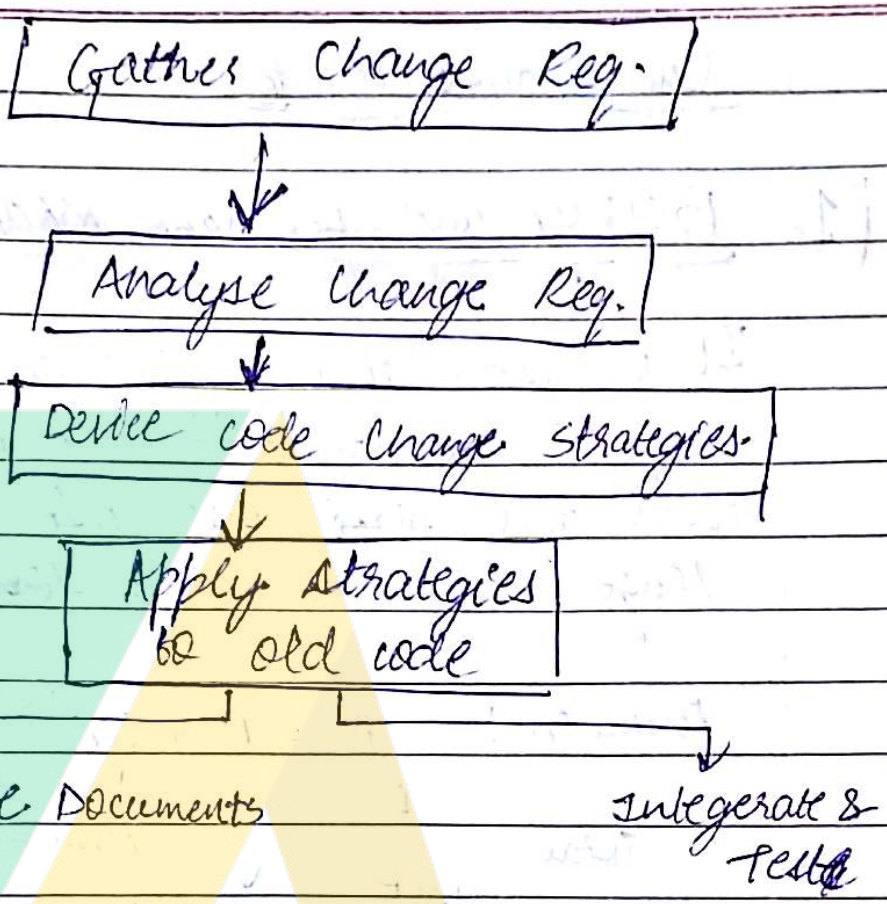
Two broad categories of process models for software maintainance can be proposed.

1) The first model is preferred for projects involving small rework where the code is changed directly and the changes are reflected in the relevant documents later.

2) The second process model for s/w maintenance is preferred for projects where the amt of rework is significant. ~~A reverse~~
A reverse engineering cycle is followed by forward engineering cycle can be represented by this approach

Diagram →

DELTA Pg No.
Date / /



DELTA Pg No.
Date 23/10/2018

Maintenance Model

(1) Belady and Lehmann Model:

It indicates that effort and cost increases if a poor software development approach is used and person that use the approach is no longer available to perform the maintenance.

Formula: $M = P + Ke^{(c-d)}$

- M: Total Maintenance effort
- P: Productive effort
- K: constant
- (c-d): degree of maintenance.

Q- The development effort for a software project is 300 person/month. The empirical deterministic constant $K = 0.3$. The complexity of the code is quite high and is equal to 8. Calculate the total maintenance effort.

i) If the maintenance team has good level of understanding of the project: $d = 0.9$

ii) The maintenance team has poor understanding of the project. $d = 0.1$

Soln

Development effort (P) = 300PM

$K = 0.3$

$c = 8$

i) Good level of understanding

$d = 0.9$

$M = P + Ke^{(c-d)}$

$= 300 + 0.3e^{(8-0.9)}$

DELTA Pg No. Date / /

= 300 + 363.59

= 663.59 PM

iii) Poor level of understanding

d = 0.1

$M = p + ke^{(c-d)}$ (8 = 0.1)

= 300 + 0.3 e

= 300 + 809.18

= 1109.18 PM

(2.) Boehm Model

Boehm used a quantity called annual change traffic which is defined as the fraction of a software product source instructions which undergo change during a year either through addition, division or modification.

Formula :-

1. The ACT is related to no. of change request

$$ACT = \frac{KLOC_{added} + KLOC_{deleted}}{KLOC_{Total}}$$

(SI → source Instruction)

2. Total cost is maintained by.

Maintainance cost = ACT × development cost

[ACT → Annual change traffic]

3. AME = ACT × SDE

[AME → Anual maintenance effort
SDE → Software Development effort]

4. AME = ACT * SD * EAF

[EAF → Effort adjustment factor.]

Reverse Engineering & Re-Engineering

Re-Engineering

The process of modifying the internal mechanism of a system or program or data structure without changing its functionality.

Process :-

i) Goal setting
ii) Critical Analysis of Existing Scenario such as Process, task, etc.
iii) Identifying the problems and solving them by new inovating thinking

Process [Detail]

1. Source code translation
2. Reverse Engineering
3. Program Structure improvement
4. Program Modularisation
5. Data Re-Engineering

Advantages :-

~~Disadvantage :-~~

1. Lower cost
2. Lower Risk
3. Better use of existing staff

Disadvantage :- Re-Engineered system is not likely to be as maintainable as new system

DELTA Pg No.
Date / /

Reverse Engineering

It is any activity that improves ~~understan~~ understanding of s/w, prepares or improves the s/w itself usually for increased maintainability, reusability, etc.

Goals :-

1. Facilitating s/w Re Use
2. Coping with complexity
3. Detecting side effects

Application areas of Reverse Engineering

1. Re-documentation
2. Program Comprehension
3. ~~Re-use~~ Identifying Reusable components.

DELTA Pg No.
Date 29/10/2018

FURPS → (Study Yourself)

- Functionality
- Usability
- Reliability
- Performance
- Supportability

May Mod.

Q — Write a short note on the following diagram

1. Sequence Diagram
2. Collaboration Diagram
3. Component "
4. Deployment "

Reliability

According to ZEEE standard software reliability is defined as the ability of a system or component to perform its required functions under stated conditions for a specified period of time.

How likely is it that a s/w component will produce an incorrect outcome software failure is usually distinct from hardware failure. In that S/w does not wear out.

* Reasons for s/w reliability being difficult to measure.

1) The peake of reliability is highly observer dependent.
2) The reliability of product keep changing as error are detected and fixed.

DELTA Pg No. Date / /

* Factors due to which S/w may retire
i) Change in environment
ii) Change in technology
iii) Major change in requirement
iv) Increase in complexity.
v) Extremely difficult to maintain
vi) Poor QUZ.

Hardware reliability

→ What is the probability of the h/w component failing or how long does it take to repair that component?

Most h/w failure are due to component wear & tear

Errors :-

Fault - It is defect in the program that when executed under particular condition, causes of failure. A fault is created when a priegramm makes a error.

Failure - It is the departure of the external results of program operation from requirements. A system is said to have failure if the service it delivers to the user deviated from compliance with the system specification for specified period of time.

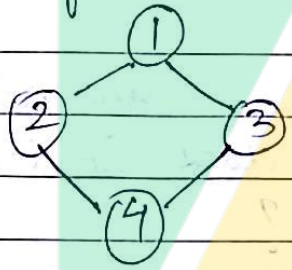
DELTA Pg No.
Date 30/10/2018

Cyclomatic complexity

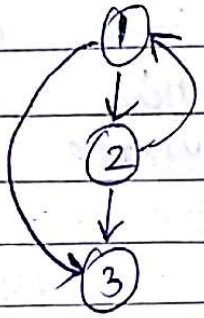
Basic Notations

(1) Sequence.

(2) If then else

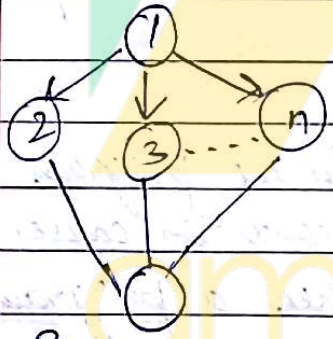


(3) While Loop



(4.) do-while

(5.) Switch



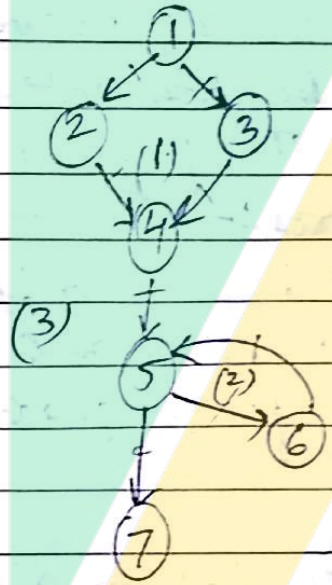
Program:

```
1)  void foo (float y, float a*, int n)
    {
        float x = sin(y);
        if (x > 0.01)
2)          z = tan(x);
        else
3)          z = cos(x);
4)      for (int i=0; i<n; ++i)
5)      {
6)          a[i] = a[i] * 2;
```

DELTA Pg No
Date / /

```
        cout<<a[i];
    }
7.) cout<<i;
}
```

Program Flow Graph



No. of Regions = 3

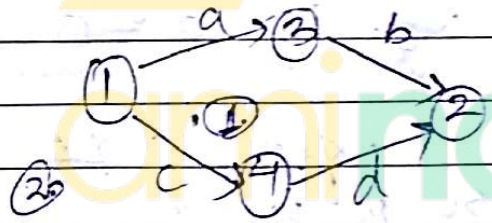
P+1 = 2+1 = 3

V(G) = E-N+2

= 8-7+2

= 10-7 = 3

Graph Matric



	1	2	3	4	Formula → (count occurance) − 1
1			1	1	2−1=1
2					
3		1			1−1=0
4		1			1−1=0

1+1=2

DELTA | Pg No.
Date / /

Black Box Testing Techniques.

(1.) Boundary Value Analysis (BVA)

Q- Consider a program for the determination of nature of roots for a quadratic equation. Its inputs is a triple of the integers say a,b,c and values may be ~~from~~ interval [0,100]
The program O/p may one the following words :-
Not a quadratic eq; real roots ; quad. roots;
~~Design~~ not a imaginary root ;
Design a BVA for the quadratic eqn.

Sol^n :- $ax^2 + bx + c = 0$

Roots are real if $(b^2 - 4ac) > 0$
imaginary if $(b^2 - 4ac) < 0$
Equal if $(b^2 - 4ac) = 0$
Not ~~quad~~ if $a = 0$

BVA

Test Case	a	b	c	Expected Output
1	0	50	50	Not Quadratic
2	1	50	50	Real Roots
3	50	50	50	Imaginary
4	99	50	50	"
5	100	50	50	"
6	50	0	50	"
7	50	1	50	"
8	50	99	50	"
9	50	100	50	equal
10	50	50	0	Real
11	50	50	1	Real
12	50	50	99	Imaginary
13	50	50	100	Imaginary

Interval :-
0 - 100
0, 100, 50, 1, 99

Problems:-

→ Triangle
→ Quadratic eq
→ Date

Theory from PPT.

(2.) Equalance Class Partioning (ECP)

0-100

-1, 0, 100, 50, 101

31/10/2016

ECP

Quadratic Eq.

Test Case	a	b	c	Expected Output
1	0	50	50	Not a quadratic
2	-1	50	50	Invalid
3	50	50	50	Imaginary
4	101	50	50	Invalid
5	50	50	50	Imaginary
6	50	-1	50	Invalid
7	50	101	50	Invalid.
8	50	50	50	Imaginary
9	50	50	-1	Invalid
10	50	50	101	Invalid.

(3.) Cause Effective Graphic Technique.

Q- The Characters in column 1 must be a or b
The charater in col. 2 must be digit.
In this situation file update is maintained
If the character in column 1 is incorrect
message x is issued. If character in col 2 is not a digit, message y is issued.

DELTA

sol 4 The causes are

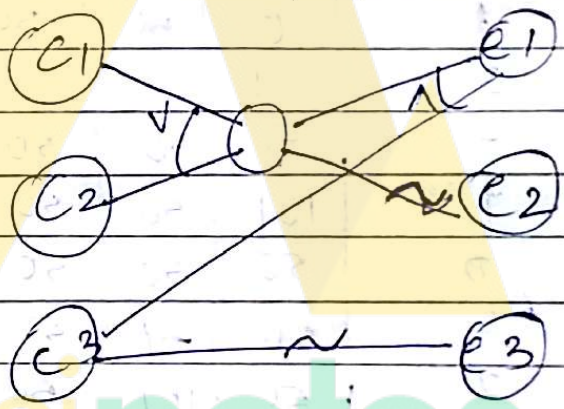
C_1 Character in column 1 is A
C_2 " " " B
C_3 " " column 2 is digit

and the effect are

e1 : update made
e2 message X
e3 message Y

Symbols
AND ∧
OR ∨
NOT ~

Graph



(4) Decision Table

Email
Id

Conditions					
Email	F	T	F	T	
Password	F	F	T	T	
Action					
Expected O/P	error	error	error	login.	

DELTA Pg No.
Date / /

Q. Write a short note on Data Flow Testing. Explain the no. of step.

No. of combination = condition value 1 X condition value 2

= 2 X T/F

= 2 X 2 = 4.