

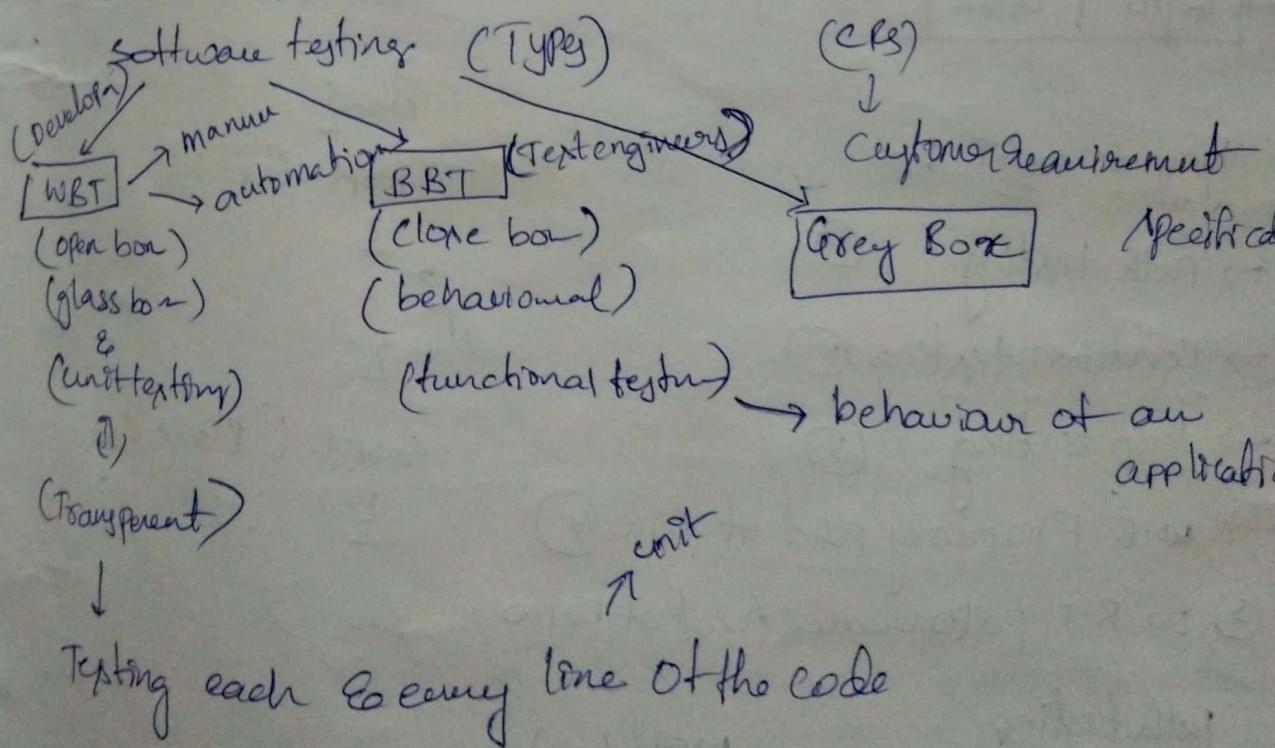
Software testing :-

Verifying the functionality of an application against requirement specifications is called software testing.

- The process of finding defects in a software is called software testing.
- execution of the program with an intention to find bugs - ✎

Why:-

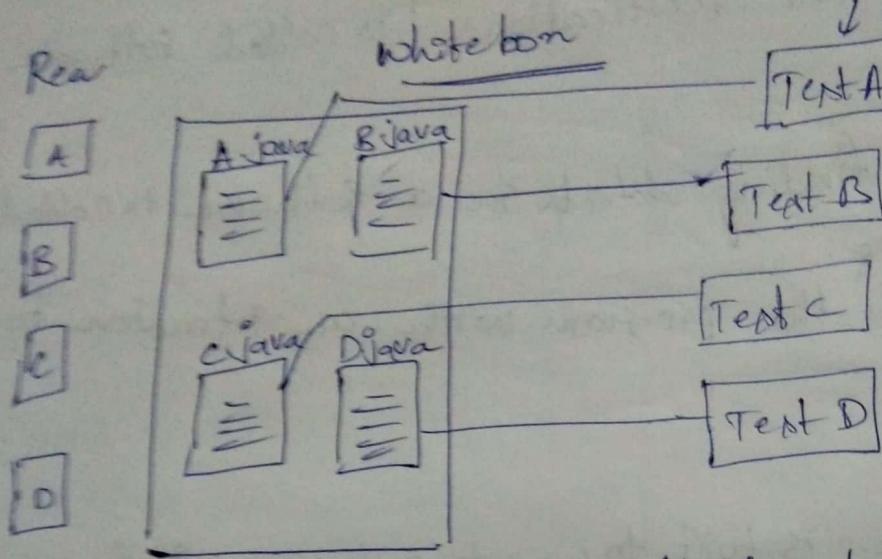
- ① every software is built to support a business if there are any problems in the software it will effect business. So before you launch in to market recognize all problems and fix them.
- ② To improve the quality of the product
- ③ To make sure software works according to customer requirement.



NOKIA

→ Combination of WBT & BBT

grey box



API → translates from one platform to other platforms.

unit test case → will be copy to remember the test case.

SIP		OIP	
A	B		
10	6	led	
5	5	Black	
10	10	Green	

Type of WBT :-

→ Path testing. ①

→ Condition testing ②

→ Loop testing ③

→ W.B.T memory point of view ④

⑤ W.B.T performance point of view.

if (a == 10)

{

if (b > r)

{

System.out.println("Red");

} else

System.out.println("Blue");

}

else {

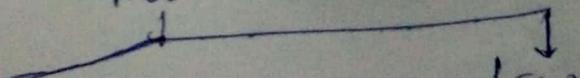
System.out.println("Black");

}

NOKIA

Path testing

main()



→ Testing the path of the Independent function
⇒ Condition testing :-

Testing the logical condition i.e., True, False is called condition testing.

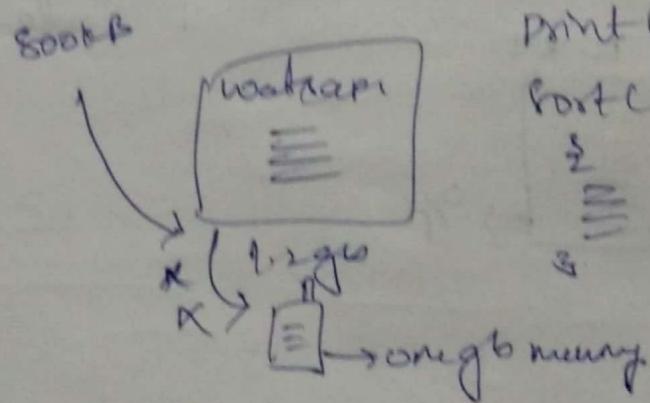
⇒ loop testing :- Testing the condition or iterations is called loop testing.
⇒ Memory point of view :-
→ Efficiency of code.

Ex:- \times not used
 $= \text{int } i, b;$

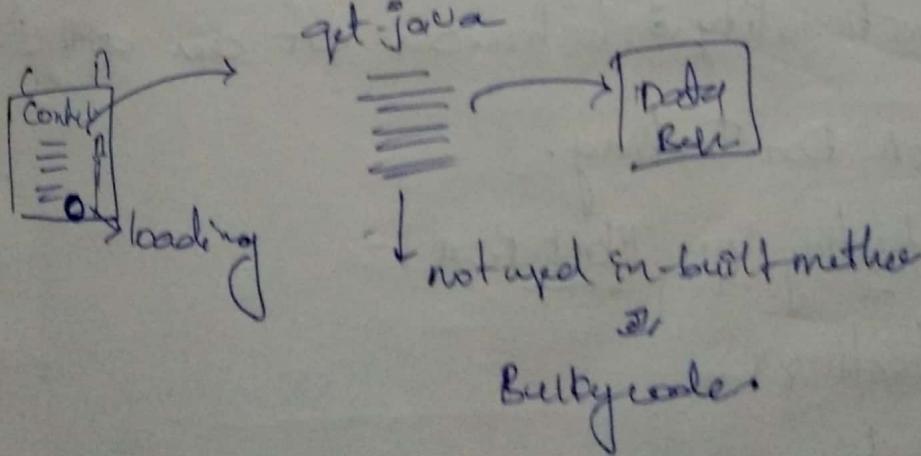
$b = 10$
`print(b);`

`for (c)`

$\{$ \equiv $\}$ \times not called
 \times used.



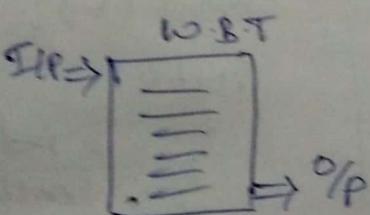
⇒ Performance point of view :-



Difference b/w W.B.T & B.B.T.

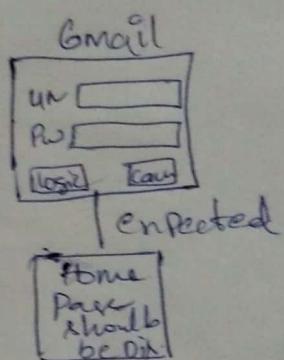
[W.B.T]

- ① Testing each & every line of a code.
- ② Developers are involved in W.B.T
- ③ Source code is visible.
- ④ We should know the programming language.
- ⑤ We should know the internal structure of the code.



[B.B.T]

- ① Testing the functionality & behaviour of an application is called - B.B.T
- ② Test engineers are involved here.
- ③ Some code is invisible.
- ④ We need not know the programming language.
- ⑤ We need not know the internal structure of the code.



SDET

Black Box Testing :-

Testing the functionality & behaviour of an application is called Black Box testing.

Different types of testing in Black Box :-

- Functionality testing.
- Integration testing.
- System testing. (end to end testing)
- smoke testing.
- Adhoc testing.

NOKIA

- ① Usability testing
 - ② Regression testing
 - ③ Exploratory testing
 - ④ Performance testing
 - ⑤ Accessibility testing
 - ⑥ Globalization testing
 - ⑦ Recovery testing
 - ⑧ Reliability testing
 - ⑨ Functionality Testing :-
- (I18N)
Internationalization Testing
- Localization Testing
(L10N)

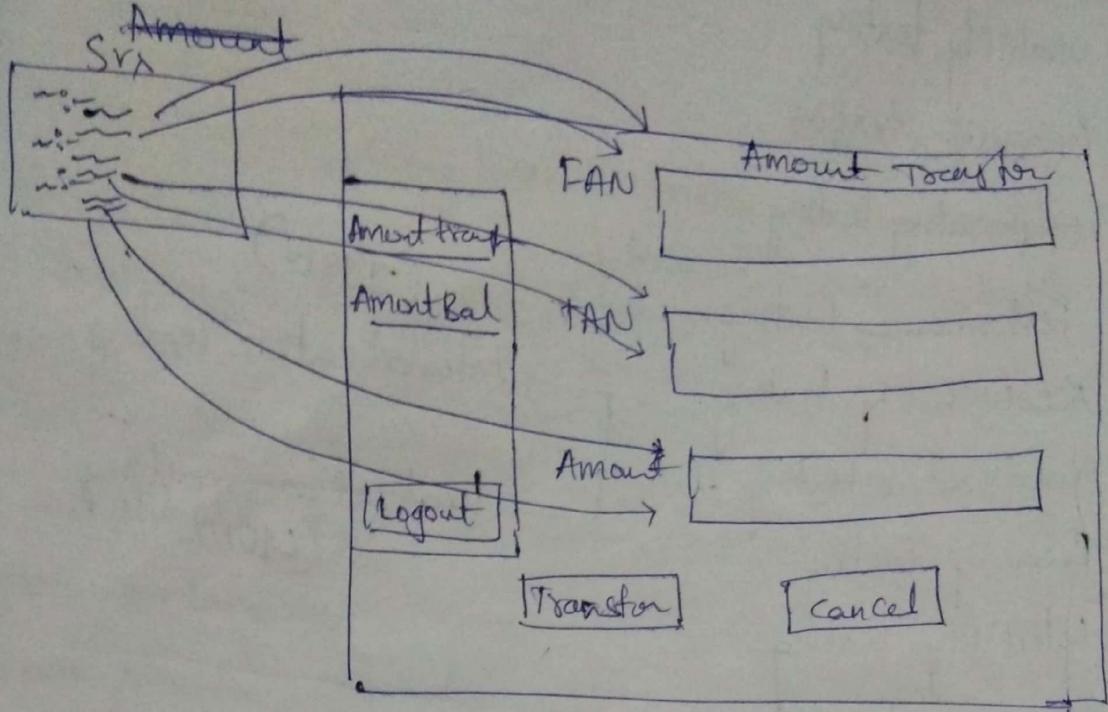
CBO — SRS (Software Requirement Specification)

- 3.0 Amount transfer page
- 3.0.1:- TAN Text field
 - 3.0.1.1:- Should accept 10 digits
 - Valid Alc number
- 3.0.1.2:- Should accept Alc number which are created by manager.
- 3.0.2 :- TAN Text field
 - 3.0.2.1:- Should accept 10 digit
 - Valid Alc No.
- 3.0.2.2 :- Should accept Alc number which
- 3.0.2.3: Amount Text field
- 3.0.3.1 :- Should accept 100-5000 integer numbers.
- 3.0.3.2 :- Should not accept amount greater than the Balance

NOKIA

Checking each & every component of a functionality by

On the requirement.

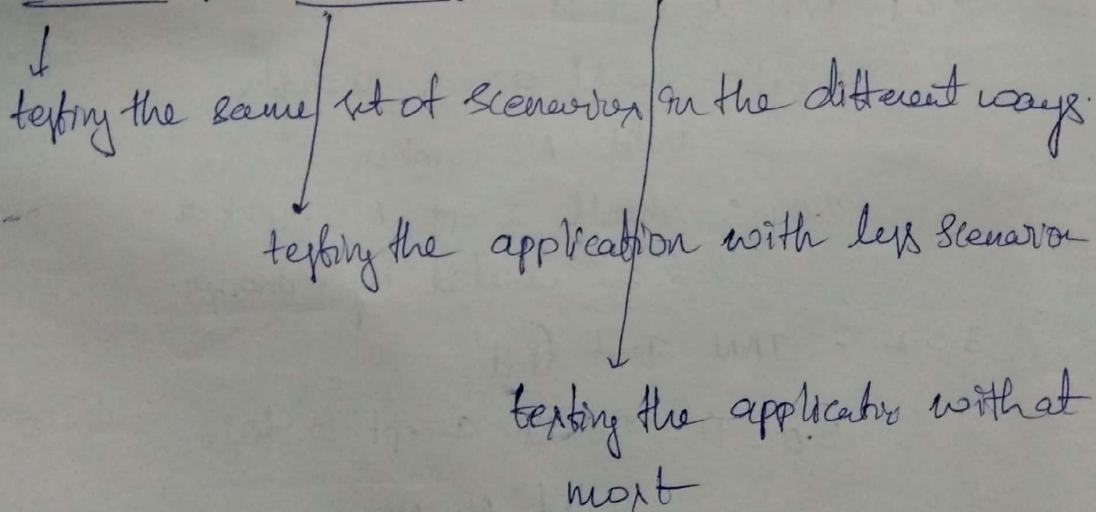


→ always testing should be starting with the requirement.

① positive scenario

② negative scenario

Over testing, under testing, optimize testing



For positive we can do functionality testing on different ways:-

① over testing :-

NOKIA
- testing the application

② functionality with

→ Here we are going to face lot of bugs because we spent lot of time in testing the same scenario.

Under testing :-

Testing the application with minimum set of scenarios is called under testing.

Optimize testing :-

Testing the application only with those scenarios, which makes sense, is called optimize testing. Here we are going to miss less no. of bugs.

Positive testing :-

Testing the application with expecting data & valid data. It is called positive testing.

Negative testing :-

Testing the application with unexpected & invalid data is called negative testing.

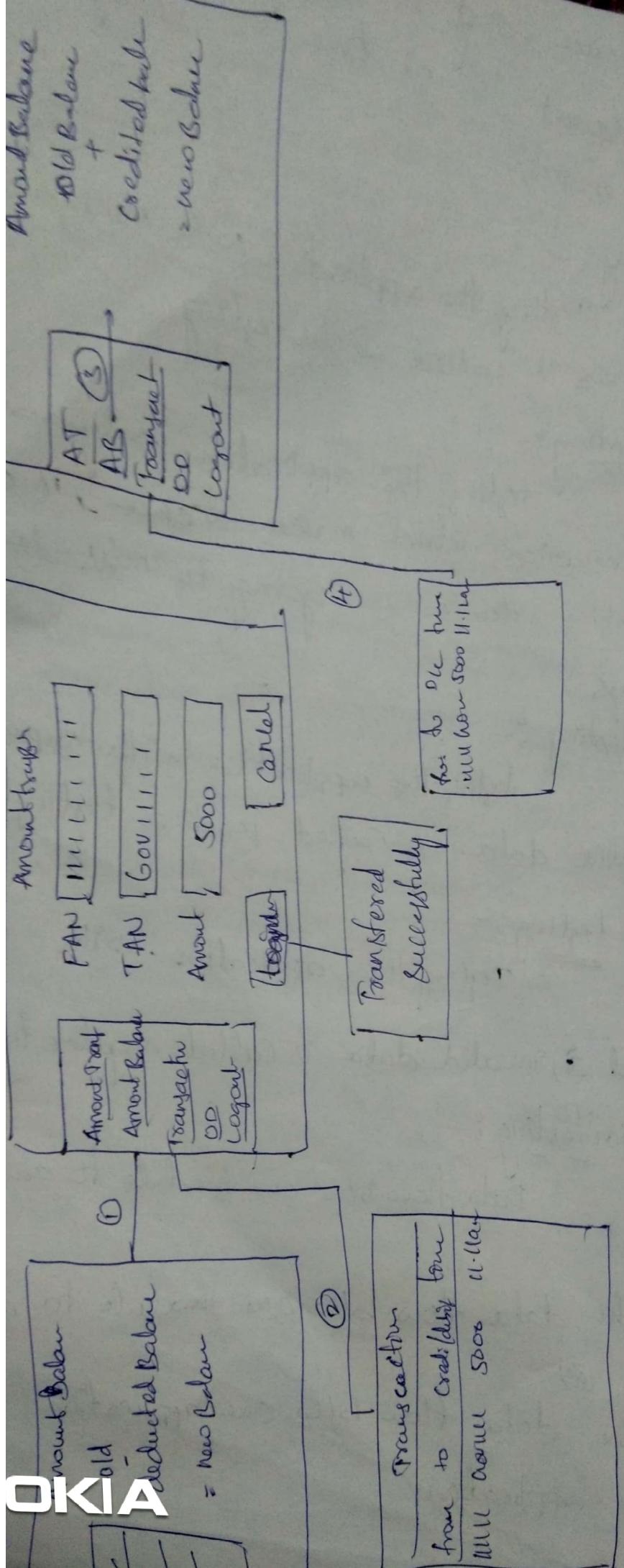
Integration Testing :-

Data flow b/w one module to another module.

→ Testing the data flow b/w one module to another module (3)

Testing the data flow b/w one application & another application.

NOKIA

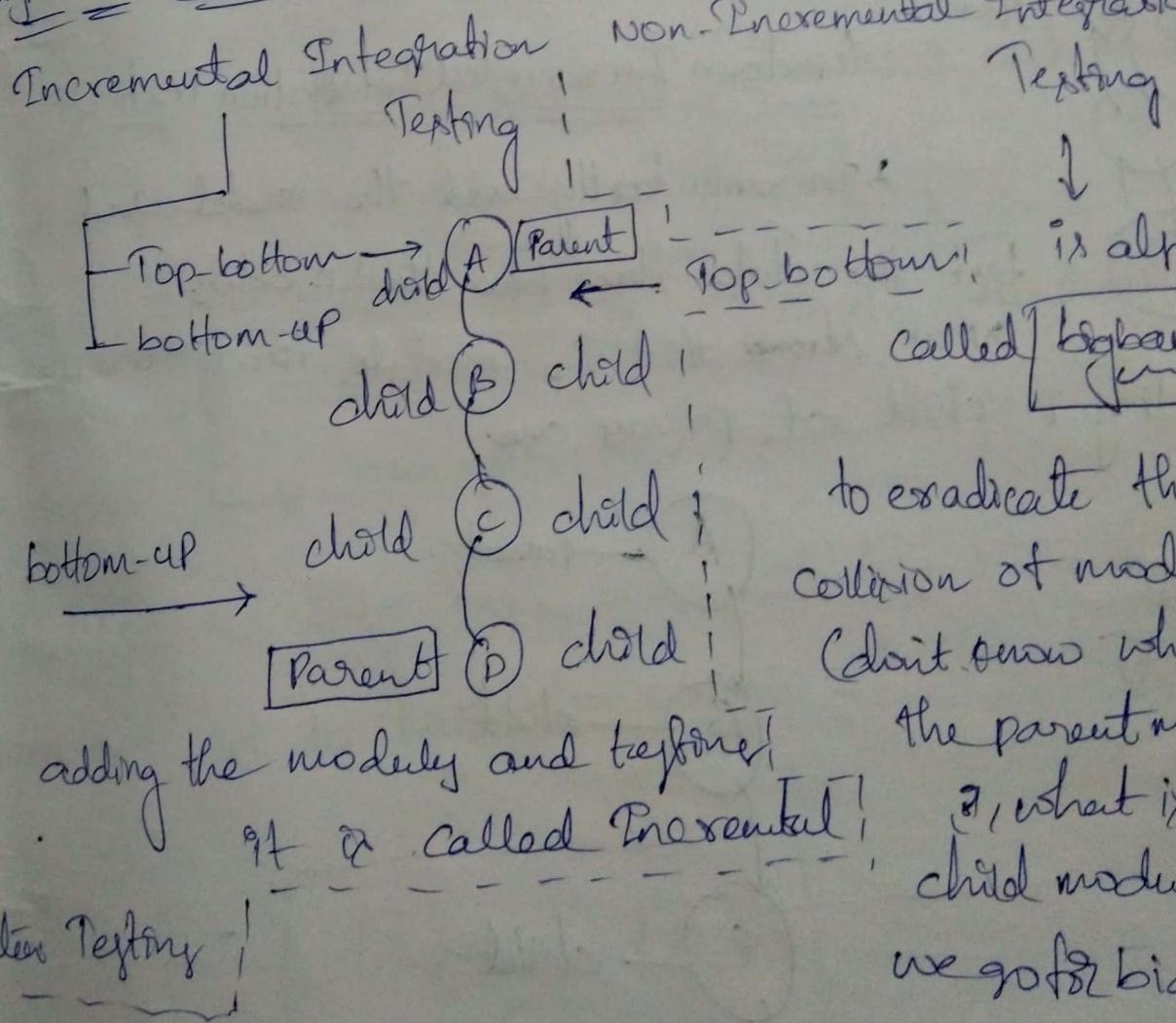


OKIA

How to do integration Testing?

- ① understand the app.
- ② understand each component how it works and how it related to other component.
- ③ Identifying all possible integration scenarios
- ④ prioritize the identified scenarios.
- ⑤ Document the identified scenarios.
- ⑥ execute the app. acc to the documented scenarios
- ⑦ while testing if you opt any defect report it to developing team.

types of integration testing :-



NOKIA

Incremental Integration testing :-

→ Take two modules check the data flow b/w the two if working fine, if it is working, then add one more module and test again. Continue like this incrementally. Add the module and test the data flow b/w the modules.

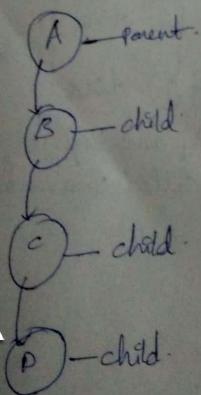
→ There are two ways :-

- ① top-down Incremental Integration Testing.
- ② Bottom-up Incremental Integration Testing.



top-down Incremental Integration Testing :-

→ Incrementally add the module and test the data flow b/w the modules
→ make sure that the module that we are adding is child of previous one.



NOKIA

Bottom-up integration testing:-

→ testing starts from bottom up to child.
→ incrementally add the modules and test the data flow b/w the modules.
→ make sure that the module you are adding is the parent of the previous one.

Non-Incremental Integration Testing :-

→ we use this method when

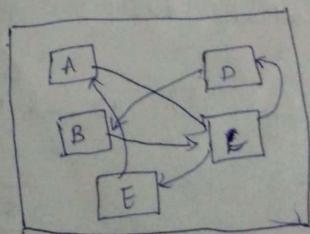
- a) when data flow is very complex.
- b) when it is difficult to identify who is parent and who is child. They also called as "Bigbang" method.

c) Combine all the modules at a shot and start testing the data flow b/w the modules.

→ The disadvantages of bigbang method is -

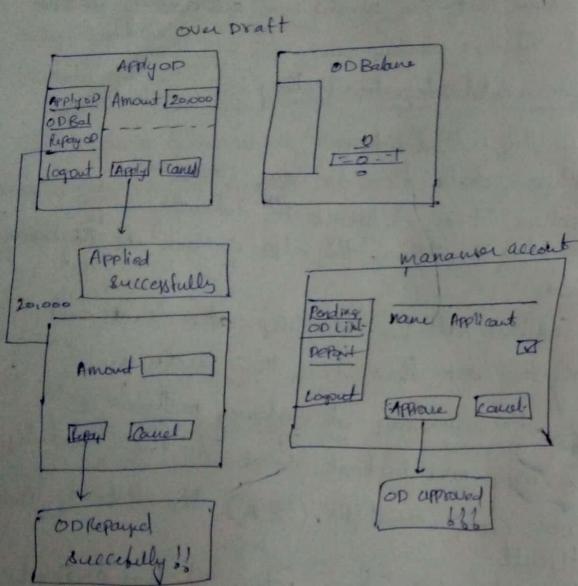
a) we may miss to test some of the modules.
b) root cause analysis (RCA). The defect is difficult.

Identifying the bug from where it has originated and will not get to know the origin of the bug.



System testing

→ it is nothing but end-to-end testing where in test environment is similar to production environment.



① Scenario

OD → 20,000
40,000
800 → 2% Interest
200 → Application fee

41,000

② OD → 40,000
100
000

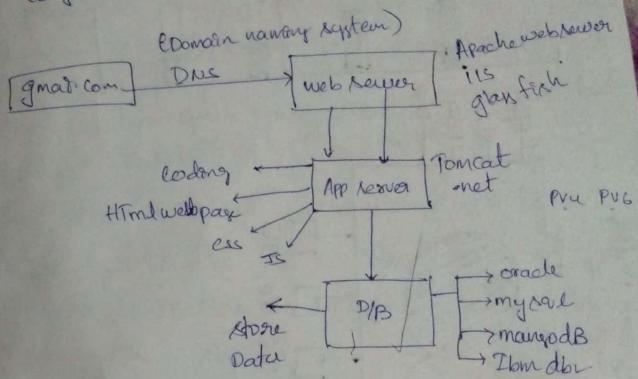
Increased
OD → 50,000
500 → 1%
000
50,500

NOKIA

web server

Development Environment :-

→ It is nothing but a set up which is used to develop the software.
It consists of hardware, software and network.

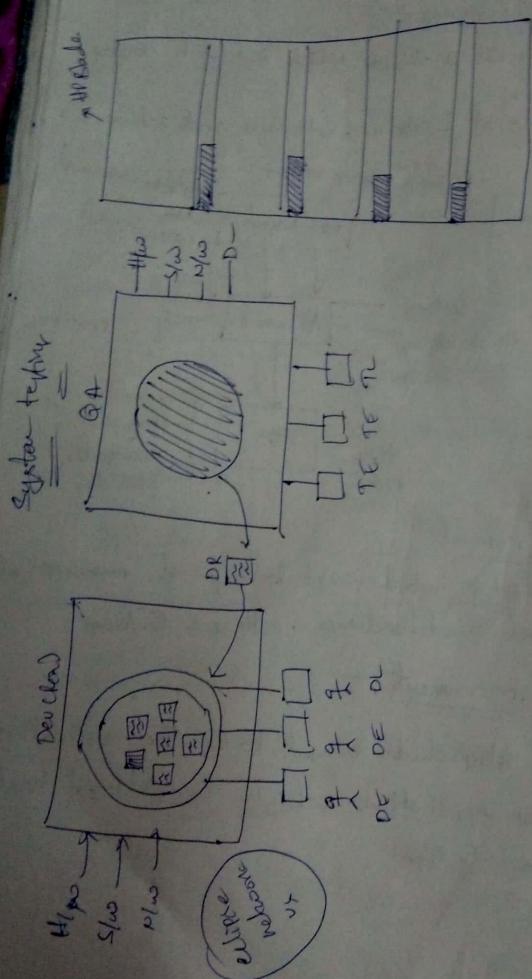


Testing Environment :-

→ It is a setup which is used to test the application which consists of hardware, software & O/W.

Production Environment :-

→ It is a setup which is used to run the application & serve the application which consists of hardware, software & O/W.



NOKIA

The testing environment similar to production environment means:

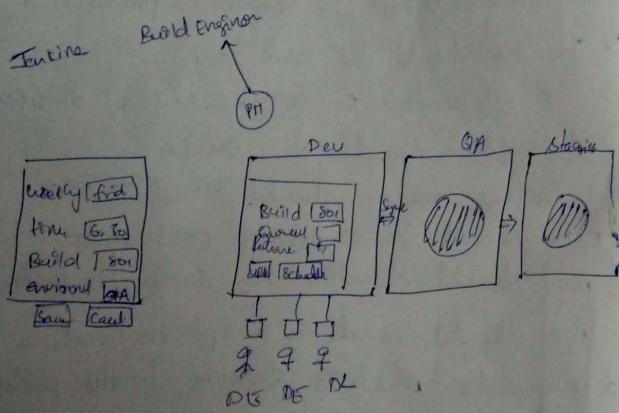
- ① H/w should be similar to production
- ② The make (manufactured by) should be similar to production server. (For example if the production server is H-P then test dev should be H-P)
- ③ Configuration and make must be similar, but different capacities. (i.e. no. of cores).
- ④ The software should be similar to production
- ⑤ The O/S should be similar.
- ⑥ The application server should be similar.
- ⑦ Web server should be similar.
- ⑧ Database servers should be similar.
- ⑨ Data should be similar to production.
- ⑩ We should create data similar to production.
- ⑪ We should create a script to create a dummy data which is similar to the production environment.

In real time environment we may make lots of entries into database, but while testing we can't enter manually lots of entries. So, we are going to write test script program which generate lots of user data, and

They can be for testing.

Intertesting environment who is involved in installing application

- ① any body from testing team.
 - ② any body from development team.
 - ③ Build \oplus release engineer.
- > Build is a piece of software
 > which is compiled, compressed and installed in all the environments.



Q) Zentinel?

It is the continuous integration tool it is used to automate the building process.

Q) Devops?

NOKIAing but a development operation

which involving activity like development.

manage the code, Build a new Build, install and uninstall the application in all the environments
 It called devops (Development, QA environment)

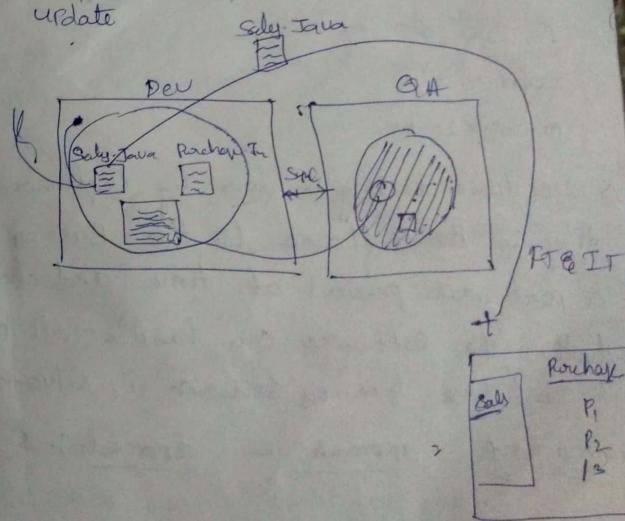
Q) what is continuous Integration?
 It is a process which includes development operation where it will make sure that all the environments are synchronized continuously.

Q) what is patch?

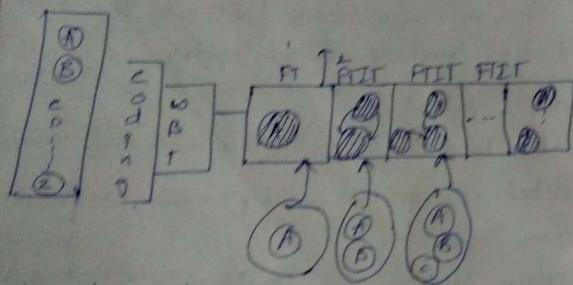
A) patch is nothing but it is a modified program,
 B) deleted program C) added program, which will be done in less time

→ when patch is given to QA we have going to uninstall the old build and just we are going to install the patch.

update



Time spent on testing a build of software completely is called test cycle.



Acceptance testing:-

critical = 0

It is done by customer.

Blocker = 0

major = 4 - 10

minor = 20 - 40

→ acceptance testing is done by customer here they use the software for the business for a particular period of time and check whether the software can handle all kinds of real time business scenario or situations.

Approach 1 Approach 2 Approach 3

NOKIA

why we do acceptance testing :-

→ banking due to lot of business pressure software companies will push the software to find out the defects, customer will do acceptance testing.

→ developer misunderstands the requirement and develops the wrong requirement, then the customer do acceptance testing.

→ unit testing, test engineer will miss some of the critical bugs due to negligence, 2) misunderstanding of requirement to find out those mistakes customer will do acceptance testing.

① difference b/w ui prototyping and functional testing :-
② ui prototyping team do fun - ① functional testing teams do acceptance testing, U-T, system - end to end testing.

CF = Change requirement

RFE = Request for Enhancement

F.T
I.T
S.T
A.T

→ functional Testing.

[Performance, Usability] → non-functional testing.

AGILE Model :-

- we are going to split all the requirements.
- > Agile is a model, where we develop & test the software in an incremental & iterative way.
- > They came up with this model in order to overcome the drawbacks, that were there in the traditional model. Here they build product in shorter cycles (4-8) (one month).
- > Product backlog is the list of stories.
- > Product backlog is the list of stories.
Priority.

Scrum:-

to Develop and trying the app.
Both are involved in the same project is called as Scrum.

<u>Story</u>	<u>Assign Task</u>	<u>Task</u>
coding		
ui/ux design		
D/B		
LLD		
WBT		

NOKIA

Product Backlog:-

- > it is a prioritized list of features & requirements/stories.
- > Here stories need not to be detailed.
- > Product backlog is owned and managed by product owner.
- > Sprint backlog, which is a list of stories and associated tasks, that must be completed within a sprint is called sprint backlog.
- > It is a list of stories and tasks committed by scrum team to be delivered in one sprint.

Sprint Planning meeting :-

- Here entire scrum teams meets together and pull the stories from the product backlog.

- a) whatever they can build with in a sprint
- b) we assign this story to the engineer
- c) the engineer derives the task to be performed to complete (to build the complete story)
- d) They estimated the time taken for each story. ^(to complete)

Scrum Master:

- Scrum Master drives the Sprint planning meeting.
- His prime role is to facilitate. Complete meeting and co-ordinate b/w the stakeholders.
- In this meeting the product holder clarifies if any queries or questions are there w.r.t to the stories.
- In this meeting D.E. ^(Development Engineer) drives the task for building every task.
- Development engineer prioritizes the stories and tasks.
- In this meeting, test engineer should derive the task to implement the feature built for a story.

NOKIA

→ Retrospective meeting
 → entire scrum team & discuss about achievement and mistakes (good practices) is followed and we document it and that document is called as retrospective doc.

→ when next sprint starts, while doing sprint planning, we refer this document and we planned in a such way that old mistakes are not repeated. and good activities are again adopted.

Daily standup meeting (10-15) mins.

- Scrum master drives this meeting.
- Scrum master facilitates this meeting.
- Discuss clearly, yesterday and upcoming task whether it is completed, unfinished. etc.
- Discuss about impediments (issues) in meeting. if it is solvable then and there solve it or else take the note on that issue & solve it later.

Note:-

- a) each engineer should tell what he has done yesterday.
- b) what you are planning to do to

Q what are the expected impediments

(d) what are the impediments to faced.

(e) generally the meeting should go for 10-15 min.

(f) generally this meeting is done on the evening of the day.

Scrum team :-

→ group of members (6) individual working together to deliver & to complete the committed stories.

→ generally Scrum team size will be 7-8 people.

→ There two kinds of teams, Core team, Shared team.

→ Core team involves, Scrum Master, Developers, and Testers.

→ Shared team involves,

① BA

② Architect

③ UI/ux designer

④ Product owner

⑤ Admin, Database.

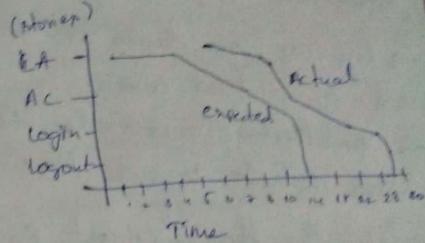
NOKIA

who leads the Scrum master

= Scrum master leads the complete Scrum team and he facilitates everyone to complete part.

Burn down chart

It is the graphical representation of work left & time



Daily can Burn Board or Story board or white board :-

① it contains the list of pending and completed task of each engineer.

②

Start	In progress	Completed
Create ALC		
WT	→	→
RTC		
apply ALC		
IP		
Review		
Review		

Sprint review meeting

(Last meeting at the agile)

- Here entire scrum team meets at the end of the Sprint and discuss about the how well the Sprint went.
- Engineers give demo of what ever they build to product owner.
- Product owner tells, what is done and what is not done.
- also they discuss about how to plan a next Sprint.

Spit over (chicken) :-

Some people don't actually working, but they will be there to ~~see~~ observe what is happening in the Sprint and that person is called chicken.

Sprint over :-

- There are certain boundary are notes that we can build in current ~~sprint~~ build which will be moved to next sprint it is called Spillover.

NOKIA

Smoke testing

- what is smoke testing? Testing the basic & critical application before doing thorough testing is called smoke testing.
- Smoke testing can also called as sanity testing, skin testing, driven testing, verification testing why we do smoke testing?

- ① Just to ensure that product is testable.
- ② To catch bugs in basic feature and send it to development time so that development time is sufficient to fix it.
- ③ Just to ensure product is launched satisfies properly ④ not
- ④ we do smoke testing in beginning.

Test points:-

- when we are doing smoke testing we do only positive testing.
- Here we test only basic & critical features.
- Hence we take basic feature and test its important scenarios.
- when ever the build comes to the customer

→ He also does smoke testing
→ when the product is installed in production, we do smoke testing to ensure product is installed properly.

④ not.
→ also it acts as a self check for software.

Adhoc testing

→ Testing the application randomly is called adhoc testing. (It is also called as monkey testing or Gorilla testing).

⑤ Why we do adhoc testing?

⑥ end user uses the application randomly and he may find the defect, but by engineer as the application systematically so, he may not found the same defect.

→ In order to avoid these scenarios, test engineers should ~~not~~ go for the application to test randomly (behave like end-user)

→ developer team, look at the requirement and build the product of testing team of look at the requirements and do the testing.

→ If we do not catch many bugs.

NOKIA

They think every thing works fine. In order to avoid this we do adhoc testing.

⑦ apart from testing value we don't follow the requirements.

⑧ when we do adhoc testing?
or ST FT IT ST Adhoc Testing if it had time then only go for adhoc testing.

→ When the product is stable then only we go for adhoc testing.

→ When the feature is new we will go for adhoc testing.

→ While doing smoke testing we don't do adhoc testing because, if we do adhoc testing, we don't get time to test basic features.

→ When ever we are free, after testing the product according to the requirement, if some time left out, then we should spend time in doing adhoc testing.

→ If we get too many ~~the~~ scenarios, note down the scenarios and execute of when ever you get time.

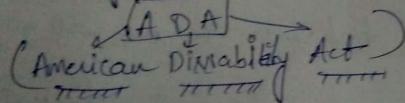
Reliability Testing :-

Testing the functionality of an application continuously for certain period of time is called Reliability Testing.

Recovery Testing :-

Testing the application to check how well it recovers from crash or disaster is called Recovery Testing.

Accessibility Testing :- (ADA)



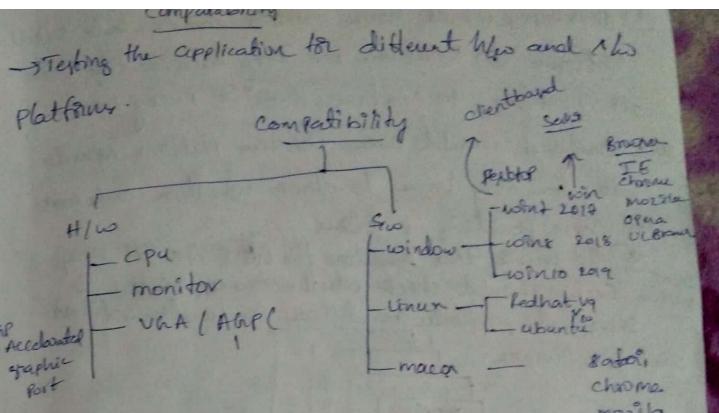
Testing the application for physically handicapped persons is called accessibility Testing.

Exploratory Testing :-

Testing the application without following any formal document (requirement).

- ① CFT, BT (IT), ST

NOKIA



→ Testing the application for different H/Ws and S/Ws platforms.

Compatibility

→ Testing functionality of an application for different H/W & S/W platforms is called a Compatibility testing.

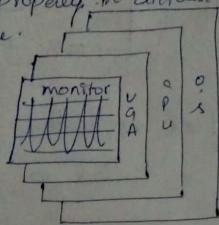
- ② Why we do compatibility testing?

Common code → should run on all O/S.

③ Developers develop the application in one platform if test engineers, test the application in same platform, when it is launched to the business. If customer might use application in different platform, because of that certain feature may not work, that spreads the bad name in the market. and the number of customer, who buys this product will reduce, to avoid such situation we should do compatibility testing.

- ④ To check whether feature are working correctly in all the platform, we do compatibility testing.

- (3) Developers would have written common code for all platforms. But browser we may have to test it on all platforms, to confirm that it really works.
- (4) Developers would have written platform specific code. we may have to check whether code works in corresponding platform.
- (5) diff. also & this renders the GUI, in diff ways. So we should check whether our application is rendered properly in different combination of hardware & software.



→ Return on Investment (ROI)

when we do compatibility testing?

- (1) when the product becomes stable on base platform, then we think of trying the application in different platforms.

Compatibility testing

Hardware compatibility testing :-

Testing the functionality of an application in different hardware environment.

NOKIA not test application in different
hardware like:

- (1) different processor.
- different speed.
 " make (Intel, AMD).
- " Bit size (64 bit, 32 bit)
- (2) different Mother Boards
 " make (intel, mercury).
- different VGA Cards.
- different monitor with different resolutions.
- usability Testing

Testing the user-friendly way of an application is called usability testing.

> let us consider an example such that we have two applications A & B which are different while doing the same job. we see which one is user friendly.

given below are some of the parameters we look into for testing.

- (1) Speed, help.
- (2) navigation should be simple.
- (3) compatibility.
- (4) look & feel.
- (5) features.
- (6) location of Components.

One important parameter other than the above set parameters is "effort needed to learn the application".

Concl.

software testing ② we have to understand what user is looking for ③ by to understand.

Let us take different cases here ① Since we understand software ④ or ⑤ by. It becomes user friendly. Suppose look & feel is not good for ⑥. In this case although we understand ⑦ we say we cannot say that

⑧ it is user friendly.

⑨ therefore we look into many parameters before we say user friendliness of a software.

⑩ what is look and feel?

⑪ The application should be such away that it should be pleasant looking.

⑫ How to conduct usability testing?

⑬ If we derive the checklist, we don't derive a checklist we may miss some feature in the application.

Example of an checklist for an application

⑭ Whenever we click on link.

(checkboxes) → Home → contact
→ about → privacy

⑮ for the application, one of the checklist which color of already checked should be changed to

NOKIA

⑯ all pages should have alt tags

- ⑰ all the pages should have back to home page
- ⑱ login feature should have forgot password link
- ⑲ If we have the above checklist we can derive a many checklist as possible based on the applicability of product.
- ⑳ While deriving checklist we should derive a common checklist which can be executed for all the pages.
- ㉑ There is another case where the customer gives the checklist to the application.

Test cases

- ㉒ what are the drawbacks if we test the application seeing the requirement?
 - ㉓ there will be no consistency in test execution.
 - ㉔ quality of testing over the period of time depends on ⑤ memory power of test engineer.
 - ㉕ quality of testing depends on mood of test engineer.
 - ㉖ quality of testing differ from person to person

㉗ if engineers are more experienced they derive more scenarios, if engineers are less experienced than they derive less scenarios.

what is test case?

It is a document which contains all possible scenarios for a specific requirement.

Text cases containing diff test cases like :-

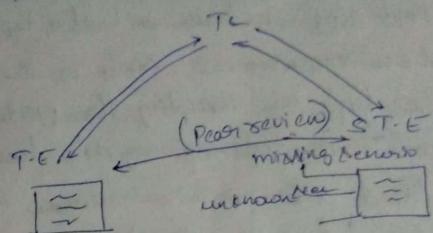
- ① step no.
- ② description
- ③ input
- ④ expected output
- ⑤ actual output
- ⑥ status
- ⑦ comments

When do we write the test cases?

- ① After when developer building the product the tester will start writing the test cases.
- ② When developer adds a new feature, tester will write test cases for new feature.
- ③ When developer modify the feature, tester will modify the test cases.
- ④ When developer delete the feature, tester will delete the corresponding test cases.
- ⑤ When developer complete the product and giving build to tester, tester will start testing the product according to the test cases written.

NOKIA

Test cases review process



Process of reviewing test cases :-

- ① Developers start building the product.
- ② Tester will start writing the test cases.
- ③ Once he or she finishes writing all possible test cases, tester will give test cases to his/her TL through email.
- ④ Now TL will assign them test cases for review to Sr. T.E.
- ⑤ Once the Sr. T.E. reviews and gives his/her comments or addition or modification or missing scenarios.
- ⑥ Now, T.E. should correct all the mistakes and get reviewed by Sr. Test engineer, the senior T.E. should review again and check whether all the connections are made and send it to team lead for approval.

Q) Why we write test cases?

- ① we write test cases, to have a better test coverage. when requirement comes in the development are busy in building the product at the same time test engineer are free so, they identified all possible scenarios and documented. when the built long we can spend time in executing the scenario because of this, no of scenario that you are covering is more.

- ② To have consistency in test execution. that means if we document the scenario we can make sure that we are executing all the scenario in all test cycles. ③ Sprint 0, release.

- ③ To depend on process rather than the person
- ④ To avoid training to every individual on the product.

- ⑤ Test case the only document which acts like a proof for customer, development team and management team, so, that we can tell them to the team that we have covered all possible scenarios.

- ⑥ Test case acts like a key document for writing the automation scripts. if you prefer the test case and write automation script we

NOKIA

can ensure that same kind of coverage in there even in automation script.

- ⑦ if you have documented the test case you don't have to remember the scenario.
- ⑧ if you have documented the test case. test case execution will happen in a very organized way.
- ⑨ if you have written the test case. time taken to execute the test case be very less.

Format of Test case sheet

TestCaseName: CBO - Amount transferfield
Requirement no.: 3.0.3.2, 3.0.3.1, 3.0.3 - CBO

Severity: Critical

Test Data:

Pre-condition:

Test case type: functionality Brief Description Verify amount
text field.

Stepno	Description	Input	Expected Result	Actual Result	Status	Comments
1)	Enter -ve Integer value ^{now} into the textfield	-100	Should throw appropriate message			
2)	Enter char into the Amount textfield	hundred rupees only	Should throw appropriate message			

Author: chunky

Reviewed by: Pinky

Approved by: Pinky

Approval date: 10/10/2018

Test Case Design Techniques:-

→ Error guessing.

→ Equivalence partition

→ Boundary Value analysis.

① Error guessing :-

② guess all possible data (s) except we ignore the exec. based on experience Intuition

③ Intuition

④ By requirement.

Ex:- -100, Blank, characters etc.

② Equivalence partition

Prerequisite

Requirements

Practically

If I/P is range
Values then design
TC for 1 valid value
and 2 invalid values

If amount > 100.200
deduct 10.2 transfer

" 200 - 300

deduct 2% E transfer

" 300 - 4000

3% E transfer

" 400 - 5000

4% E transfer

deduct 2% E transfer

" 500 - 6000

5% E transfer

" 600 - 7000

6% E transfer

" 700 - 8000

7% E transfer

" 800 - 9000

8% E transfer

" 900 - 10000

9% E transfer

" 1000 - 11000

10% E transfer

" 1100 - 12000

11% E transfer

" 1200 - 13000

12% E transfer

" 1300 - 14000

13% E transfer

" 1400 - 15000

14% E transfer

" 1500 - 16000

15% E transfer

" 1600 - 17000

16% E transfer

" 1700 - 18000

17% E transfer

" 1800 - 19000

18% E transfer

" 1900 - 20000

19% E transfer

" 2000 - 21000

20% E transfer

" 2100 - 22000

21% E transfer

" 2200 - 23000

22% E transfer

" 2300 - 24000

23% E transfer

" 2400 - 25000

24% E transfer

" 2500 - 26000

25% E transfer

" 2600 - 27000

26% E transfer

" 2700 - 28000

27% E transfer

" 2800 - 29000

28% E transfer

" 2900 - 30000

29% E transfer

" 3000 - 31000

30% E transfer

" 3100 - 32000

31% E transfer

" 3200 - 33000

32% E transfer

" 3300 - 34000

33% E transfer

" 3400 - 35000

34% E transfer

" 3500 - 36000

35% E transfer

" 3600 - 37000

36% E transfer

" 3700 - 38000

37% E transfer

" 3800 - 39000

38% E transfer

" 3900 - 40000

39% E transfer

" 4000 - 41000

40% E transfer

" 4100 - 42000

41% E transfer

" 4200 - 43000

42% E transfer

" 4300 - 44000

43% E transfer

" 4400 - 45000

44% E transfer

" 4500 - 46000

45% E transfer

" 4600 - 47000

46% E transfer

" 4700 - 48000

47% E transfer

" 4800 - 49000

48% E transfer

" 4900 - 50000

49% E transfer

" 5000 - 51000

50% E transfer

" 5100 - 52000

51% E transfer

" 5200 - 53000

52% E transfer

" 5300 - 54000

53% E transfer

" 5400 - 55000

54% E transfer

" 5500 - 56000

55% E transfer

" 5600 - 57000

56% E transfer

" 5700 - 58000

57% E transfer

" 5800 - 59000

58% E transfer

" 5900 - 60000

59% E transfer

" 6000 - 61000

60% E transfer

" 6100 - 62000

61% E transfer

" 6200 - 63000

62% E transfer

" 6300 - 64000

63% E transfer

" 6400 - 65000

64% E transfer

" 6500 - 66000

65% E transfer

" 6600 - 67000

66% E transfer

" 6700 - 68000

67% E transfer

" 6800 - 69000

68% E transfer

" 6900 - 70000

69% E transfer

" 7000 - 71000

70% E transfer

" 7100 - 72000

71% E transfer

" 7200 - 73000

72% E transfer

" 7300 - 74000

73% E transfer

" 7400 - 75000

74% E transfer

" 7500 - 76000

75% E transfer

" 7600 - 77000

76% E transfer

" 7700 - 78000

77% E transfer

" 7800 - 79000

78% E transfer

" 7900 - 80000

79% E transfer

" 8000 - 81000

80% E transfer

" 8100 - 82000

81% E transfer

" 8200 - 83000

82% E transfer

" 8300 - 84000

83% E transfer

" 8400 - 85000

84% E transfer

" 8500 - 86000

85% E transfer

" 8600 - 87000

86% E transfer

" 8700 - 88000

87% E transfer

" 8800 - 89000

88% E transfer

" 8900 - 90000

89% E transfer

" 9000 - 91000

90% E transfer

" 9100 - 92000

91% E transfer

" 9200 - 93000

92% E transfer

" 9300 - 94000

93% E transfer

" 9400 - 95000

94% E transfer

" 9500 - 96000

95% E transfer

" 9600 - 97000

96% E transfer

" 9700 - 98000

97% E transfer

" 9800 - 99000

98% E transfer

" 9900 - 100000

99% E transfer

" 100000 - 101000

100% E transfer

" 101000 - 102000

101% E transfer

" 102000 - 103000

102% E transfer

" 103000 - 104000

103% E transfer

" 104000 - 105000

104% E transfer

" 105000 - 106000

105% E transfer

" 106000 - 107000

106% E transfer

" 107000 - 108000

107% E transfer

" 108000 - 109000

108% E transfer

" 109000 - 110000

109% E transfer

" 110000 - 111000

110% E transfer

" 111000 - 112000

111% E transfer

" 112000 - 113000

112% E transfer

" 113000 - 114000

113% E transfer

" 114000 - 115000

114% E transfer

" 115000 - 116000

115% E transfer

" 116000 - 117000

116% E transfer

" 117000 - 118000

117% E transfer

" 118000 - 119000

118% E transfer

" 119000 - 120000

119% E transfer

have followed do not elaborate all the steps.

Lesson 6:-

when we write test cases we should always imagine the application.

Lesson 7:-

If we organize the scenarios properly total no. of steps will be reduced.

Lesson 8:-

Some company might use input column (I) for companies will remove actual column.

Lesson 9:-

If we cover scenario (P.T) don't cover same scenario in (D.T). if we cover scenario in (D.T) don't cover scenario in (S.T).

(v) what is the approach to write functionality test case?

Case 1:-

① go to the body of test case.

② start with the navigation steps.

③ take first page and start writing test case.

④ start with valid data.

⑤ cover the error guessing scenarios.

⑥ cover all equivalence partitioning scenarios

⑦ cover all boundary value analysis.

⑧ take second component repeat 3a, 3b, 3c, 3d.

(vi) on what basis they assign the test case for 2nd iteration reviewer?

1. There is a fellow working on the project.

2. There is a fellow worked on the same module in the previous project.

3. There is a fellow working on the project since beginning, and knows every product.

4. The is a fellow who is very responsible, he will understand the requirement very fast, and identifies more ~~more~~ mistakes.

(vii) How do they expire review dog his/his job?

1) The TC should assign primary and secondary reviewer.

2) The TC also should randomly review and find mistakes.

3) Intentionally introduce the mistake and give it to the reviewer and check whether it is caught by reviewer or not.

(viii) Test case review ethics?

1) always reviews the content not the author.

2) while reviewing spend time in identifying the mistake, not in finding the solution to it.

3) even after the review, if there are any mistakes both author & reviewer.

NOKIA

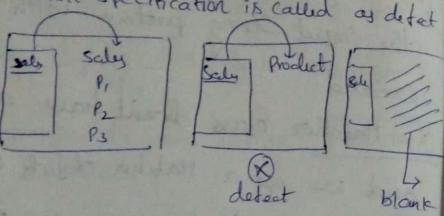
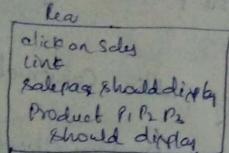
- ④ why we review test case?
- A) ① we will look into the header of the test case
② understand the requirement which
the test case is written.
- ③ then go to the body of the test case try to
find, following both logic.
- ④ Basic scenario -
- ⑤ repeated "
- ⑥ wrong scenario.
- we will check whether the scenario are
organized and you test case should have
logic flow of execution.
- ⑦ we will check whether it is simple to understand.
so that I am given to new engineer he/she
should be able to execute that without
asking any question.
- ⑧ look up to header of the test case.
- ⑨ check all the attribute (name) is not
⑩ check whether content in the attribute
are correct or not.
- ⑪ check whether test case format is template
is correct & if it should be able to the
standard defined in the project

Defect :-

Software Testing

any feature which is not working acc to customer requirement

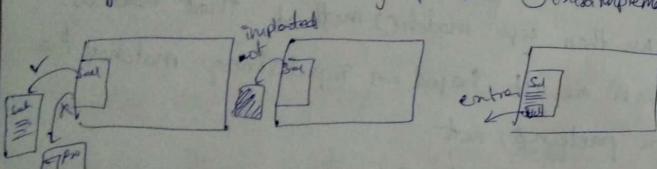
3) deviation from requirement specification is called as defect



detected

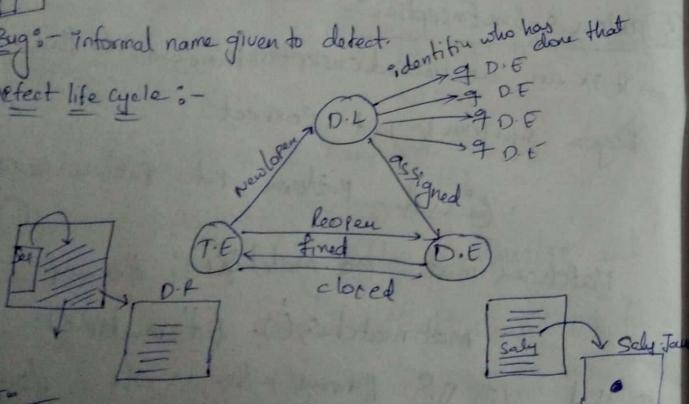
blank page

- ① Wrong implementation
- ② Missing implementation
- ③ Extra implementation



Bug :- Informal name given to defect.

Defect life cycle :-



T.E :-

Test engineer points the defect.

will prepare defect report.

Send it to development lead as new.

D.L :-

Send it to development lead

identifies developer who did the mistake.

change the status to Assigned.

Send it to development engineer.

D.E :-

Development engineer read the report and understand the problem.
Go with source code and fix the bug.
Change the status to fixed.

Send the report to test engineer and cc to Development lead.

T.E :-

He will read the report and understand the problem.

Retest the bug if the bug is fixed, he will change the status to closed.

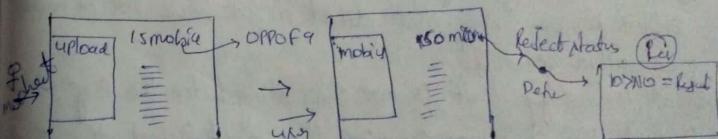
Otherwise change the status to re-open and send report to Development lead & Development engineer.

Reject status :-

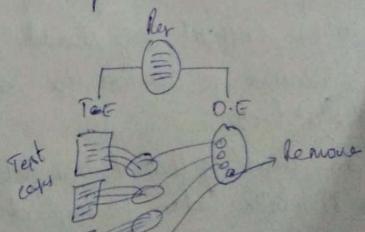
Test engineer finds the defect and sends it to developer and developer say that it is a feature and change the status to reject.

Reason for Reject status :-

- ① Because of misunderstanding during the requirements.



- ② Because of retesting old requirements.



③ When you find extra feature as a bug and send it to developer, developer might reject it. In such a case tester should explain it and ask developer to fix it.

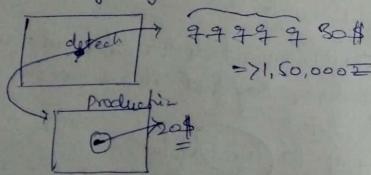
④ Because of wrong installation & configuration of a system there might be a defect, when you send it developer, they say that you have not installed properly.

⑤ What is Cannot be fixed status?

① Developers are accepting that it is a defect, but they say that they are not in a stage of fixing it, because of few reasons. They are:-

② Cost of fixing the bug is more than cost of bug (loss in the business because of having bug in the software).

Cannot be fixed



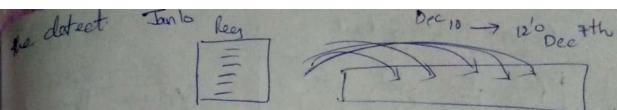
③ If there is a bug present in the root of the product and if it is minor fixing that defect might have lot of impact on other features.

④ If technology itself is not supporting to fix the bug and if that bug is minor or major they might reject the bug. That means if bug is minor or major they can say cannot be fixed, but if it is critical they should come up with alternative solution to develop the same feature.

⑤ What is Pending status?

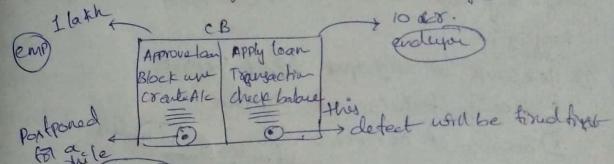
A) Developers are accepting that it is a defect but they are fixing it little later because of two reasons. (i) There is a minor (ii) major bug developer are not having time to fix it, in such a case they might postpone.

NOKIA



⑥ When you send bug to developer they say that automator is expecting lot of changes in the same feature, so better postpone the work, until we get the clarity of what they are changing.

⑦ If there is a minor bug ensure to internal user.



⑧ What is Duplicate status?

⑨ If you send a defect which is already sent by some one else then developer say duplicate reason :-

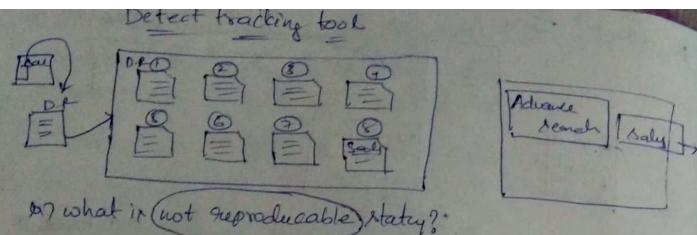
① Because of common feature.

② Someone else might come and catch the defect in your module and send it to developer. If you find same bug and send it to developer, they say Duplicate.

③ Old test engineer have found lot of bugs and send it to developer, and some bugs are still pending. If you join same old project and send same old bug, they change status to Duplicate.

⑩ How to avoid duplicate bugs

A) Before we prepare the report and lock the bug should make sure that it is not duplicate. So, we should search for duplicate bugs in the tool by entering certain keywords (In this example salary in a key word).

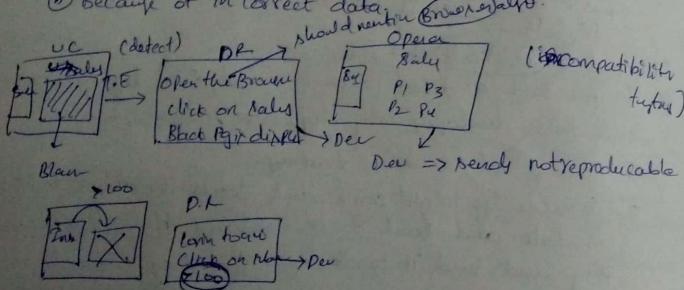


Q what is (not reproducible, static)?

A) Test engineer will be able to see the defect but developer will be not able to see the same defect in such a case they'll say that it is not reproducible.

Reason :-

- ① Because of improper detect report
- ② Because of incorrect platform.
- ③ Because of incorrect data.



- ④ Because of inconsistency bug, sometimes it appears some time it disappears.

RFE/CR:- it is a problem which is not a part of requirement it is called as Request for Enhancement / Change Request

NOKIA

RELEASE NAME: Detect Report

BUILD NAME:

STATUS :-

SEVERITY: New/Open, Assigned, Fixed, Closed/Reopen

PRIORITY: Blocker, Critical, Minor, Major.

High, medium, Low

Test DATA :- UN<abc>, PW=123

TEST ENVIRONMENT:- W17, W8, W10

TEST-CASE-NAME:- Gmail SI-mail-link

Found By :- Your NAME.

DETAILED DESCRIPTION:- S.I. is not displaying in SI Page

DEFECT ID:- unique code given by the tool.

BUILD ID :- b04

BRIEF DESCRIPTION: following all the procedure to reproduce the bug.

- ① Login to Gmail.
- ② Click on Compose Link.
- ③ Enter all valid data & click on send button.
- ④ Click on S.I. link

EXPECTED RESULT:- S.I. should display on S.I.PG

ACTUAL RESULT:- Not displaying

- ① Find defect
- ② X duplicate
- ③ Prepare D.R. (detect report)
- ④ Send it to D.L. (Development lead)
- ⑤ Manage D.L.C (Defect life cycle).

UI-TP

Uname	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

→ Unique TP (Bug ID)
→ Shared in DB
→ Send it to DL
→ It should get updated

↳ (

<input type="checkbox"/> BugID	<input type="checkbox"/> Assign
for a devt handle	
<input type="button" value="Submit"/>	