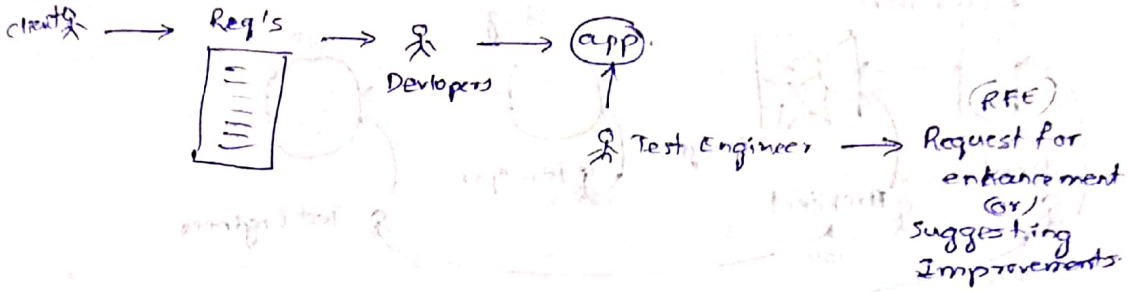
12/9/17

Software Testing

→ Testing the software
→ Finding the bugs
→ verification and validation of an app.
→ checking the quality
→ checking the functionality
→ checking the Usefriendlyness
→ checking compatibility
→ " s/w as per Requirements.

① Checking application



Software Testing

→ It is a process of checking an application as per the requirements and suggesting improvements (or) enhancements if possible.
→ Any s/w is developed for a bussiness. If the s/w is not working properly then there can be a huge loss to the business.
→ Some softwares are used in machine critical (or) life critical applications. hence, if the s/w fails then there can be loss of life in addition to the loss in business.

Risk

→ A risk is any event which has not happened and it may not happen in future but if it happens then there can be a negative impact on the bussines of the application.

Risk Analysis

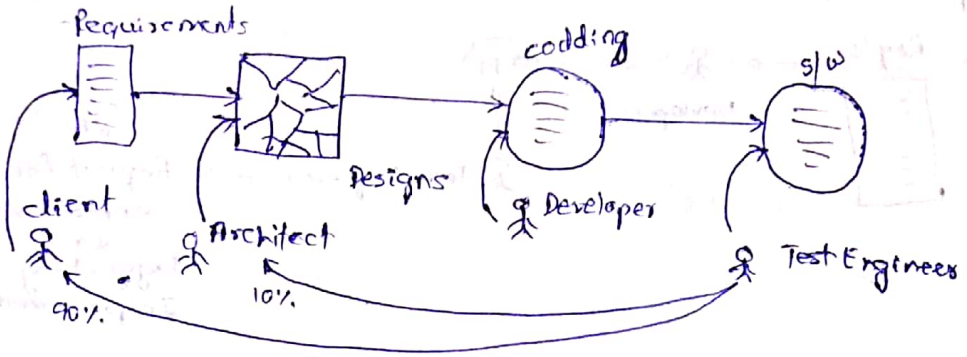
→ while using any software, analysing the events which might have negative impact (or) which might lead to failure of the software is called as risk analysis.

ex:- Lift
- Risk analysis - If it over weight it will strucks (or) power failure will happens. It will stuck
- counter measures - If it is over weight lift should not move till it below the rated weight
  - If it get failed or power fail lift has to reach at a perticular floor and should open the door automatically.

Scanned by CamScanner

Flipkart → Risk analysis
- Normal days it will work properly, in any seasons like festivals because of more users on that days application may get fails, misbehaves, struck or crashes, etc.
→ Counter measures - If app get overloaded, payments get struck slow down the app, overloaded conditions should check.

Aviation system ← Risk analysis - Risks involved in application software working misbehaving, turbulance system problems in critical wheather conditions. leads to crash (or) problems in land (or) takeoff.



Error, failure, fault

Errors → Mistake done by user in using the software is called as error.
→ when ever there is an error it's not because of flaws in the software but it's due to lack of knowledge in using the software or an intentional mishandling of the software.

Fault / Defect / Bug

→ whenever there are any flaws present in the code (or) the design (or) the requirement document then it is called as fault/defect/Bug.

Diff b/w Bug and Defect

→ A Bug is nothing but a defect present in software
→ A defect can be present in both software and hardware. But the defects essentially present in the software is called bug.

Failure

→ Errors/ Defects/ Bugs can lead to work the software in an unexpected way (or) because of this presence software might not work such situation is called as failure.

Scanned by CamScanner

Classification of Bugs

As per the location of Bugs the Bugs can be classified in 3 types.

1. Bug in code
2. Bug in design
3. Bug in requirement.

1. Bug in code

→ when ever the developper does a mistake in creating the code because of which there are some bugs in the software then those bugs are called as bug in code.

→ The cost of fixing for these types bugs is very less since the rework is involved only in the code

2. Bug in design

→ Bugs in design are the result of flaws in design due to the mistake of architects.

→ The cost of fixing for these types of bugs is relatively higher than bugs in code, because of more rework involved in both the design and code

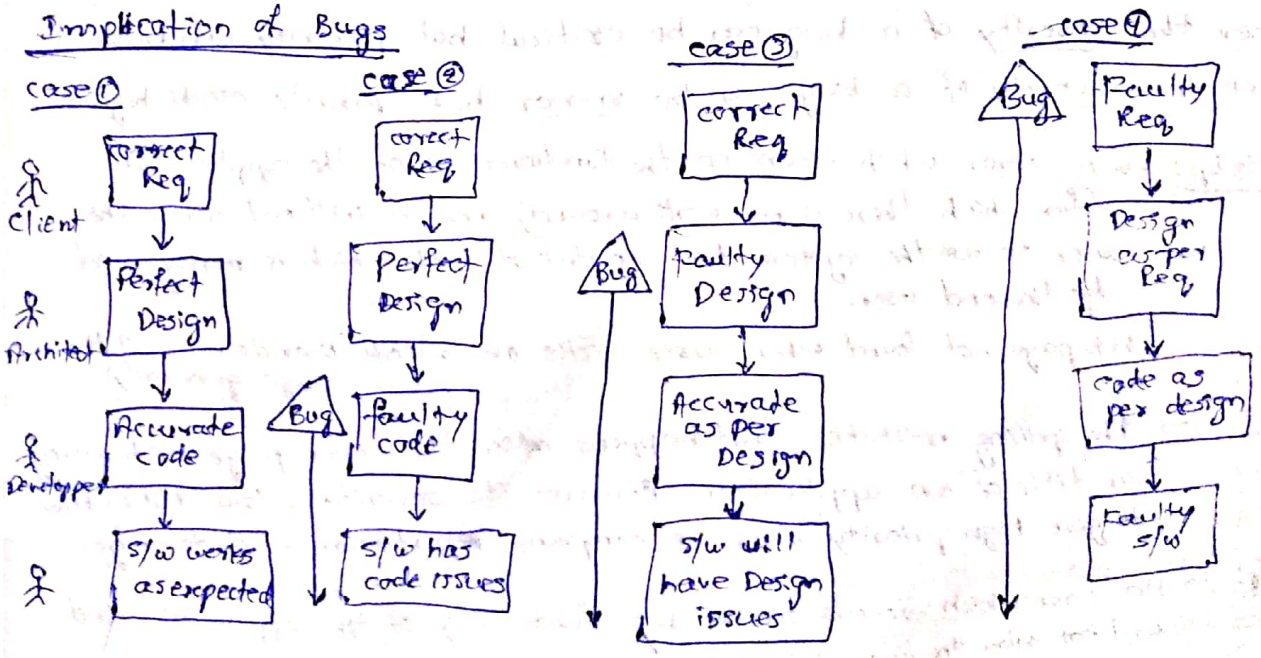
3. Bug in Requirement

→ Bugs in requirements are due to faulty requirements given by customer

→ These types of bugs are costliest since the rework involved is very high and the changes have to be done in all the requirement documents, design features and code.

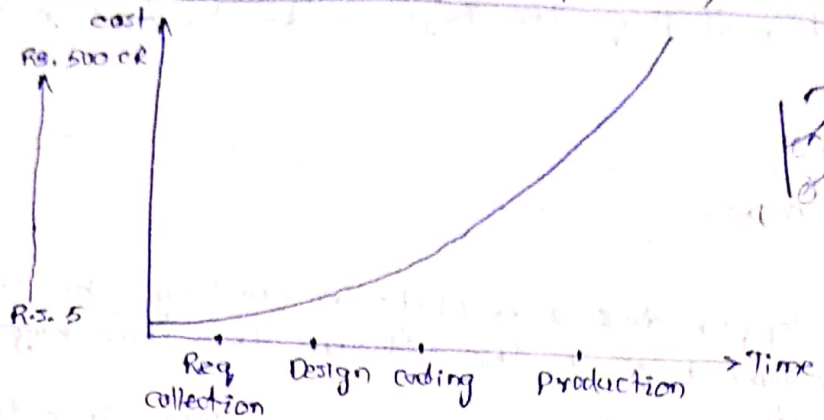
Note: Finding a bug in requirement is very difficut and it can be caught by the client in UAT (user acceptance testing) (or) by a very experienced testing engineer and who is also a domain expert.

Implication of Bugs



Scanned by CamScanner

Why the cost of fixing a bug increases exponentially with time?



→ The cost of fixing the bug increases exponentially because of the rework which also increases with time.

Severity of Bug

→ Severity of bug means how ~~fast~~ much impact on the bussiness of the application.

Types

① Blocker Bugs – It blocks us from conducting further testing
② Critical Bug – High impact on the bussiness of the application.
③ Major " – Moderate impact " "
④ Minor " – Low impact " " (color & spelling)

Priority of the Bug

→ It means how fast the bug should be fixed.

① High – Immediate delay
② Medium – Little delay
③ Low – At the end (or) next release

Assignment

① when the severity of a bug can be critical but priority can be low?
② when the severity of a bug can be minor but priority can be high?

Ans

A: ① HsLp: → As error which occurs on the functionality of the application (for which there is no work around) and it will not allow the user to use the system but on click of link which is rarely used by the end user.

→ Web page not found when user clicks on a link (user doesn't visit the page generally).

HpLs: → The spelling mistakes that happens on the cover page or heading or title of an application. Of course its severity is low but we have to give high priority otherwise company reputation will damage.

HsHp: → An error which occurs on the basic functionality of the application and will not allow the user to use the system.

eg: User is not able to login the application.

HP

Scanned by CamScanner

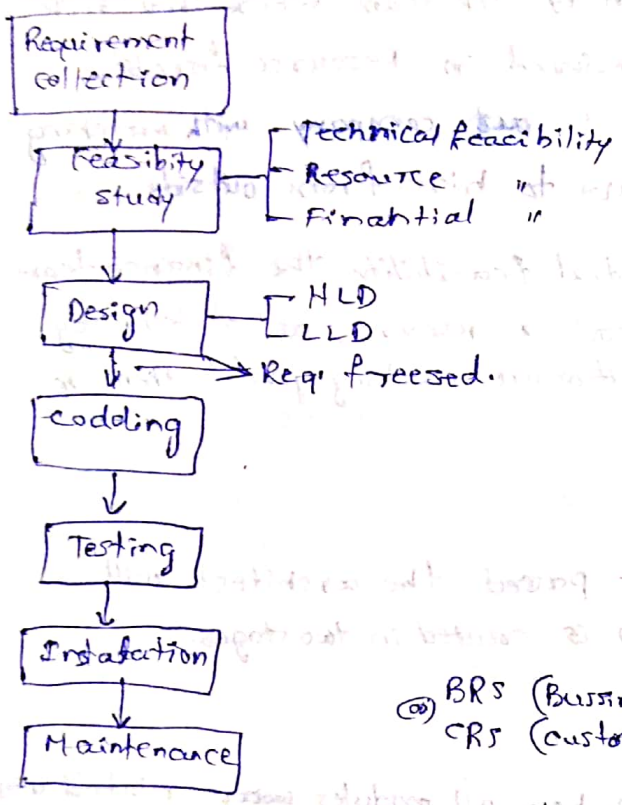
14/9/22

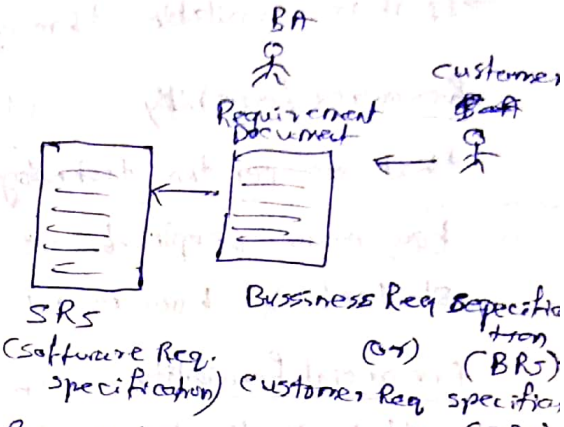
Software Development Life Cycle (SDLC)

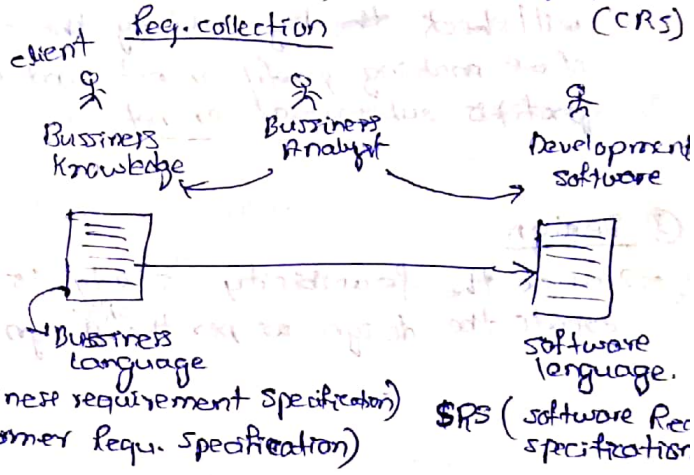
→ Software development cycle (SDLC) is a process to develop or create a software. Any SDLC is implemented to control the entire software development process.

→ Any SDLC should result in a very high quality software which meets (or) exceeds the customer expectations, reaches completion within time limits, works effectively & efficiently and it does easy to main and cost effective to enhance.

Water Fall Model







(or) BRS (Bussiness requirement Specification)
CRS (customer Requ. Specification)

→ Water fall model is a traditional model where the flow of work is vertically downwards like water fall.
Following are the steps in water fall model.

① Requirement Collection

The B.A collects the requirements from customer in bussiness language. and he converts into SRS which is in software language.

CRS - Customer Requirement specifications
↳ It is a document in which all the Req. of customer are specified and it also shows the working of bussiness in addition to the expectations from the software.

SRS - Software requirement Specification
↳ It is a document in which all the requirements for creating s/w is specified. Generally all the modules of s/w will be explained in addition to their functionalities.

Scanned by CamScanner

② Feasibility study:

→ Here we will check whether creating the s/w is feasible (or) possible (or) not this is done in 3 stages.

Technical feasibility – This is done by project managers, Architects and senior developers. here we will check

→ whether the technology available or not while creating s/w.

→ If it is available through what technology we can create it.

Resource feasibility – It is done by HR team, where they check that as per the technology mentioned in Resource feasibility, how many people do we have in our company with matching skillset & how many we have to hire from outside

Financial feasibility – In financial feasibility the finance team will check ~~the~~ by adding the cost of resource, cost of technology if we making profit or not and if we are making profit then the profit is substancial or not.

③ Design

→ once the feasibility study is passed the architect will create the design as per the SRS. design is created in two stages.

HLD (High level Design) – In HLD how all modules ~~were~~ related with each other and how the data flow happen between them is decribed.

LLD (Low level Design) – Here the functionality of each and every small small components ~~will~~ in each and every module will be defined.

→ once the design is ready all the requirements freased.

④ Codding

→ The codding is done by developper. The developpers will implaments low level designs ~~into~~ through their code to create all the components as per the functionality in all modules. once all the components are ready then all the modules will be integrated as per the high level design. Here requirements are freezed ~~after~~ after design.

Note

when ever two modules are integrated then data flow must happen b/w them.

Scanned by CamScanner

⑤ Testing

→ In water fall model testing is very small stage which comes after coding. Hence developer will only test the code if there is any bug they have to fix it.

⑥ Installation

→ Here the software is delivered to the customer and it is installed in the client location.

⑦ Maintenance

- Once the s/w is delivered and installed the customer uses the software for his/her business. while using the s/w they might find some bugs or they might ask for small changes like changing the background color or adding some extra functions these things can be done in this stage.

Note
→ For major changes the customer have to wait for the next release.
→ The entire time duration for req. collection till delivery of software is called one release.
→ The software which is released is called one released.
→ The software delivered in a release is the version of the s/w.

Advantages

① This model is simple to understand and easy to use.
② The cost of using this model is very less
③ Since the requirement changes are not allowed after design ~~changes~~ hence we get a very stable s/w.

Disadvantages

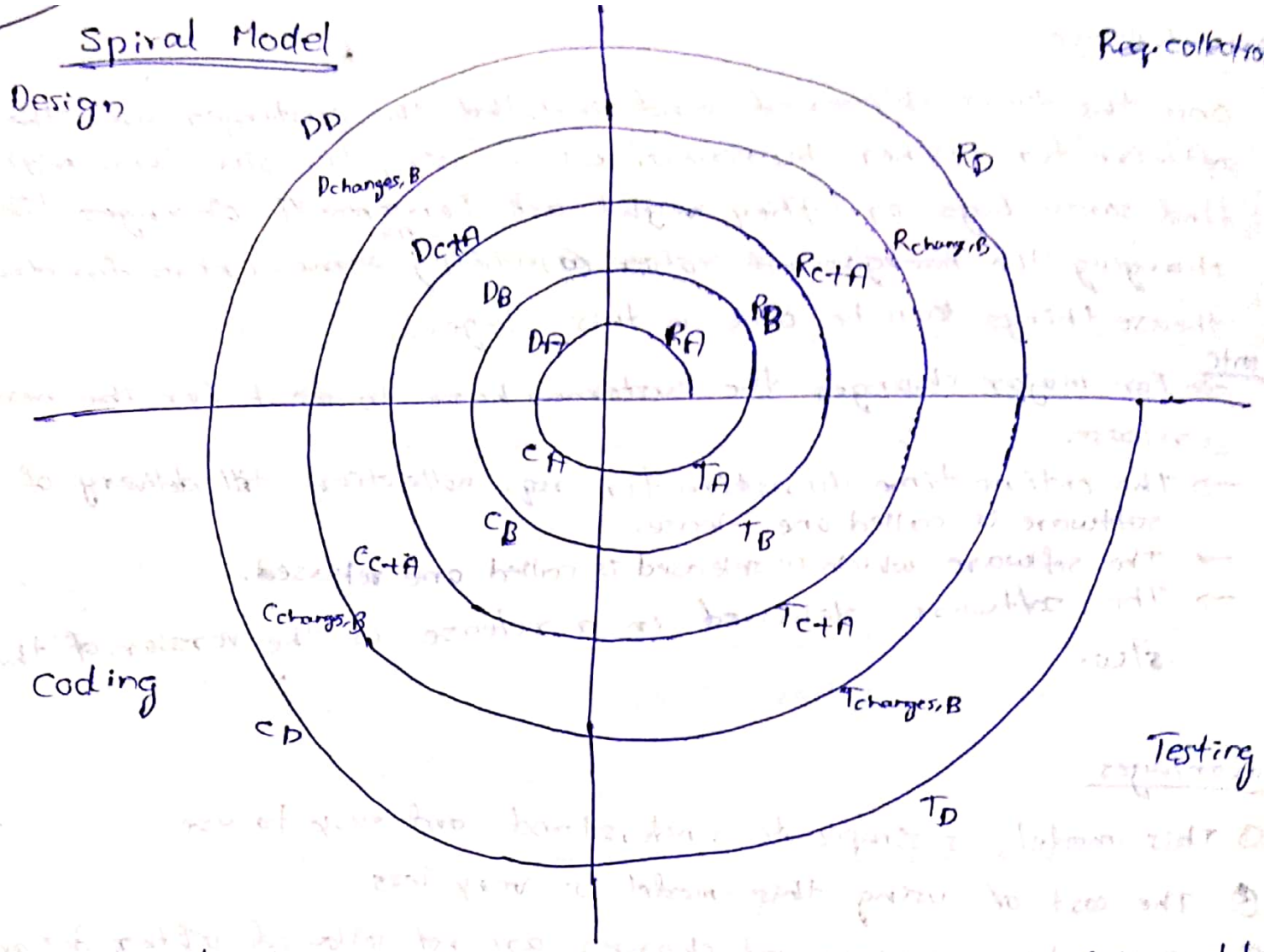
① Req. changes are not allowed which can make the customer unsatisfied and the s/w not suitable for the business.
② Since developers are only doing testing and testing is a very small stage which comes after coding hence the quality of testing activity will be very less.
③ Both req. & design are not tested hence if there is any bug in req. or design, a lot of rework will be involved.

Application

① For any short term project.
② For any simple proj where the req. changes are not frequent.

Scanned by CamScanner

Spiral Model.



→ Inorder to overcome the short comings of waterfall model the IT industry came up with a new model called as spiral mod

→ In spiral model the entire software development process is of step by step process where one module of the software is created at once.

→ For eg. if any software has 4 modules like module A, B, C, D then the process of software development will be as follows.

① In the first stage we will collect the req. for module 'A' then we will design module 'A' and after that we will implement the design into the code, while testing module 'A' we might find some bugs those bugs will be fixed then only we will start RB.

② After completion of module 'A' the Req. for module 'B' (RB) will be collected then, we will design module 'B' and implement design into code, while testing we will test module 'B', integrati

Scanned by CamScanner

between module A & B and then we will test module 'A'.

③ If the customer is asking for some minor changes then, since the minor changes involved less amount of rework hence we can do it with the next module. Hence we will collect the req for module 'C' along with the minor changes in module 'A'. After this we will create the design for module 'C' and update the design of module 'A' as per the changes. while coding we will first code for module 'C' then we will update the code of module 'A'. while testing we will test module 'C', minor changes in module 'A', intergration b/w AC, BC and AB. and then we will separately test modules and unchanged features in 'A'.

④ If the customer wants for major changes then since major changes involve ~~more~~ more work then we will come up with separate cycle for major changes in module 'B'.

→ At first we will collect all the major changes to be done as requirement and we will update all these change in the design of module 'B'. As per the updations ~~and~~ in design we will also update the code for module 'B'. once the code is ready we will first test all the major changes in module 'B' and after that we will test the integration b/w module AB, BC and AC and then we will test module 'C', all the unchanged features in module 'B' and test the module 'A' separately.

⑤ For module 'D' req for module 'D' can be done, D_D, C_D and then test module 'D', integration b/w AB, BC, CD, AC, BD and AD and we will separately test each modules individually.

Advantages

① Major & minor changes allowed

② stable application

Disadvantages

① Testing is not done in Req. & Design

② Developers only testing S/w.

Applications

① Applicable for long term projects.

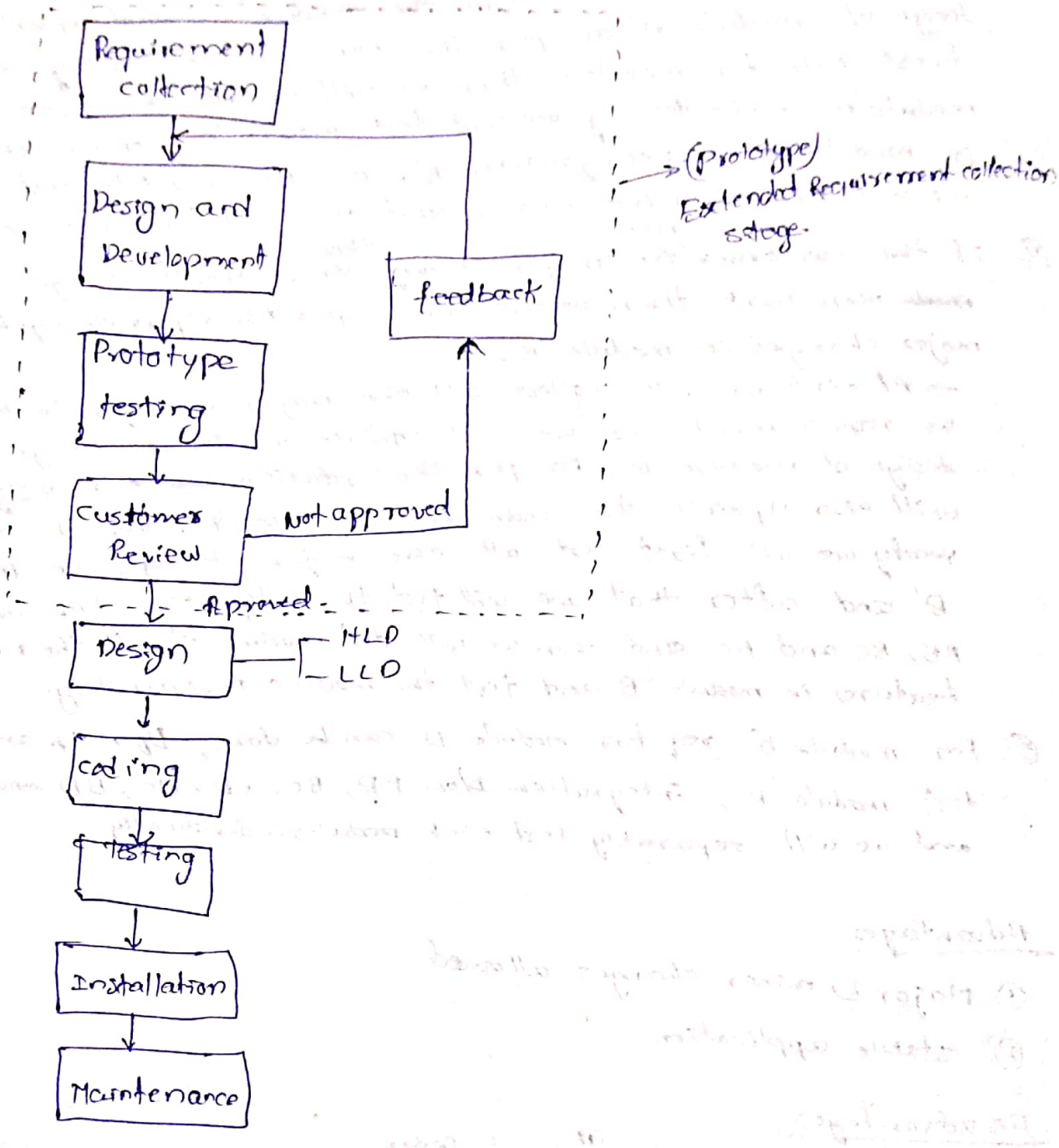
② Aplicable where there is a dependency b/w modules.

③ Applicable when customer is going to give requirement in stages.

Scanned by CamScanner

19/9/17

Proto Type Model



→ We use prototype whenever the customer is not sure about the s/w.

→ A proto type is a dummy model which will have all the components as in actual s/w but those will not be functional

→ The flow of work in prototype model is as follows,

① Requirement collection

In this stage the customer will sit back with the developpers, architects and the bussiness analysts and together they will try to propose a solution for the problem faced by the bussiness.

Scanned by CamScanner

This solution will be converted in the form of SRS.

③ Design & Development of prototype

→ In this stage the architects will create the design of the s/w and one HTML designer or web designer will create a dummy web app. through html codes. This dummy model will be our proto type.

④ Prototype testing

→ Here we will test if all the features are present as per the SRS and the design.

⑤ Customer Review

→ once the ~~software is~~ prototype is ready and tested we will send the prototype to the customer, the customer will check if the prototype suits his requirements or not.

→ If the prototype doesn't suit the requirement or the needs of the business then the customer will reject the prototype and gives his feedback. As per the feedback again the prototype will be created and it will be tested before ~~tested~~ sending it once again for customer review.

→ If the customer approves then the actual design, coding, testing installation and maintenance for s/w will take place.

Advantages

① the customers gets enough opportunity to make the requirement changes.

② The developers and the customer will have a clear picture about the look and feel and working of the actual software from the beginning itself.

Disadvantages

① There is increase in the cost of the project due to the creation of several prototypes.

② This is the traditional model hence testing is done by developers only hence the quality of testing is less

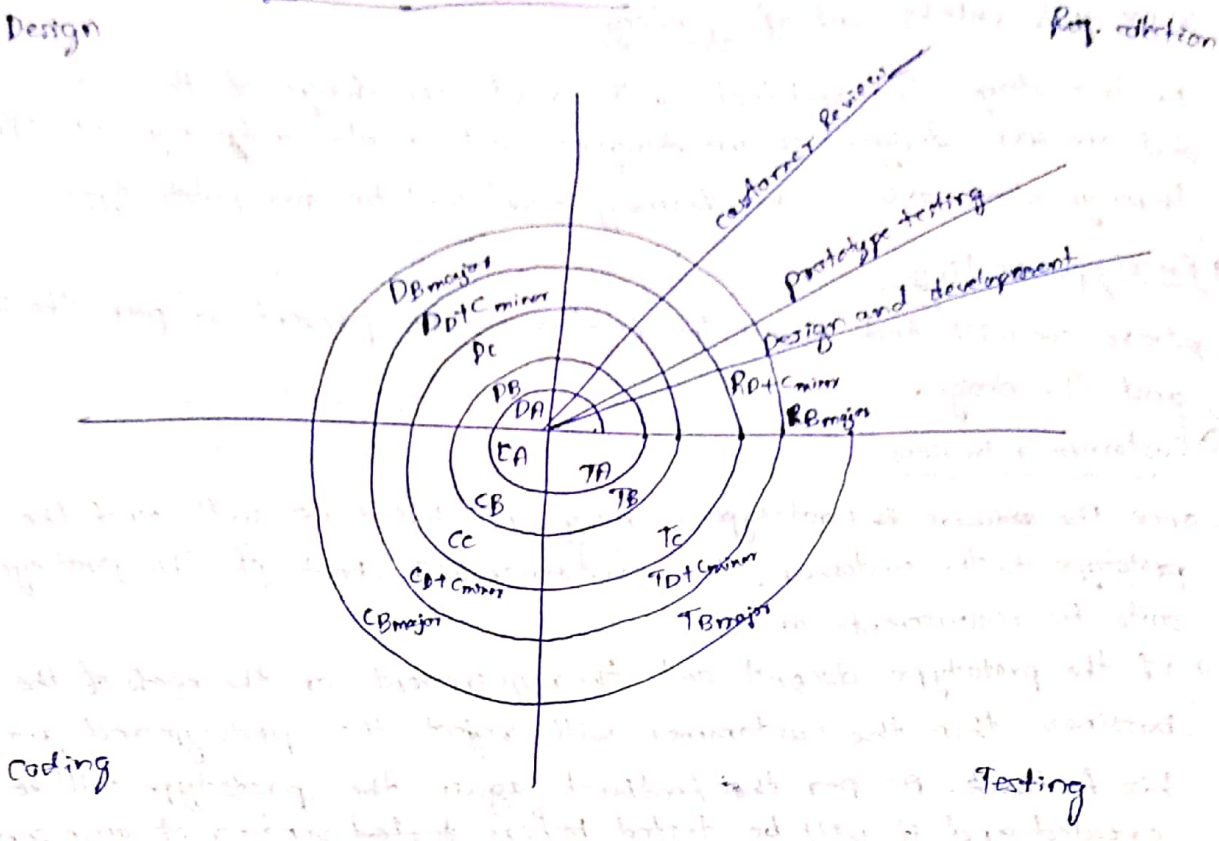
Applications

① Whenever the customer is not sure about the requirements

② Whenever the developer are new to the domain.

Scanned by CamScanner

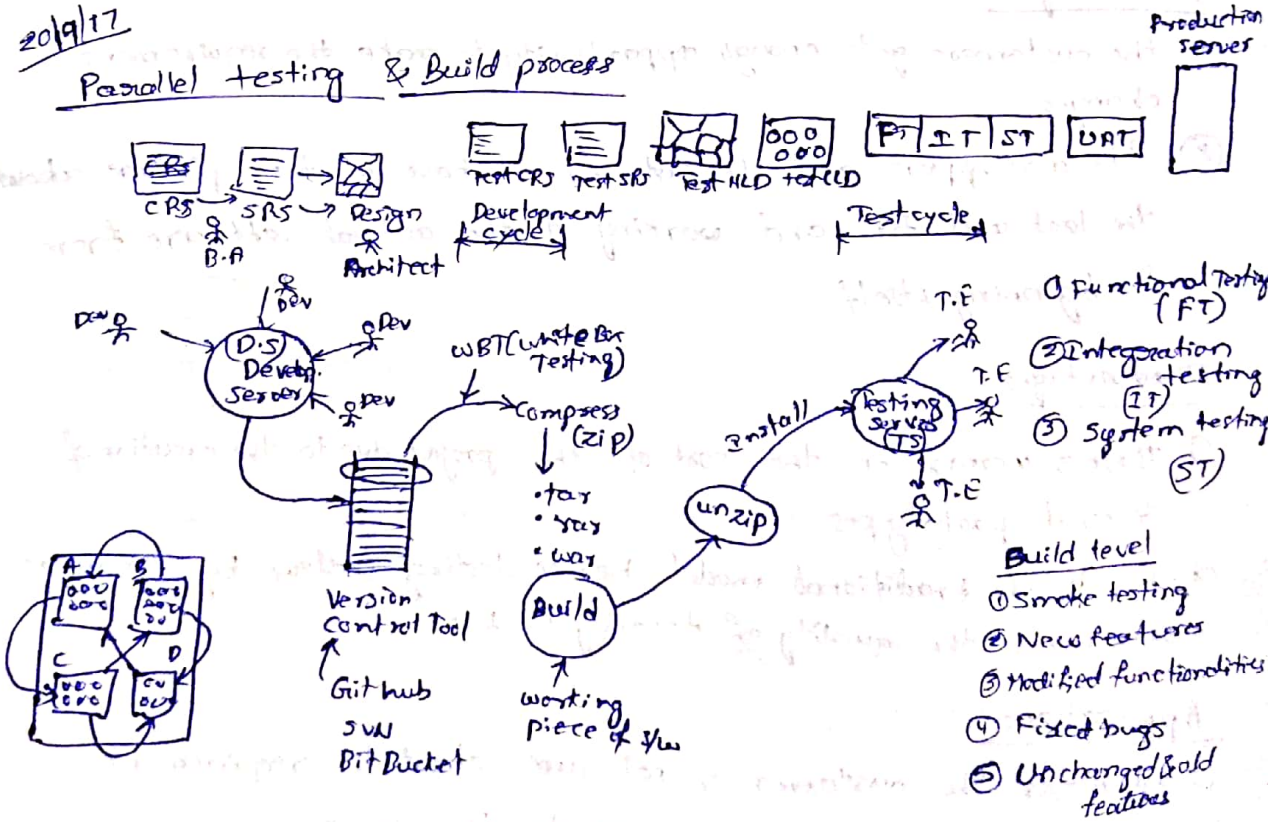
Spiral - Prototype Model (Hybrid Model)



→ The spiral prototype model is a hybrid model in which the advantages of both spiral and prototype can be achieved. We use this one where there is dependency b/w modules and customer is not sure about the end software.

20/9/17

Parallel testing & Build process



Scanned by CamScanner

Testing done at project level

→ In parallel testing process the development team and testing team works parallelly.

→ As soon as the CRS is received the CRS is tested for 3 things.

① Wrong req.

② Missing req.

③ Conflicts in req.

→ Once the CRS is tested and SRS is ready, the testing team will test the SRS for following things.

① If all the req. are converted from CRS to SRS

② If all the req. are properly converted from CRS to SRS.

→ After testing SRS, testing team will test the HLD and LLD to check if it is made inaccordance with the SRS.

→ Testing at development of build,

In project level:

(A)

→ In the initial stages of development, the dev. team will implement the LLD as LLD contains only the functionality of small small features and components present in each and every module. Hence the developpers will also create only the components and features in the modules at every build.

→ Thus the testing engineers can conduct functional testing (FT) at this stage

Def. Functional Testing:-

Testing the functionality of each and every component as per the requirement specifications is called functional testing

(B)

→ Once the LLD is implemented then the developpers will start integrating all the modules as per the HLD

→ When ever two modules are integrated then data flow must happen b/w them. Testing this data flow is called as integration testing.

→ At this stage the T.E will conduct integration testing.

(C)

→ once the functional and integration testing is passed then the testing engineer will test the end to end bussiness of the application called as system testing.

Scanned by CamScanner

→ System testing is done inorder to make sure that the bussiness of app. working properly or not. It is done only when all the features are created and all the integration are done as per the requirement specification.

At Build Level

→ As soon as the build received build can have new features, modified features, fixed bugs and old features.

→ In build level the process of testing is as follows.

① Smoke testing

→ It is the 1st type of testing which is done as soon as the build is received. If the smoke testing is passed then only we will accept build for further testing otherwise build will be rejected.

Def: Testing the basic and critical features of the application as soon as the build is received is called as smoke testing.

② Testing New features

If smoke testing is passed then we will first test the new features since there is a very high probability of getting bugs.

③ Testing modified features

Whenever there is chang request (CR) from customer then only a feature will be modified for its functionality. Since after modification it is not tested hence after testing the new features we will test the modified features.

④ Testing fixed bugs

☆ Testing happens through test cases and while ~~testin~~ execution of test cases we might find some bugs. If the dev. team claims that the bug is fixed then we will reexecute the same test case to make sure that the bug is fixed or not.

Scanned by CamScanner

Def: Re-testing

Re-execution of the same test case through which bug was found is called as re-testing.

⑤ Testing unchanged/old features

For testing the unchanged or old features we will conduct regression testing. since the unchanged features might be effected due to the changes done in the code of application.

Def: Regression testing

Testing the old features or unchanged features of app. due to changes is called regression testing.

⑥ Testing out of test cases.

Testing app. out of the testcases is called Adhoc testing many times the bugs cannot be caught through the test case execution because those bugs may be out of requirements Hence to find those bugs we conduct which is out of test cases.

Build Process

① The development team and testing team works on development server and testing server respectively.

② The dev. team will build the code and store it in version control tool for the entire dev. cycle.

③ At the end of dev. cycle they will take the latest version of code from the version control tool and conduct White Box testing.

Def: WBT

Testing the business logic of the code is called as white box testing.

④ After WBT the developers will take the code and compress into a zip. file and send it to the testing team. The testing team will unzip the code and instal it in the testing server before ~~testing~~ starting the testing.

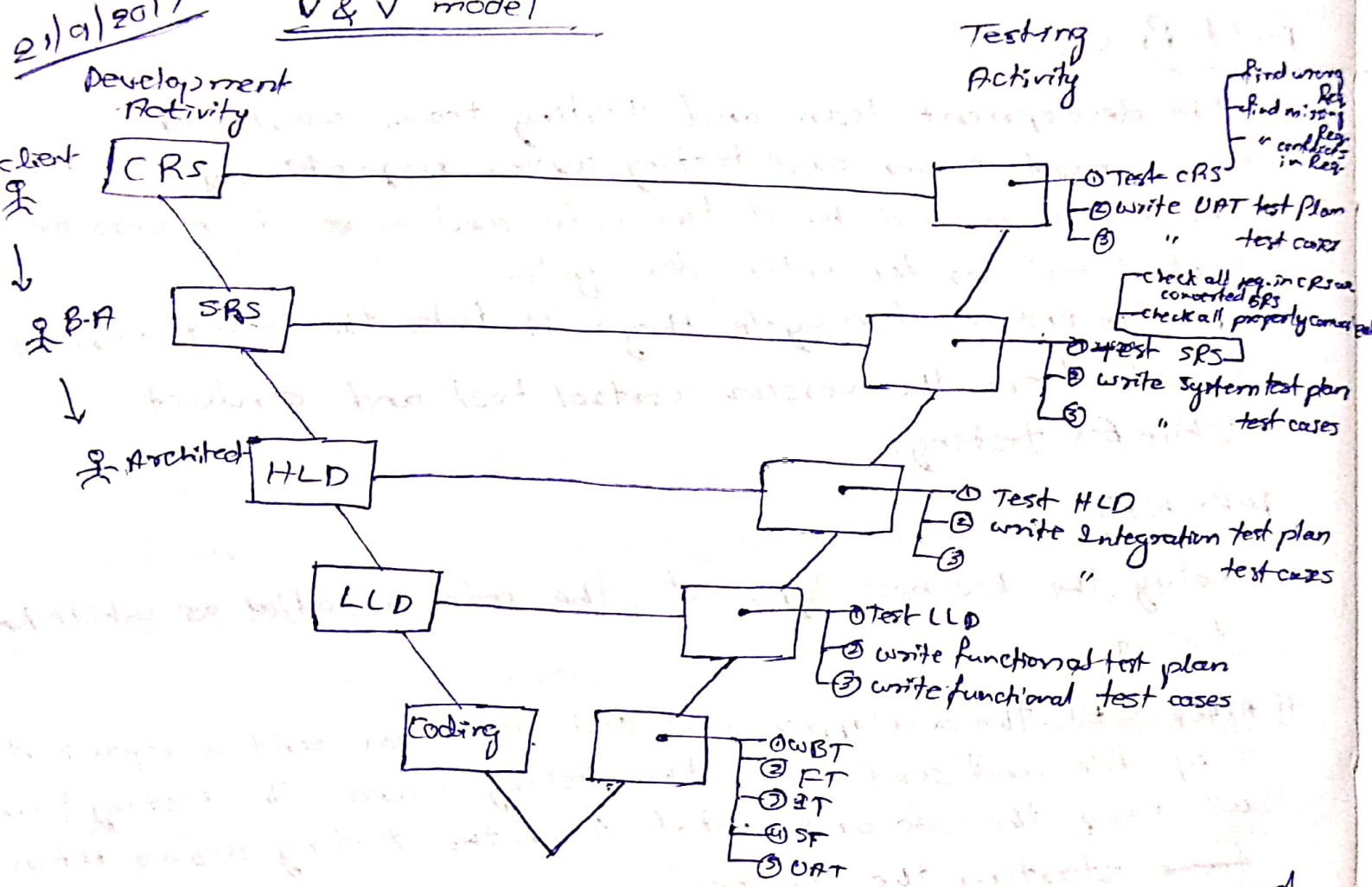
Def: Build

Build is of working piece of s/w which is compressed and given to the testing team further process of test execution.

Scanned by CamScanner

01/01/2017

V & V model



→ V & V model or verification and validation model is used whenever we require a very high quality product for some long term project (product).

→ Here the verification means checking if we are creating the right s/w suitable for the bussiness.

Scanned by CamScanner

→ Validation means checking if the s/w is carried properly or not.

→ For verification and validation we need to test the requirement, design, code and the s/w.

→ verification will involves checking the requirement and design while validation will involves testing the code and s/w.

The flow of this model is as follows.

① In the beginning the CRS is given to both the dev. team and testing team. The dev team starts converting CRS to SRS while the testing the testing team will test, Test RS
- ① Find wrong req.
- ② missing req.
- ③ conflicts in req.

② write UAT test plan
③ write UAT test cases

② As soon as the SRS is complete the dev. team will give the SRS to testing team and they will start creating the HLD for the application meanwhile the testing team will,

① Test SRS
- ① Check all req. present CRS are converted into SRS or not
- ② Check all req. in CRS are properly converted in SRS or not

② write system plan
③ write system test cases.

③ As soon as the high level design (HLD) is complete the dev. team will give it to testing team and the testing team will,

① Test the HLD
② write integration test plan
③ write integration test cases

④ As soon the LLD is complete the dev. team will give this to testing team and they will start creating the code for the s/w.

The testing team will,

① Test LLD
② write functional test plan
③ " " " test cases.

⑤ As soon as the code is ready the dev. team will conduct white Box testing (WBT) and give the s/w to the testing team.

The testing team will

① Implement functional test plan & execute Functional test cases to conduct functional testing.

② Implement integration test plan and execute integration test cases to do integration testing.

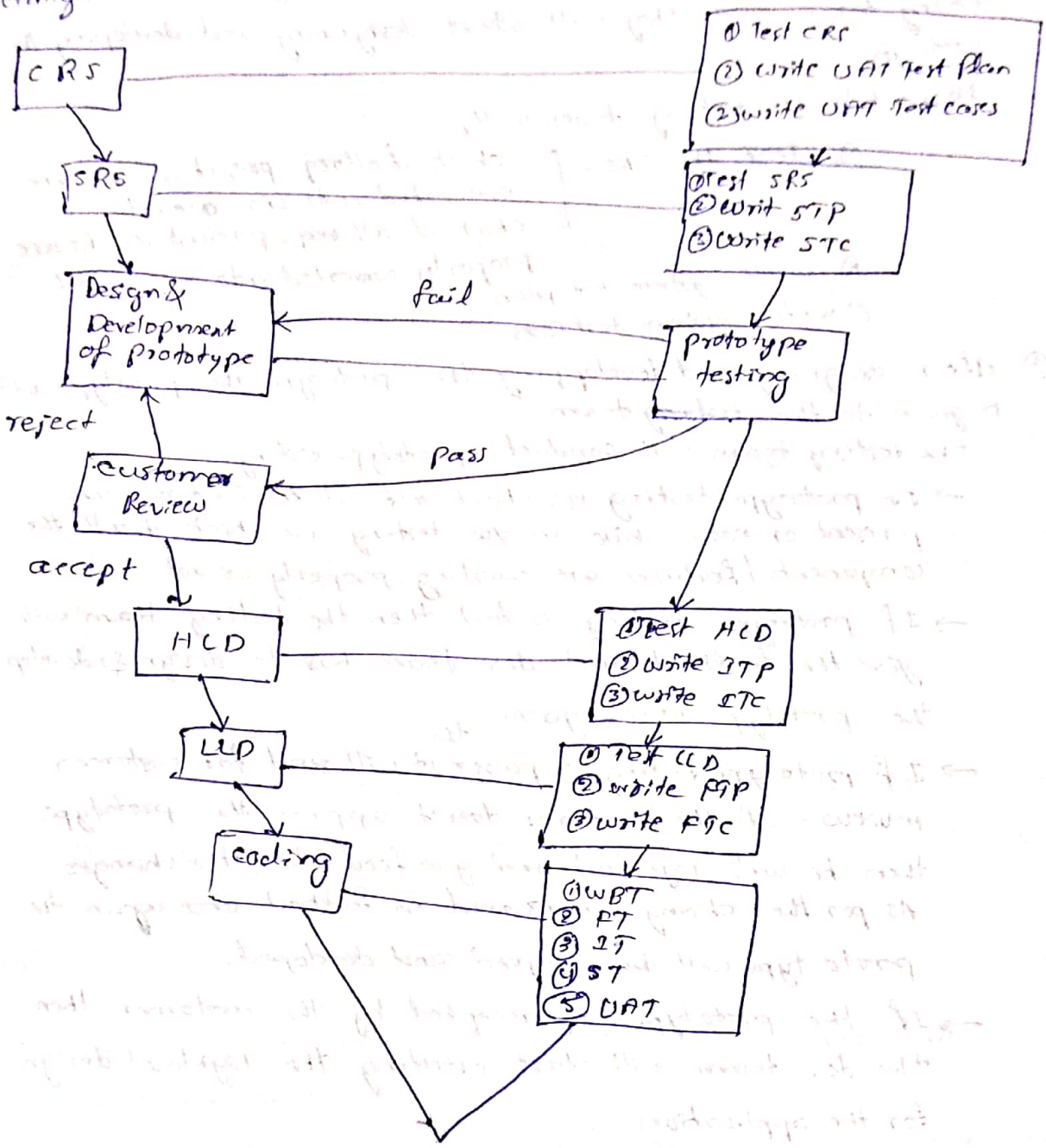
③ Implement system test plan and execute system test cases to do system testing.

→ If s/w is working properly and all bugs are fixed then we will give the s/w to the customer.

Scanned by CamScanner

27/08/2017

Prototype - V Hybrid Model



we use prototype and V model when ever the customer is not sure about the requirements and he wants a very high quality product.

The flow of prototype-v Hybrid model is as follows.

① The CRS is given to both the testing team and the dev. team the dev. team will take the CRS and start converting the CRS to SRS.

The testing team will,

- ① Test the CRS
  - Find missing req.
  - Find wrong req.
  - Find conflicts in req.

Scanned by CamScanner

(b) write unit test plan
(c) write unit test cases.

(2) Once the SRS is complete the dev. team will give the SRS to the testing team and they will start designing and developing the prototype.

Meanwhile the testing team will,

(a) Test the SRS — [check if all req. present in CRS are converted into SRS or not. / Check if all req. present in CRS are properly converted into SRS or not.]

(b) write system test plan
(c) write system test cases.

(3) After designing and developing the prototype the prototype will be given to the testing team.

The testing team will conduct prototype testing.

→ In prototype testing we check if all the features are present or not. while in s/w testing we check if all the components/features are working properly or not.

→ If prototype testing is fail then the testing team will give the feedback and dev. team has to design & develop the prototype once again.

→ If prototype testing is pass the it will send for customer review. If the customer doesn't approve the prototype then he will rejected and give feed about the changes. As per the changes mentioned in feedback once again the proto type will be designed and developed.

→ If the prototype is accepted by the customer then the dev. team will start creating the high level design for the application.

(4) After the HLD is ready the developers will give it to the testing team and they will start creating the LLD for the application.

Meanwhile the testing team will test,

(a) Test HLD
(b) write Integration test plan
(c) " " test cases

(5) After the LLD is ready the developerers will give it to testing team the developers will start coding.

Meanwhile testing will, ~~to~~

(a) Test LLD
(b) write Function test plan
(c) write " " cases

Scanned by CamScanner

(a) As soon as the coding is done the developers will check the logic for the code by conducting white box testing (WBT). After WBT the testing team will,

(a) Implement Functional test plan and execute functional test cases to do functional testing.

(b) Implement integration test plan and execute integration test cases to do integration testing.

(c) Implement system test plan and execute system test cases to do system testing.

(d) once the system testing is over and all the bugs present in application are fixed then the s/w will be given to the customer. the customer will implement Acceptance test plans and execute acceptance ~~testance~~ cases to do UAT.

Advantages

1. Both developers and customers are aware of the look and feel as well the functionality of the end product.
2. Both requirement and design are tested hence there is very low downward flow of defects.
3. The overal cost of using this model is low

Disadvantages

1. High initial investment
2. Increase in cost of project due to the development of prototype
3. Delay in starting the actual development because of extended req collection phase.

Applications

1. whenever the customer is new to the s/w and requires a very high quality of s/w.
2. Whenever developers are new to the domain and the project is highly complex.

Scanned by CamScanner

28/9/22

Agile Methodology

Agile methodology is a SDLC which is a combination of both the linear (waterfall model) and the iterative (spiral model).
Literally agile means ability to move fast and quickly adapt changes.
The main advantage of this model is that the s/w is always production ready.

Important Terminology in Scrum Process

① **Scrum Team**

The scrum team comprises of 7 with ±2 team members (7±2)
These members of mixture of developpers, testers, database handler, support engineers along with the product owner and a scrum master who leads the entire scrum team. All these members will work together in close colaboration for a recursive and definite interval to develop and impliment the required features.

② **Sprint**

Sprint is a predefined interval or the time frame in which the work has to be completed and make it ready for review or ready for production deployment. This time frame usually lies b/w 1 week to 1 month.

③ **Product Owner**

The product owner is the key stake holder or lead user of app. to be developed. He actually represents the customer side and has final authority.

- All the user stories that are to be implemented in a perticular sprint is decided by the product owner as per priority.

④ **Scrum Master**

He is the lead for entire scrum team and he make sure that the scrum team is progressive and productive.

⑤ **User stories**

User stories are nothing but the req. or features that have to be implemented. In a scrum we don't have CRS/SRS documents rather it is given in a single paragraph.

syntax:-
As a < user / user type >
I want to < Action/goal >
to achieve < specific purpose >

eg:-
As a admin
I want to ~~active~~ enable password lock after 3 failure attempts to achieve more security.

⑥ **Product Backlog**

It is a kind of bucket or source where all the user stories are stored and ~~it~~ it is maintained by product owner.

Scanned by CamScanner

⑦ Sprint Backlog

All the user stories which are going to be implemented during certain duration of sprint is call sprint back log.

⑧ Epic

It is a bucket or source where all the user stories which are going to be implemented in future sprints are stored.

⑨ story points

→ The story point is a quantitative measurement for complexity in implementing a user story.

→ Some user stories can be highly complex to implement while implementation, certain user stories can be very easy.
Hence to "measure/ quantify the complexity in implementation for the user stories" the story points are used. The story points are always given in terms of fibonacci series.

eg:- story points

1, 2, 3 } Low complexity

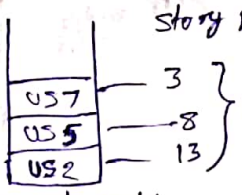
5, 8 } Medium "

13, 21 } High "

⑩ Velocity of a Sprint

The sum of all the story points for the user stories present in sprint backlog is called as velocity of sprint.

eg:-



Story points: 3, 8, 13 } 24 → velocity of sprint.

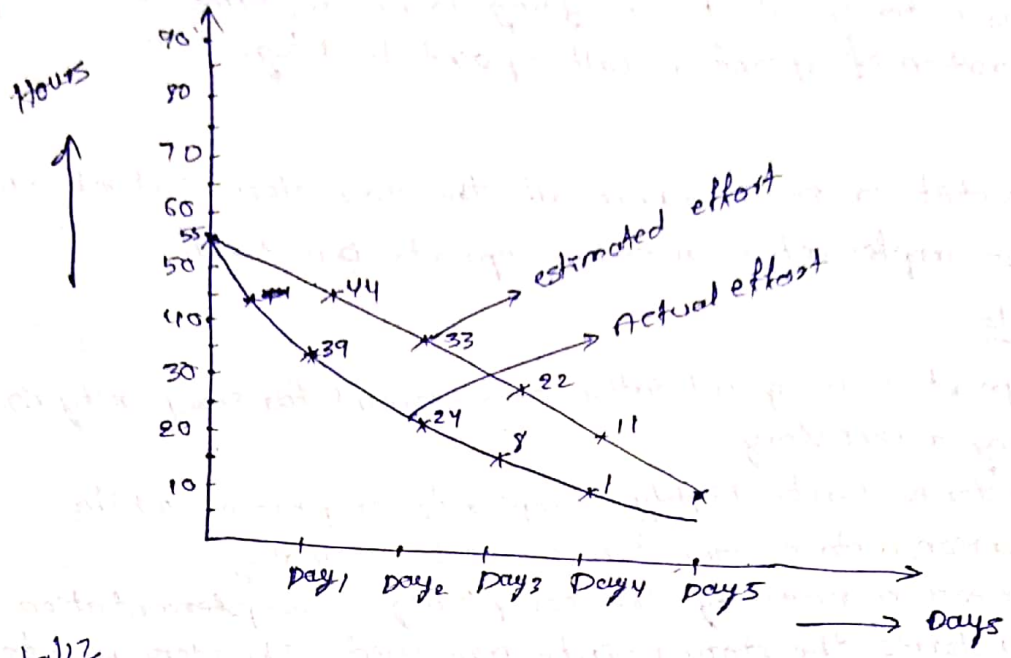
Sprint Backlog

⑪ Burn Down Chart

S.No.	User story	Tasks	Effort in hours	Day 1	Day 2	Day 3	Day 4	Day 5
1	US2	Task 1	8	5	3	0	0	0
		Task 2	7	6	4	3	1	0
		Task 3	8	4	3	3	0	0
2	US5	Task 1	12	7	3	0	0	0
		Task 2	6	5	4	1	0	0
		Task 3	5	1	0	0	0	0
3	US7	Task 1	7	7	6	1	0	0
		Task 2	4	4	1	0	0	0
Estimated Effort			55hrs	11	11	11	11	11
Actual efforts			55hrs	16	15	16	7	1

Scanned by CamScanner

Burn down chart is a chart of estimated efforts vs actual efforts

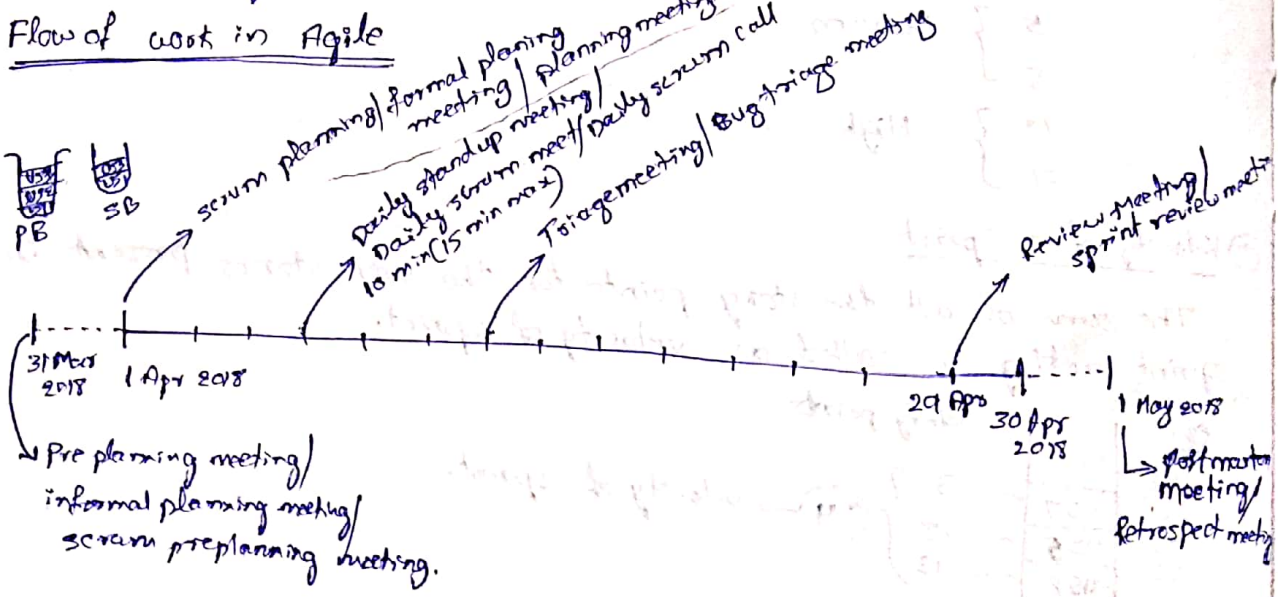


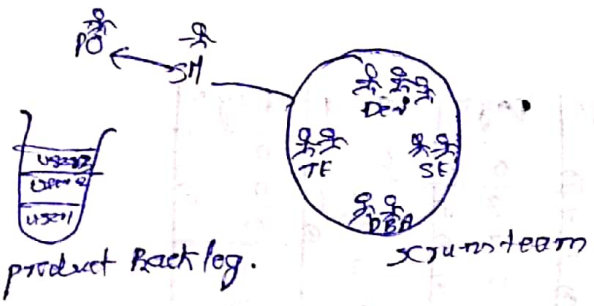
29/9/17

(12) Dom

Whenever a user story is implemented in code has passed all QA activities and it is ready to be deployed into production server then the user story is said to be 'dom'.

Flow of work in Agile





Scanned by CamScanner

① One day before the official starting of the sprint the product owner and scrum team will conduct scrum pre-planning meeting (or) informal planning meeting (or) pre-planning meeting.

→ In this meeting the product owner will select the user stories from product backlog (PB) as per priority and he will give presentation about the user stories.

→ The scrum team will,

ⓐ Decide the tasks and roles and responsibilities for all the members of the scrum team.

ⓑ All the members of scrum team as per their tasks will give the estimated no. of hrs in which then can finish the tasks.

ⓒ The scrum team will decide the story point for all the user stories.

② One the 1st day of sprint we will conduct scrum planning meeting / formal planning meeting / planning meeting. In this meeting scrum team and product owner will meet once again and following actions will takes place.

ⓐ The product owner will move the user stories which are going to be implemented from product backlog (PB) to sprint backlog (SB).

ⓑ The scrum master will enter all estimated hours in some agile management application and he will prepare a burndown chart.

ⓒ If the scrum team wishes they can change the story points for the user stories.

ⓓ the official anouncement for begining of sprint will be done and the product owner will calculate the velocity of sprint.

③ As soon as the sprint bigans all team member of scrum team will start working on tasks and every day one meeting called as daily standup meeting / daily scrum meet / daily scrum call will be done by the team.

→ In this meeting every team member will discuss 3 things.

ⓐ Work done yesterday

ⓑ work to be done today

ⓒ Any impediments / road blocks / difficulties they are facing in their work.

Scanned by CamScanner

(4) During later stages of the sprint a Triage / Bug triage meeting will be done where the priority for the bugs to be fixed will be set.

(5) One day before the official end of sprint the product owner and scrum team will conduct a meeting called as review/sprint review meeting. following actions will be taken during this meeting

(a) A team member will demonstrate the implemented user stories

(b) The product owner will conduct UAT on the implemented user stories

(c) If the product owner wants for some changes to be done, then scrum team has to implement it.

(d) If the product owner approves the user stories then the user stories will be deployed to production.

(6) One day after the end of the sprint the scrum team will meet and conduct Retrospect / postmortem meeting. In this meeting the scrum team will deside and discuss on following 3 points.

(a) Process which worked well (achievements)

(b) process which didn't go well

(c) lessons learnt

→ this meeting is done inorder to check, the processes which worked well to continue and to discontinue the processes which didn't go well and as per the lessons learnt we will plan for the next sprint.

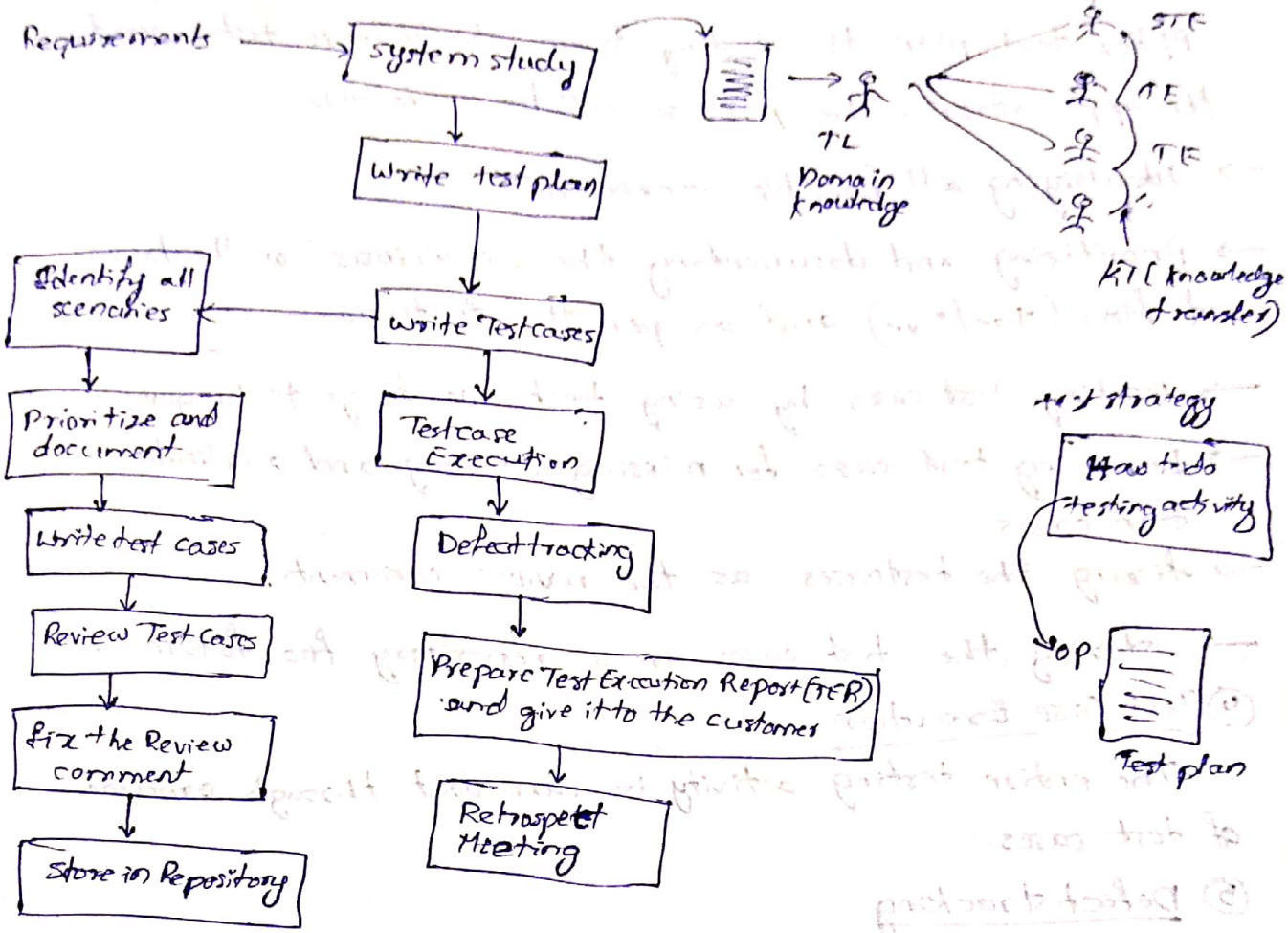
Note:

→ If any user story is not finished during the entire duration of the sprint, then these user stories carried forward to the next sprint but their story points will not be added while calculating the velocity of next sprint

Scanned by CamScanner

3/10/17

Software Testing Life Cycle (STLC)



STLC / software testing life cycle is a ~~plan~~ process to test the s/w.

It has following stages.

① The Requirements serve as the i/p for the entire testing activity. The requirements can be in the form of CRS, SRS, user stories, etc.

② System study

Here the testing team will do following things.

→ Read and understand the requirement document.

→ Provide knowledge for the T.E who are new to the domain.

→ Conduct brain-storming sessions and presentations to ensure the better understanding among the test engineers.

→ After system study testing team will conduct a meeting called test strategy in which they will decide together on the basis of their understanding about the s/w on how to test it or the processes required to carryout the testing activities in future.

The out come of this meeting is a document called as Test Plan, which will drive all the future testing activity in entire project.

Scanned by CamScanner

③ Writing test Cases

After test plan the testing team should write test cases for the application. The process will be as follows:

→ Identifying all possible scenarios.

→ Prioritizing and documenting the scenarios on the basis of flow (+ve/-ve) and as per the features.

→ writing test cases by using test case design techniques.

→ Reviewing test cases for missing, wrong and duplicate test cases.

→ fixing the testcases as the review comments.

→ storing the test cases in a repository for future use.

④ Test Case Execution

The entire testing activity is carriedout through execution of test cases.

⑤ Defect tracking

→ The defect found through test case execution will be tracked by the testing engineer. If the defects are fixed then we will conduct retesting inorder to check if it is really fixed or not. If the defects are fixed the we will close it otherwise we will reopen it.

→ For each and every cycle of test execution we will prepare a test execution report where you will mention the features which are being tested, no.of cases which are passed, no.of test cases which are failed, no.of bugs found, no.of bugs which are fixed and the no.of bugs which are not fixed.

⑥ Retrospect Meating

Here testing team will decide the following 3 things.

→ Processes which worked well

→ " didnot work well

→ The lessons during the entire cycle of testing

Scanned by CamScanner

4/10/22

Classification of Software Testing

① Based on code

Software Testing

- White ~~Black~~ Box testing (or) Glass Box testing (or) Structural testing (or) skelleton testing (or) Unit testing
- Black Box testing
- Gray Box testing.

Software testing can be classified into various types Based on the code.

→ Based on code we can classify ~~we can classify~~ testing into 3 types

① White Box Testing

Verifying the business logic of the application by testing the code is called white Box testing

② Black Box testing (BBT)

Verifying the functionalities or features of application without checking source code is called as BBT.

③ Gray Box Testing

It is a combination of white Box testing Black box testing as we are verifying both the source code and the features of the software

Diff. B/w white Box & Black Box

BBT	WBT
① carried out By testers	① carried out By developpers (or) test. engg. with good prog. knowledge.
② Implementation logic is not required.	② Implementation logic is required.
③ Programming knowledge is not required.	③ Programming knowledge is required.
④ External testing	④ Internal testing.
⑤ Based on requirement specifications document.	⑤ Based on detailed design document.

Scanned by CamScanner

② Based on Functionality

- Functional testing
  - functionality of component
  - data flow
  - end to end Bussiness flow
  - Functionality testing (or) component testing
  - Integration
  - system
  - smoke
  - Adhoc
  - Regression
  - sanity.
- Non-functional testing
  - speed of app. on applying load
  - compatibility with Browser
  - User friendyness
  - Accessibility
  - testing for diff. language & cultural standards
  - performance testing
  - compatibility
  - Usability
  - Accessibility
  - Globalization (G12N) (I18N & L10N)

Based on testing the functionalities ~~tes~~ s/w testing can be classified into 2 types.

① Functional
② Non-functional

① Functional testing (FISSAR)

→ In functional testing we test the functionality of the components, data flow b/w modules (or) end-end bussiness flow.

→ Functional testing can be done by following diff types of testing activities.

① Functionality
② Integration
③ System
④ Smoke
⑤ sanity
⑥ Adhoc
⑦ Regression

② Non-functional testing (PUACG)

Testing the non-functional features like, speed of application on applying load, compatibility with diff. browsers, user friendliness, accessibility, testing the behaviour of app. when it is used in multiple languages and checking in cultural standards is called non-functional testing.

Scanned by CamScanner

following are types of non-functional testings.

1. Performance
2. Usability
3. Accessibility
4. compatibility
5. Globalization.

5) Based on Process

- Static testing
  - Review → Req.'s, design, test cases documents
  - static Analysis → walkthrough source code
- Dynamic testing
  - Entry criteria in BRT/...
  - Exit criteria "

Based on the process of testing we have two types.

1. static testing
2. Dynamic testing

1) static

→ It is a process of reviews and static analysis. Here we will review the requirements, design, test cases and related documents while we will also do static analysis for the source code.

→ It is a process of early detection of bugs which can lead to structural defects or design flaws.

2) Dynamic

It is a process where the s/w is tested for a set of input data and we verify the output results

Exam Questions

1. write 20 -ve scenarios from fresh menu application
2. Explain the flow of work in agile
3. Explain prototype & V-Hybrid model
4. Explain the classification of software testing based on various factors.
5. Explain the spiral prototype hybrid model.

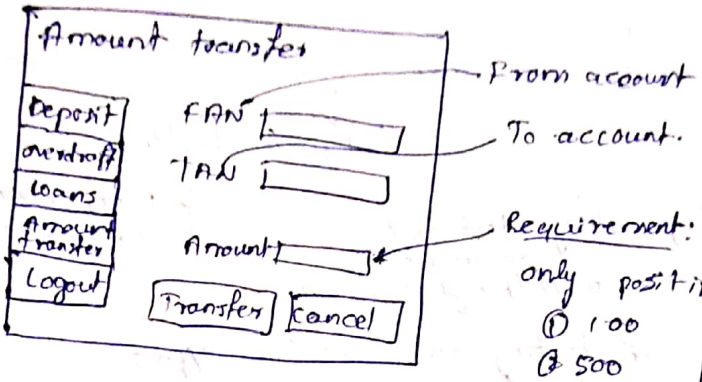
-ve scenarios
Fresh menu

Scanned by CamScanner

Black Box Testing

① Functional Testing

Testing the functionality of any component thoroughly against the requirements is called functional testing. Also called as component testing since we are checking the throughly the functionality of some specific components.



Requirement: Amount should accept only positive integer zero-20000

① 100
② 500
③ -100
④ Hundred
⑤ 100$
⑥ Rs 100
⑦ 100.00
⑧ 20000
⑨ 1□0□0
⑩ Empty
⑪ □□□□ ← spaces

To test any component the test engineer can adopt following diff. ways.

① Over testing:

Testing the component for more than the required no. of scenarios (or) values is called as over testing. The disadvantage with over testing is that it consumes a lot of time and to test other features very less amount of time is left.

② Under Testing:

Testing the application or a component with less than the exact no. of scenarios. The disadvantage with under testing is we might miss a lot of bugs during testing.

③ Optimized Testing

Testing the application or component with the exact no. of required scenarios is called as optimized testing.

How to do optimized testing?

Using test case design techniques we will find the exact no. of values required to do testing.

Test Case Design Techniques

① Boundary Value Analysis (BVA)

→ We use BVA when ever the input is a range of values

→ As per BVA if there is a range from A to B then we will test the component for A, A+1, A-1 and
B, B+1, B-1

Scanned by CamScanner

for eg. in the amount text field is in range from 100 – 20000 then the values for which we should do testing

(A)	(B)
100	20000
100+1 = 101	20001 (+1)
100-1 = 99	19999 (-1)

A	B
A+1	B+1
A-1	B-1

→ As per ISTQB (don't tell in Interview)

	A	B-1
	A+1	B
eg:-	100	19999
	101	20000

100, 101, 19999, 20000 } valid
99, 20001 } Invalid

Scenarios

1. verify that user should be able to transfer Rs.100 to the valid user
2. " " " Rs 101 "
3. " " " Rs 19999 "
4. " " " Rs 20000 "
5. " should not be able to " Rs 99 "
6. " " " Rs 20001 "

How will you update the scenarios for change Requests (CR)?

Requirement: Amount text field should accept only positive integers between 1000 to 20000 in multiple 100

1000, 1100, 19900, 20000 } valid
900, 20100 } Invalid

Scenarios

1. verify that user should be able to transfer Rs 1000 to the valid user
2. " " Rs 1100 "
3. " " Rs 19900 "
4. " " Rs 20000 "
5. " should not " Rs 900 "
6. " " " Rs 20100 "

(2) Error Guessing

Using this technique we guess all posible errors that can be done.

(a) –200
(b) 200.50
(c) Two hundred
(d) 200$
(e) Rs.200
(f) # ! ? -
(g) □□□□
(h) empty
(i) 1□0□0

Scenarios

1. Verify that user shouldn't be able to ~~transfer~~ enter a -ve integer in the amount text field.
2. " " Decimals "
3. " " Alphabets "
4. " " combination of nos and special characters "
5. " " AlphaNumerics "
6. " " Special characters "
7. " " Blank spaces "
8. " " leaving empty "
9. " " combination of integers & blank spaces "

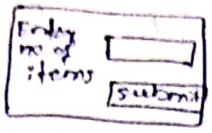
Scanned by CamScanner

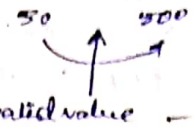
9/10/

③ Equivalance Class Partition

① Pressman's Method

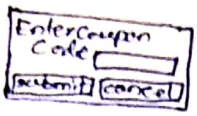
ⓐ Range





1. valid value — 105
2. Invalid values — 40 / 625

ⓑ Set



1 DMFP036
1DMFP037 ✓ valid.e
1DMFP529 — invalid.e

ⓒ Boolean

True — ⊙ Radio Button
False — ☑ check Box
} click/unclick. | 1 – click/check – true, 0 – uncheck – false.

① Range

→ If the input is a range of values between a to b then we will test the component with one valid value with in the range and two invalid values outside the range.

② Set

If the i/p is a set of items then we will select one valid value from the set and two invalid values outside the set.

③ Boolean

If any component has i/p like a boolean value (like, radiobuttons, check boxes, etc) then we will test the component for both true and false values.

② Practice Method

Range 5 lacks – 50 lacks
deviation
- 5 lack – 10 lack – 14%
- 10 lack – 20 lack – 12%
- 20 lack – 50 lack – 10%

R1:
① 4,99,999 – Invali
② 500,000
③ 9,99,999
④ 10,00,000
⑤ 10,00,001
R2:
⑥ 19,99,999
⑦ 20,00,000
⑧ 20,00,001
R3:
⑨ 49,99,999
⑩ 50,00,000
⑪ 50,00,001

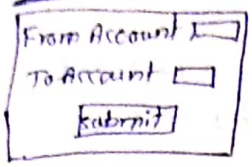
→ we use practice method when-ever the i/p is a range of value and there is a deviation with in the range

→ In this method we will find diff. sub ranges present with in the range and we will apply boundary value analysis (BVA) for them.

Scanned by CamScanner

Scenarios

Writing Scenarios



Range 100 - 20000

① Verify that user should be able to transfer 100 to any user
② Verify that user should be able to transfer 500 to any user
③ Verify that user shouldn't be able to transfer -100 (Negative Integer) to any user
④ " " " Hundred (Alphabets) to any user.
⑤ " " " 100$ (Numbers & special char) "
⑥ " " " Rs.100 (Alpha numerics) "
⑦ " should " 100.50 (Decimals) "
⑧ " shouldn't " 20000 (Numbers) "
⑨ " shouldn't " 1□0□0 (Numbers & Spaces) "
⑩ " shouldn't " Empty "
⑪ " " " Blank Spaces "

→ Range 1000 - 20000 (multiple of 100's)

① Verify that customer should be allowed to transfer Rs.1000 from his account to another account.
② " 1100 "
③ " 19900 "
④ " 20000 "
⑤ " 15000 "
⑥ shouldn't 900 "
⑦ " -900 "
⑧ " Rs.1000$ "
⑨ " Thousand Rupees "
⑩ " 1□0□0□0 "
⑪ " Empty "
⑫ " □□□□ (spaces) "
⑬ " $%!& "
⑭ " 1000.25 "
⑮ " 100 "
⑯ " 30000 ✓ "

BVA

1000 - 20000

1000	20000
900	19900
1100	20100

Error Guessing

Rs.100$ (Nos & special char)
Thousand Rupees (Alphabets)
1□0□0□0 (Number & spaces)
Empty (Bank)
□□□□ (spaces)
$%!& (special characters)
1000.50 (Decimals)

Pressman's method

100 (-ve)
15000 ✓ (middle)
30000 (+ve)

Paytm Amount transfer

1 - 5000 range multiple of 100

BVA

① 1 (min)
② 101 (min+100)
③ -100 (min-100)
④ 5000 (max)
⑤ 5100 (max+100)

Error Guessing (-ve scenarios)

⑥ Rs.100
⑦ Hundred
⑧ 1□0□0
⑨ □□□□
⑩ $%!-&
⑪ 100.50
⑫ 100$

Pressman's method

⑬ 3000 ✗
⑭ 6000 ✓
⑮ -200 ✗

Scanned by CamScanner

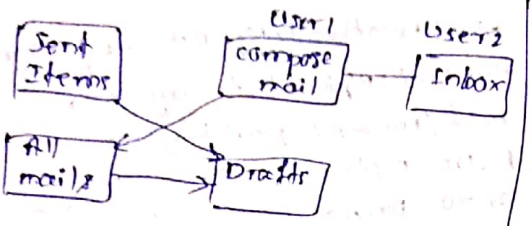
Integration Testing

Integration testing

→ Data flow between two modules or interfaces should be tested whenever two modules or interfaces are integrated with each other.

→ Testing this data flow between two modules or interfaces is called as integration testing.

eg:-



Integration between two modules

1. login as a valid user1
2. Send a mail to user2
3. Logout
4. Login as user2 [mail sent should be present]
5. Logout
6. Login as user1
7. check in [mail should present]
8. Check in all mails [mail should present]

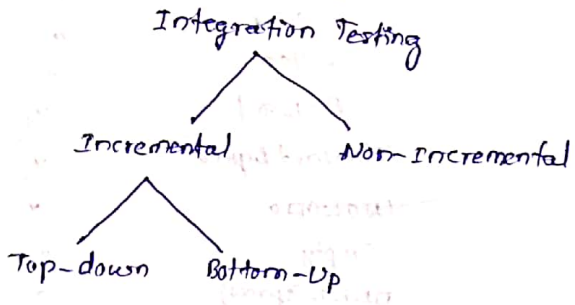
Integration between two Interfaces (Data flow b/w two applications)

→ WhatsApp & phone contacts. – If we add any phone contact it reflects in whatsApp contacts.

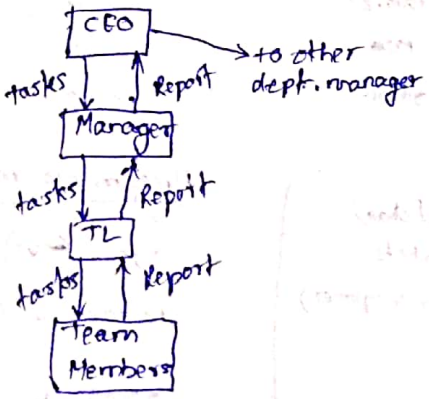
→ Paytm contacts/users should present in contacts.

→ Payment through netbanking

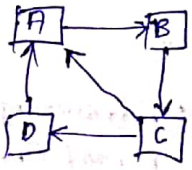
types



Incremental Integration



Non-Incremental Incremental



Incremental integration testing

Testing the data flow between parent and child modules.

Types

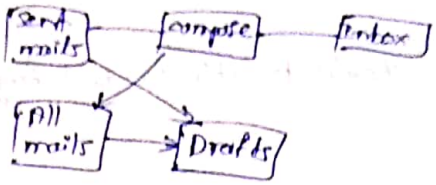
1. **Top-Down**: Testing data flow between parent and subsequent child modules.
2. **Bottom-Top**: Testing data flow from child modules to subsequent parent modules.

Scanned by CamScanner

Non-Incremental Integration Testing

Testing the data flow between parent and child modules is not possible due to complex integration between the modules.

eg:- Gmail



System Testing

→ Testing the end to end bussiness of the application where the test environment is similar to the production environment.

→ Test environment is similar to production environment means,

① The hartware should be similar
② The software "
③ The test data "

Why we do system testing ?

→ In system testing we check the end to end bussiness process of the application in a similar environment to production environment as application has a bussiness to perform (or) certain work to fullfill because of which only people will use it. Hence testing these end to end bussiness process is very important from the business point of view of application.

When we do system testing ?

→ System testing should be done at the end when all the features are ready and they are working properly.

→ Generally the application should be pass in both functional and integration testing then only we can start with system testing.

How to perform System testing ?

① Identify all the business process of the application
② Identify the end to end process
③ Create end to end bussiness scenarios
④ Convert the scenarios into test cases
⑤ Execute the test cases to perform system testing

Note

→ Functional / Integration / System testing done through Test cases.

eg:

writing scenarios for makemytrip

Low level scenarios

① Verify that any new customer should be able to book flights
② Verify that any new customer through referal code should be able to book flights.
③ Verify that existing customer should be able to book flights after login
④ Verify that any new customer should be able to book flight as guest user.

Scanned by CamScanner

(5) verify that any new customer can create account while booking flight.

(6) verify that any existing customer should be able to book flights and login while making payments.

High level scenarios

(1) verify that any new customer should be able to book flights
(2) Install the application
(3) Create account
(4) Login
(5) Search flights
(6) Make payments
(7) Receive e-ticket in email

---

exam Questions

(1) what/why/when/How to do functional testing
(2) " integration " (∵ After functional testing we do integration testing)
(3) " system "

16/1/17

Smoke Testing (+ve)

"Testing the basic and critical features of the applications as soon as the build is received is called as smoke testing."

→ Smoke testing is a positive testing (testing with valid values only) which is done to ensure that the features which are very very important for bussiness of application are working properly or not.

→ This is done to check the testability of the build.

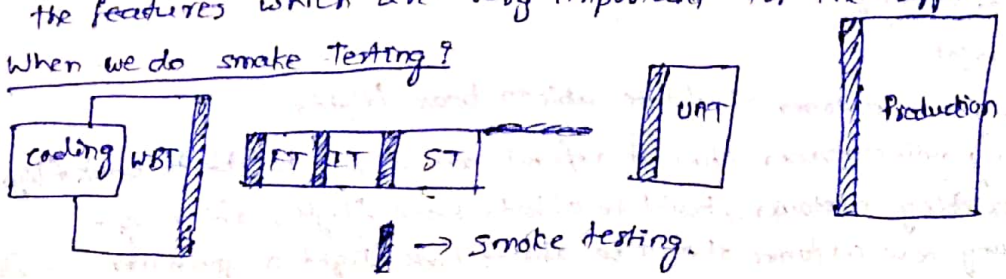
Why we conduct smoke testing?

→ To ensure that the build is testable
→ To find blocker (or) critical bugs in early stage.
→ To ensure that the build is properly installed.

When the build can be Rejected During Smoke testing By the testing team?

→ When build is unstable
→ when all the basic features are not working properly or some of the basic features are missing.
→ when there are lot of blocker bugs and critical bugs present in the features which are very important for the application

When we do smoke Testing?



Scanned by CamScanner

→ As soon as the build is received we do smoke testing.
→ Before starting functional, Integration and System testing, the testing team should conduct smoke testing.
→ Before starting UAT the customer should conduct smoke testing.
→ As soon as the build is installed into production the ~~customer~~ s installation team should conduct smoke testing.
→ After white Box testing (WBT) before giving build to the testing team the development team should conduct smoke testing.

Type of smoke Testing

① Formal smoke testing:

When the employees are supposed to do smoke testing and asked to do so then it is called formal smoke testing.

eg: → Testing done by testing team as soon as build is received.
→ testing done by testing team before starting functional, integration and system testing.
→ The customer's IT division performing smoke testing before UAT.

② Informal smoke testing:

The employees performing smoke testing when they are not asked or not supposed to do smoke testing then it is called as informal smoke testing.

eg: → Developpers performing smoke testing after white Box testing.
→ The installation team performing smoke testing after installing s/w into the production server.

How we do smoke testing?

→ For smoke testing we don't write separate test cases but we select the test cases which are for testing the basic features of the application and those test cases are positive for the purpose of smoke testing.

→ If smoke testing is done through automation then the same test cases should be converted into test scripts.

Scanned by CamScanner

Regression Testing

"Testing the unchanged or old features in the application due to changes is called as regression testing."

→ changes means,
1. Addition of new features
2. Modification of existing features.
3. fixing of bugs
4. Deletion of old feature.

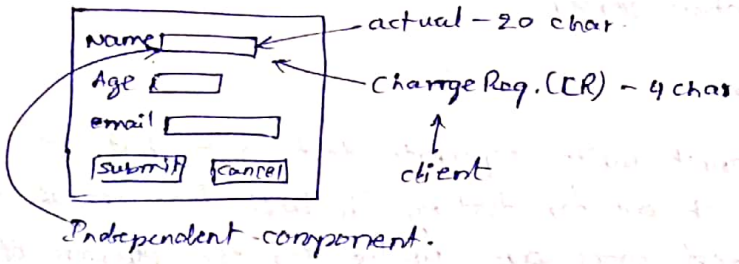
→ When ever development team gives a build then chances are there that they've implemented some of the changes in app. since these changes involve changes in code. Hence, chances are there that unchanged features might be effected. Hence we need to do testing for unchanged features as a part of Regression testing.

Types

① Unit Regression Testing

→ "Testing only the modified features is called as unit Regression testing."

→ When even some modifications are done for some independent component which has not related or dependent features then it is called as "Unit Regression testing."



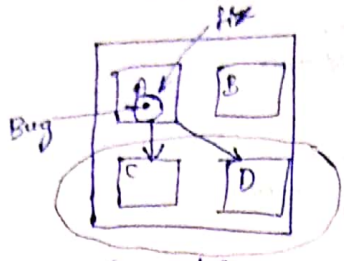
→ for the above feature the name text field takes upto 20 char but if we received a change Request (CR) from customer to increase it to 40 char then as this feature is an independent component Hence we will test only this modified feature as per CR to perform unit Regression testing.

② Regional Regression Testing

→ "Testing the impact areas due to the changes is called as Regional Regression testing."

→ When ever there are some changes done to one feature then chances are there the Related or dependent features might be effected. Hence we need to test these unchanged or dependent or

Related features to perform regional regression testing.



Regional Regression testing.

→ In the above eg. bug present in module A and module A is related with module C and module D. Hence we are testing module C and D is called as regional regression testing as these two features are impact areas.

③ Full Regression Testing

→ "Testing all unchanged features of the application is called as full regression testing"

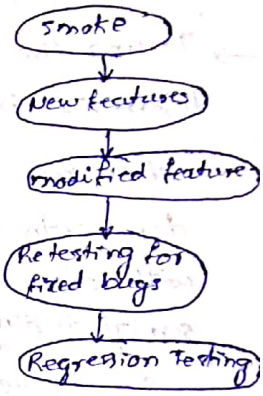
→ We do full regression testing when ever,

① Lot of code changes are done (lot of change Requests also)

② When bug is present in internal design or root of the application.

→ In both above cercumstances every feature may get effected Hence we go for full regression testing.

When we do Regression? →



How do we perform Regression testing?

→ Regression testing can be done in two ways.

Manual – By executing Test cases

Automation – ① Convert all test cases to test scripts

② Execute the test scripts to perform Regression testing

Diff. b/w Retesting & Regression Testing

→ If there is a bug present in a module and dev. team fixed. Hence testing the same module again to check whether it is fixed or not then we do Retesting.

→ After fixing the bug, ~~testing~~ testing the Related modules or features or functionality due to the fixing bugs then we do Regression testing.

Exam Questions

① Write functional testing ~~feat~~ scenarios for log in feature. (un, pw, submit, cancel, forgot pw.)

② Explain the types of integration testing with examples

③ Explain smoke testing interms of what/why/when/How.

④ Explain Regression testing and types of Regression testing.

① Functional Testing scenarios for Login

→ Verify that the login screen is having option to enter user name and password with submit button and option for forgot password.

→ Verify that user is able to login with valid username and password.

→ " not " " invalid " "

→ Verify that validation message is displayed in case user leaves username or pwd field as blank.

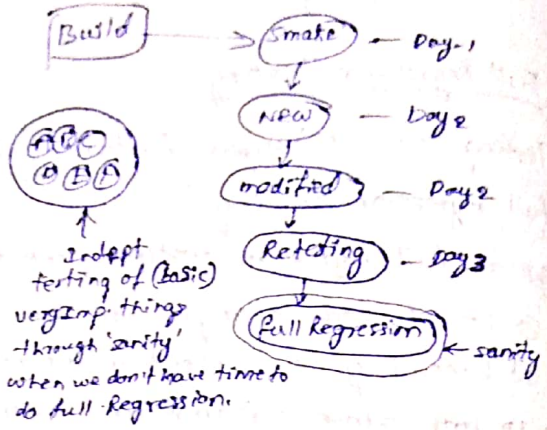
→ " " user exceeds the character limit of uname and pwd.

Scanned by CamScanner

→ verify that there is a reset button to clear the text fields.
→ verify that there is checkbox with label "remember password" in login page.
→ verify that password is in encrypted form when entered
→ verify that there is a limit on total no. of unsuccessfull attempts.
→ verify the timeout of the login session.
→ verify that if the password field can be copy-pasted or not
→ verify that once logged in, clicking back button doesn't logout user.
→ verify if SQL injection attacks works on login page
→ verify if XSS vulnerability work on login page.

23/10/2017

Sanity Testing



→ "Testing the basic features of the application in dept done without testcases is called as sanity testing."

→ When ever we receive a build and there are lot of changes done into it and to test the unchanged features we are left with very little amount of time then in such cases we will do sanity testing.

→ sanity testing is done to make sure that the basic features atleast are not effected due to changes when to test the unchanged features we are left with very less amount of time in the testing cycle.

characteristics of sanity testing

→ Testing is done without test cases

→ It is done inplace of full regression when ever there is lack of time in the testing cycle.

→ Only the basic features are tested indepth hence it is narrow and deep testing.

Diff. b/w smoke and sanity testing

Smoke	Sanity
① High level testing - for working or testability of the build	① Indepth testing - testing basic features indepth.
② with valid values	② with valid values & Invalid values.
③ Done with test cases	③ without testcases
④ Can be automated	④ Can't be automated.

Scanned by CamScanner

Exploratory Testing

Exploring the application, using the application and checking it with some random values to understand its behaviour and working is called as exploratory testing.

Why we do exploratory testing?

Exploratory testing is done to understand and findout the working of software whenever any application is going to be tested. the test engineers must conduct exploratory testing to be well aquented with all features of the software.

When we do exploratory testing?

→ When ever a new testing engineer has joined in team

→ Whenever we are going to start testing a new software.

→ Whenever requirement documents are not present inorder to understand the behaviour and find scenarios.

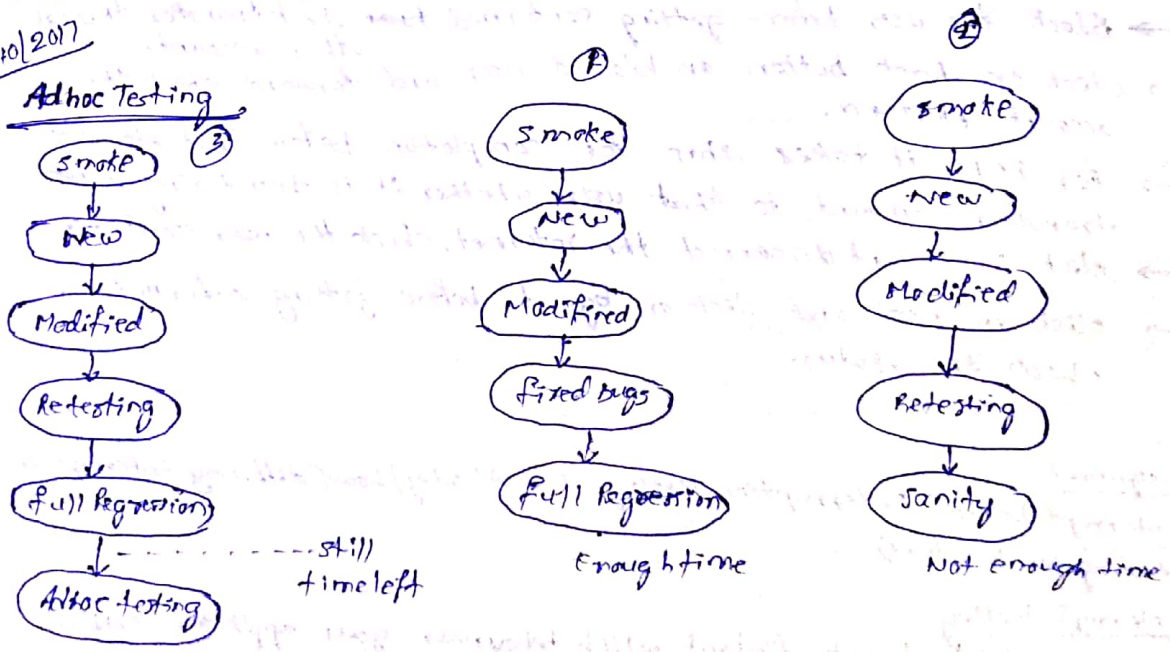
How do we conduct exploratory testing?

→ check for the end to end to end bussiness features of the application by exploring it.

→ It is done without test cases.

24/10/2017

Adhoc Testing



Adhoc / Monkey / Gorilla Testing :-

Testing the application randomly is called adhoc testing.

Characteristics

→ It is a -ve testing which is done to break the application in any way possible.

Scanned by CamScanner

→ Since the application is tested with random data hence no test cases are required.

→ In adhoc testing we test the software or application out of the requirements.

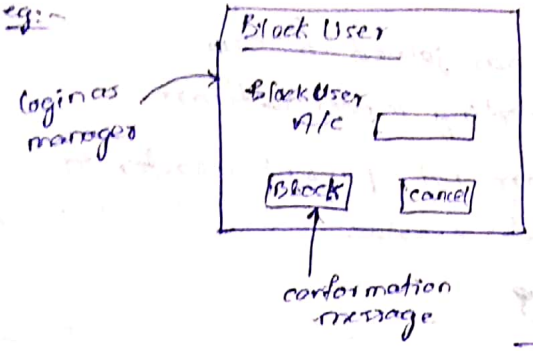
Why do we need to conduct Adhoc testing

→ Because the end users can use the application randomly.

→ lot of bugs and security issues can be mised if the application is tested systematically only through test cases.

→ To increase the bug count.

eg:-



→ login as manger block the user conformation ~~log~~ message should display.

→ login as blocked user check whether user exist or not.

→ login as other user send money to the blocked user amount is cannot transfer to this user should display.

→ Block the already blocked user

→ Deleting blocked user

→ send money to the deleted user shoudn't allowed to send.

→ Block the user before getting conformed ~~tran~~ check transfer through other account.

→ click on back button on blocked user and forward check the account existance.

→ For IMPS it takes 2hrs for completion before that block it transfer amount to that user whether it is transfering or not

→ click block and disconnect the internet, check the user existance.

→ click on block and click on logout before getting conformation, check the status.

Assignment

→ Interrupt based / Interruption testing what / why / how (diff. way interruption testing can be done).

Interrupt testing

→ Interrupt testing tries to findout which behaviour your application exhits.

→ Its a branch of mobile application testing that deals with – How an application reacts to an interruption and resumes to its previous state.

→ Interrupt testing applies to any application type – web, mobile, standalone, etc. The variety of devices, networks, configurations, etc. makes it more prominent for mobile applications than the others.

→ In interrupt testing the user must be sure that the application is working properly after interrupts are made and how the application responds to the interrupts

1. Incoming and outgoing SMS and MMS
2. Incoming and outgoing calls.
3. Battery Removal
4. Data cable insertion and removal
5. Network provider outage and recovery.
6. Device power cycle (low Battery)

How to do interrupt testing?

Interrupt testing is a subset of functional testing for mobile application and to conduct interrupt testing you would follow the same mobile application test frameworks and tools. It is the skill of testers to conceive these scenarios. once done, you would design the test cases and execute in exact same way as any other test.

25/10/2017

Tracebility Matrix

It is a document which shows the mapping between both the requirements and test cases associated with it.

Why do we need Tracebility Matrix?

→ Tracebility matrix matrix make sure that each and every requirement has atleast one test case associated with.

→ While creating test cases the test engineer might miss some of the requirements hence to make sure that each and every requirement has a test case we need to have a document which maps the test cases with its requirements and if there is any requirement which doesn't have any test case associated with it then we should be able to find it out through this document.

eg

S.No	Req. No.	Req. Description	testcase Name	Automation Script Name
1	1.1	create User functionality	city_CreateUser_TC_01 City_CreateUser_TC_02 City_CreateUser_TC_03 City_CreateUser_TC_04	CreateUser_TS_01.java CreateUser_TS_02.java CreateUser_TS_03.java CreateUser_TS_04.java
2	1.2	Login functionality	city_login_TC_05 City_Login_TC_06 City_Login_TC_07	Login_TS05.java Login_TS06.java Login_TS07.java
⋮	⋮	⋮		
56	31.1	Amount Transfer		

Tracebility Matrix.

Scanned by CamScanner

Pesticide Paradox

Pareto's Principle / 80 – 20 rule

→ 80% of bugs will come from 20% of module → Hotspot

→ when ever we are testing through testcases and the test cases are repeatedly used then the ability of our test cases to find bug reduces. This can happen because the bugs are already fixed or the developpers understand the pattern in which the application is tested and try to hide the bugs

Pareto's principle / 80-20 rule :-

→ As per this principle 80% of bugs are created by 20% of modules. This leads to "defect clustering" where the maximum issues are created by few modules. These modules may also be known as Hotspots.

→ The test designers often use these when they are creating assessments during test planning and the test cases related to these areas will be continuously revised and updated to find potentially more bugs.

26/10/17

Test Cases

→ Test case is a document which contains and covers all possible scenarios required to test all the features.

→ If test cases are not written then the testing quality depends on,

1. Mood of test engineer
2. It is not consistant
3. It varries from person to person

Why do we need to write test cases

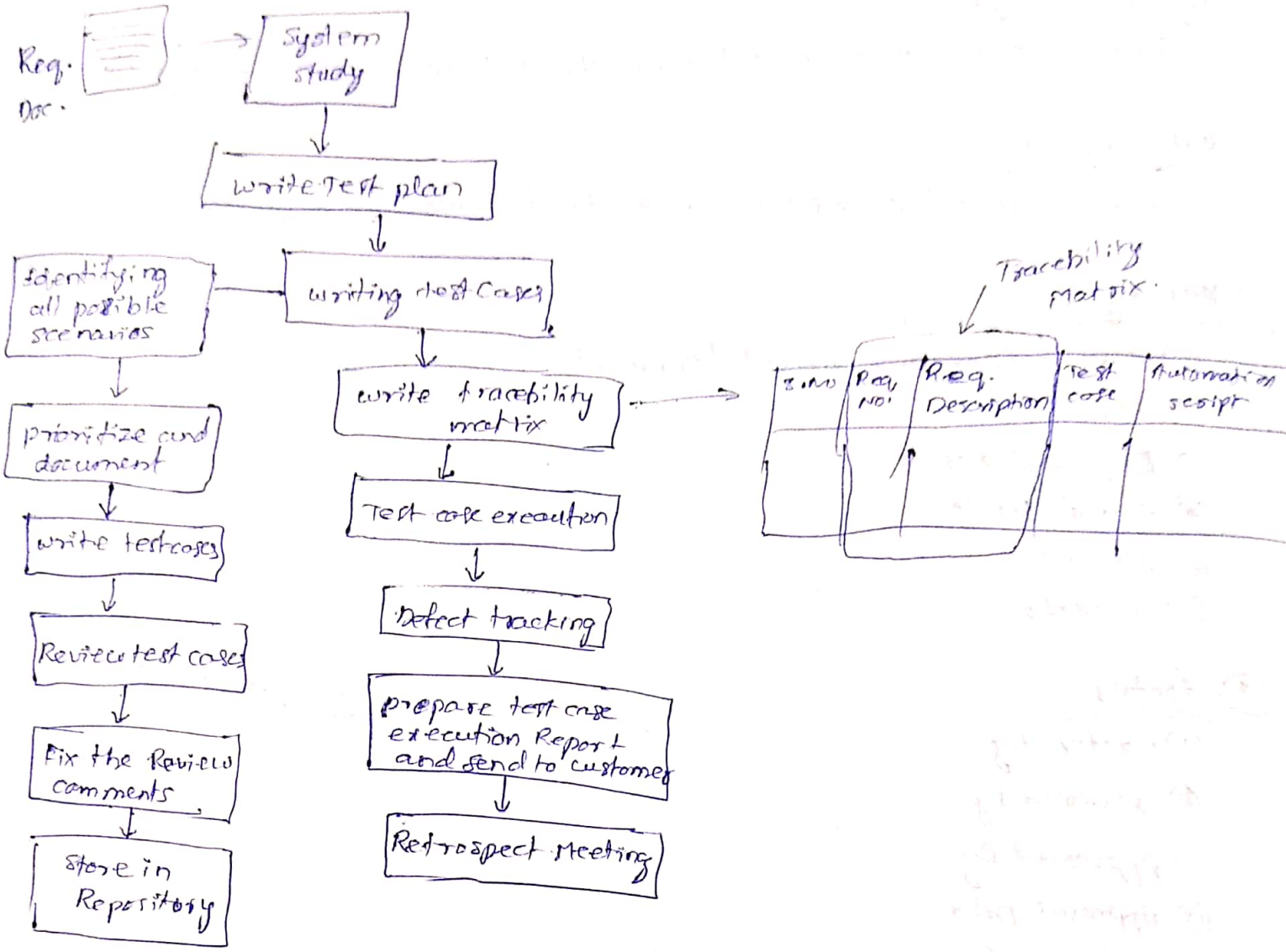
→ To have a better coverage

→ To have consistency in test execution

→ To avoid training every new test engineer who joins the team.

→ To depend on a process rather than a person.

Scanned by CamScanner



Scanned by CamScanner

(f) Test case type

Here we will mention whether the test case is positive or negative

(g) Description

Here a small description about the test case will be given

(2) Body

(a) Steps to Reproduce/Execute/Represent
(b) Input
(c) Expected Result
(d) Actual Result
(e) Status
(f) Comments

(3) footer

(a) written By
(b) Reviewed By
(c) Approved By
(d) Approved Date.

ana.co.in → ZOTC

30/10/17

Usability Testing

MMT	ERCTC
→ No need to login can use as guest user	→ cannot
→ can login through google	→ cannot
→ Multiple languages	→ less
→ session timeout is more	→ less
→ No logout	→ Clicking back button makes logout
→ Navigation	→ Poor
→ Help	
→ load	
→ location of components	

Testing the user friendlyness of the application is called as usability testing. While conducting usability testing we must look for following parameters.

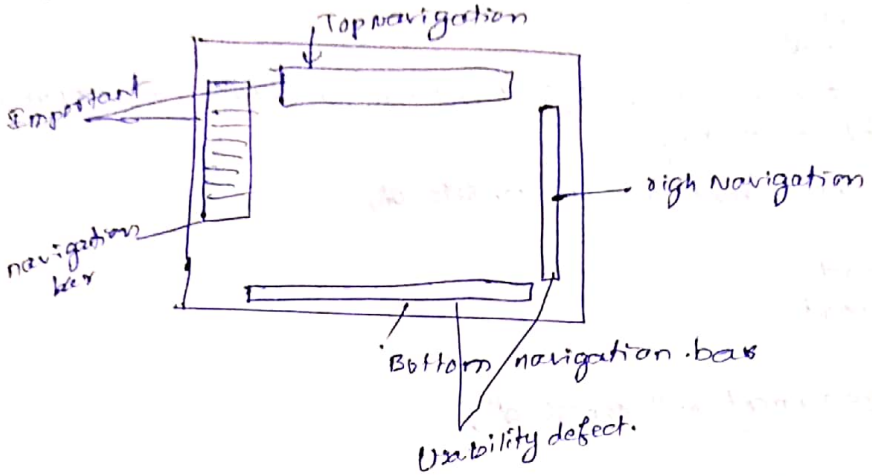
1. Ease of learning the application
2. Time Required to learn application
3. look & feel of the application

Scanned by CamScanner

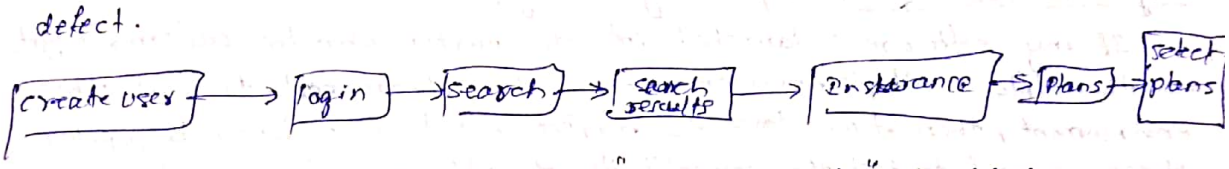
④ Speed Navigation flow
⑤ easy to use
⑥ Location of components.

concepts of usability testing

① All the important components should be present in the top navigation bar. If these components are present in the bottom ~~nag~~ navigation bar or Right navigation bar then it is usability defect.



② Any activity which takes more than 3 mouse clicks it is usability defect.

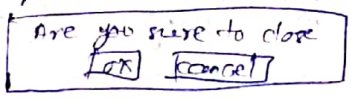


③ while navigation to the child pages "Bread crumbs" should be provided.

Home | create | login | search | Insurance | plans

④ For any web page the background should always be in the light colour while the text should be in dark colour.

⑤ The cancel button should always present in top right corner and ~~for~~ whenever cancel button is pressed confirmation pop-up should be displayed.



⑥ location of confirmation popups.

web	mobile
OK cancel	cancel OK

⑦ Undo and Redo buttons should be present. whenever some changes are done.

⑧ The remainder for saving must be given if the changes are done and without saving the changes the user is closing the application or using some other operation.

Scanned by CamScanner

Is usability testing is really Important?

Every application is created for bussiness in an open market if the application is not user friendly then it will not be popular and hence it will loose the bussiness. This is the reason which makes usability testing is very important for testing.

31/10/17

Compatibility testing

Testing the software in multiple hardware & software environments in called as compatibility testing.

The hardware environment will be consists of,

1. RAM
2. Motherboard
3. Graphics card
4. Hard disc

The software environment will consists of,

1. Browsers
2. Operating systems
3. Data bases.

Why do we need to do compatibility testing?

If any software is launched into the market then the customer might use them into multiple environments. If the software fails to work those environment, then it will loose its market and it will no longer be popular. Hence we need to test the compatibility of software with multiple environments.

There are two types of issues,

1. compatibility Issue
2. functional Issue

① **Compatibility issue**

If any feature is not working only in one environment particularly then it is called as compatibility issue.

Bug — It is present only in mozilla firefox

Bug — all environment

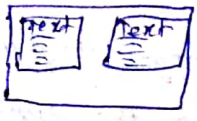
② **Functionality Issue**

If any feature is not working in all environment then it is called as functionality issue.

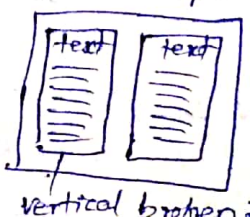
Few types of compatibility bugs:-

1. change in look and feel of the application.
2. change in font, color, text, etc.
3. Broken frames.

Mozilla firefox



Internet Explorer

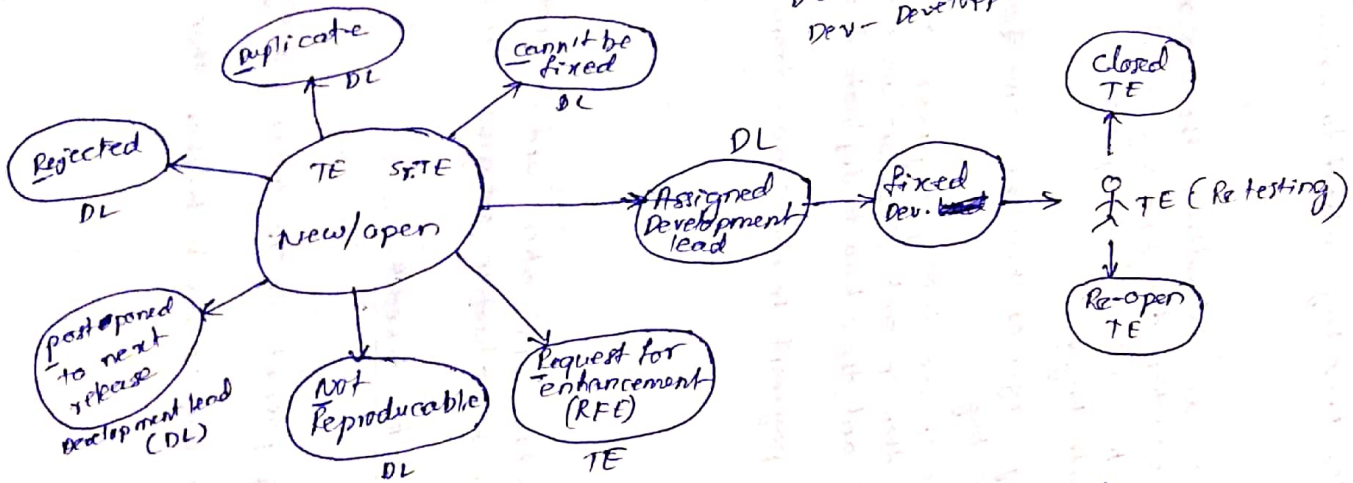


vertical broken frames

Scanned by CamScanner

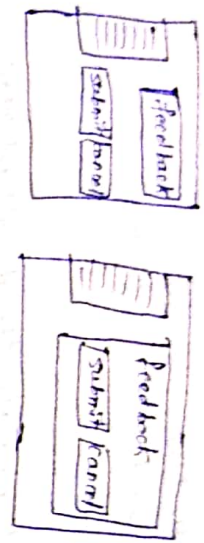
(R-D-C)-ADL-D-T KT
P-N-R
RO

TE - Test Engineer
Sr.TE - Senior Test Engineer
DL - Development Lead
Dev - Developper.

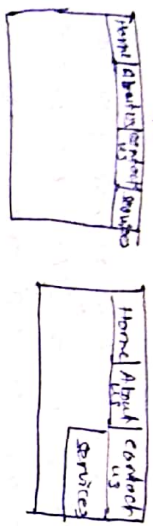


Bug life Cycle / Defect life Cycle

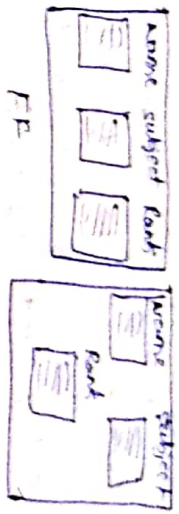
① Objects overlapping:



② Alignment Issues



③ Scattered Content



→ As soon as a testing engineer finds a bug he logs the bug in defect tracking tool and status of the bug in a bug report will be new.

→ The bug reports will be validated and checked by the senior testing engineer from the testing team. This process is called as bug validation. If it is passed in bug validation then the status of the bug will be changed to open otherwise it will be changed closed.

→ If this report has the status open, the development lead will assign a developper for fixing this bug ~~and he fixit~~ and he will change the status of the bug to assign.

→ After fixing the bug, the developper will change the status of the bug to fixed.

→ once the status of the bug is fixed then testing engineer has to perform retesting. After retesting if the bug is passed then the status of the bug will be changed closed. Otherwise it will be changed to reopen and it will be again assigned by the development lead to same developper for fixing it. And the cycle will continue, untill the bug is fixed.

Apart from these, there are several other status related.

Scanned by CamScanner

① Rejected

When the development team doesn't consider a bug to be valid one, then they will change the status of the bugs, to rejected.

When a bug can be rejected or what are the situations. Under which a bug can be rejected?

→ Due to misunderstanding of the requirements.

→ If the software is not installed properly, the it will not work properly. In such situation the bug that we may found may be rejected.

→ **Refering to old requirements:-** If there is a change request given to the development team but the same change request (CR) is not updated to this testing team then in such situation the test cases of the testing team will be refering to old requirement and the bug found with others test cases will be rejected.

→ **Due to extra features:-** If the development team has created some extra features for the improvement in functionality or usability of the application, then as these features are out of requirement. Hence the testing team may find these as bugs, these bugs related to extra features will be rejected.

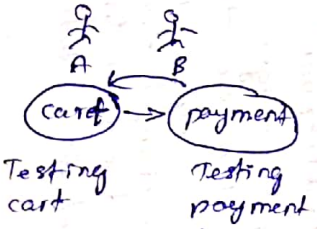
② Duplicate

If two testing engineers or more are submitting the bug report about the same bug then in such situation the first bug report will be accepted and rest all of them will be considered as duplicate.

In what situation we can get duplicate bug?

→ **Common features:-** If two testing engineer have the same feature to test then they might find some bug.

→ **Dependant features:-** If feature of one testing engineer are dependent on some other feature then in such situation we might get some bug on dependent feature and these lead to duplicate feature.



To test payment he should move from cart only.

③ Cannot be fixed

If there is a bug present in the application and it is not possible to fix the bug then, in such situation we give the status cannot be fixed.

In what situations a bug cannot be fixed?

→ Technology is not present

→ Whenever there is a minor bug present in the internal design of the application.

→ When ever the cost of fixing a bug is much higher than cost of ~~fixing a~~ the bug.

Scanned by CamScanner

(4) Not Reproducable

Whenever a bug is caught by the testing team, but the same bug cannot be reproduced by the development team in such situations the status of the bug is not reproducable.

Reasons due to which a defect can't be reproduced,

→ Inconsistent defects:- The defects which happen sometime and which dont happen some time are called as inconsistent defect. These type of bugs are not reproducable.

→ platform mismatch:- Due to any software or hardware configuration mismatch, the bugs might not be reproducable.

→ Build mismatch:- If the testing team has found a bug in software and the developement team will trying to reproduce the same bug into a different build then in situation the bug might not be reproducable.

→ Test Data mismatch:- If both teams are trying to reproduce the bug using different test data then at that situation, the bugs might not be reproducable.

→ Improper defect Report:- Due to wrong or incomplete defect report the development team might not have sufficient information to reproduce bug hence it can be a non reproducable bug.

(5) Postpone to next Release:- Whenever the testing team will finding a bug and the development team wants to fix this in next release, then will give the status postpone to next release.

In what situations we can postpone a bug to next release?

→ Whenever there is a minor or major bug found at the end of the release.

→ Whenever we find a bug in some module and that module has to be updated or removed completely in the next release.

→ When ever the defect is exposed only to the internal user.

Request for Enhancements / Improvements

→ RFE are nothing but suggestions given by the test engineers to enhance the functionality, usability or security of the application.

→ These suggestions can be accepted or can be rejected by the project manager.

Diff. b/w Change Request (CR) and Request for Enhancement (RFE)

CR	RFE
(1) It is given by the customer	(1) It is given by the test engineer.
(2) CR can't be rejected.	(2) RFE can be rejected.
(3) CR can ~~not~~ be for a deletion of feature or module	(3) RFE can't be for deletion of module.

Scanned by CamScanner

01/11/17

Globalization Testing

→ When ever we develop a application for multiple languages is called as globalization.

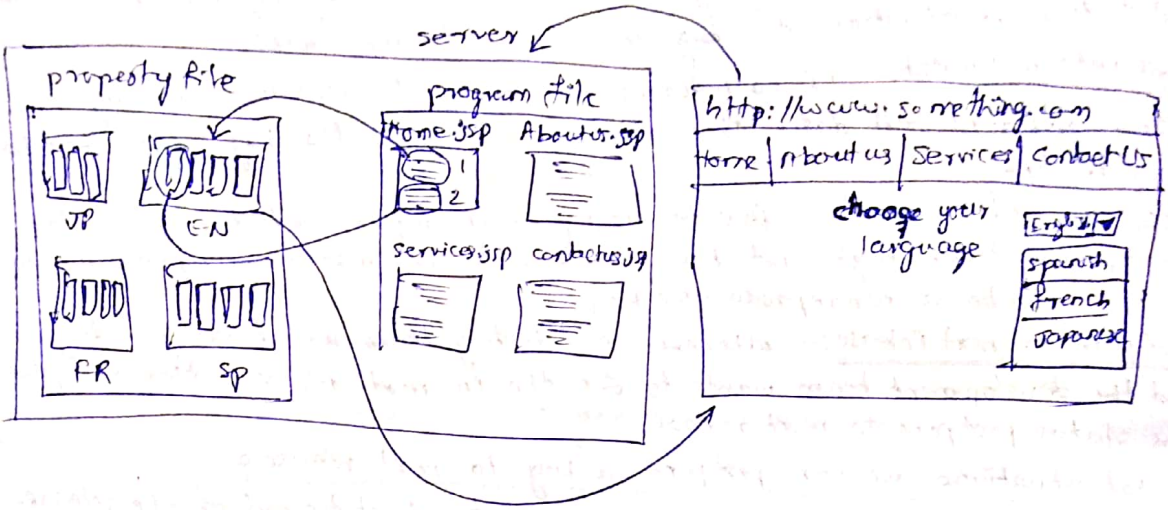
→ Testing the application which is developed for multiple languages is called as globalization testing.

Types:

① Internationalization Testing

② Localization testing.

English - En
French - Fr
Spanish - Sp
Japanese - JP



How Globalization works?

→ Globalization is possible through the concept of property files for the entire application for each and every language property files will be created by the language experts and it will checked by another lang. experts.

→ property files are nothing but are like note pad like files which is used for globalization.

Steps for the application working with multiple languages

→ As soon the user enters url of application in browser a request is sent from the browser to the server having page name and language code.

→ The server will check page name and run the corresponding jsp program.

→ This jsp program will contain two parts the first will check for the language code and establish the connection b/w prog. file and corresponding property file.

→ The second part of the code will check for the page name and corresponding property file and display it in the browser.

Scanned by CamScanner

① Internationalization Testing (I18N)

→ Checking the application for multiple languages is called as internationalization testing.

Here we check for 2 things,

① Whether application is displayin right language.

② Whether right content is displayed or not

→ Apart from these 2 things we also verify that the application is not broken if the language is broken.

How do we perform internationalization testing?

→ It is possible through the concept of pseudo codding.

Challenges:

→ The language displayed might be wrong.

→ The components might present at diff. places, when the language is changed.

How do we overcome this problem?

→ Using pseudo codding we will open the property file of the specific language and give language code as prefix.

→ For every component we will provide component name as sufix

→ Then we will run the application and check for the languages and the position of the components.

property file

Domestic
International

property file 1

sp_nacional_domestic
sp_internacional_international

property file 2

Fr_national_domestic
Fr_ international_ International

property file 3

JP_国内 _ Domestic
JP_国際 _ International

② Localization Testing (L10N)

→ Testing the application as per the local or country standards is called as localization testing.

→ In localization testing we check for,

① Currency formate

② Date and time "

③ Pin code "

④ Image "

⑤ Tool tip "

Scanned by CamScanner

(system search - local language)

① currency formate

Here we will check for currency formate of the country for which the application required to show.

India - ₹ 10,00,000.00
USA - $ 1.000,000.00
GER - € 1.000.000,00
Estonia - € 1000 000.00

② Date formate

Here we will check for the specific date formates of the specific country.

India
short date → dd-MM-YYY
Long date → dd-MMMM-YYY

USA
short date → dd-MMMM-YY
Long date → dddd/MMMM.dd,YYY

Germany
short date → dd.MM.YYYY
Language → dddd.d.MMMM.YYYY

Estonia
short date → d.MM.YY
Long date → d.MMMM.YYY.'a'.

③ Time formates (∵ search 'time formates' in system)

India
short time → HH:mm
Long time → HH:MM:SS

USA
short time → h:mmm:HH
Long time → h:mm:ss:H

Germany
short time → H:mm
Long time → H:mm:SS

④ Pincode formate

The pin code formate for any application will be as per the country standards

Bangalore - 560 001 - 6digit
estonia - 15165 - 5digit
US-California - CA 94025

⑤ Image formate

To check the images for localization we've to check for two type.

→ If there is no text in the image then same image can be used for diff. languages.

→ If there is text in the image then separate image has to be used for diff. languages

Scanned by CamScanner

Tool tip

For tool tip if the language is changed its formate also should chang as per local language. To check it using pseudo coding.

Assignment

NNA Japan

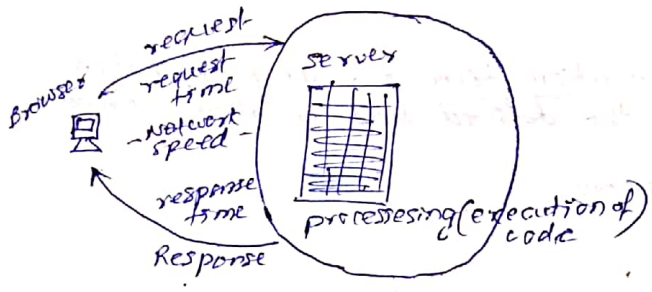
ejazah.com

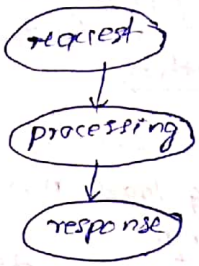
2/11/2017

Performance Testing (Non-functional)

→ Testing the execution ~~the execution~~ time of application by applying load is called performance testing.

→ Load means no. of users using the application at a particular amount of time





performance testing tools

1. Load Runner
2. J meter
3. Rational Robot
4. Neo Load.

Total delay (or) App delay = req. time + response time + processing time.

→ When ever a user uses the application a request is sent to the central server where the application source code is running and after the execution of the code a response will be sent to the user. Hence, the total time delay for any functionality is the sum of the request time, response time and execution/processing time.

→ When a large no. of users are using the application then at that situation the application will be receiving a lot of requests for which the application will execute and send a response. In such situation when a lot of load is put into the application and the application capacity reaches its boundries then the application might become slow or it can crash. Hence, at any movement of time if there is an application which is used by a lot of users then the application owner must be aware of the application's capacity to handle the load. So, for such applications performance testing is mandetory.

How performance can be done?

→ Performance testing is fully automated testing

→ performance testing is done by some tools like, J meter, Load Runner, NeoLoad, etc.

→ In any of the automation testing tool we ~~use~~ specify the application, the no. of users using them (define the load).

Scanned by CamScanner

→ The performance testing tool will send the equivalent amount of random requests into the server and it will check for the execution speed and execution time.

→ If the execution time is more than the defined execution time then the test is failed otherwise the test is passed.

Types of performance testing

① Load testing

Testing the application by ~~applicatior~~ applying load equal to or less than the capacity for which it is defined is called load testing.

S/w → Defined load 1000000 (1M)
Applied load ≤ Defined load

1500000 ✗
1000000 ✓
800000 ✓
700000 ✓

→ We do this to verify the stability & execution time for desire no. of users defined.

② Stress testing

Testing the stability and execution time for any application by applying load more than the defined load is called is called stress testing.

defined 1000000

1500000 → S/w
1200000 → S/w
1400000 → S/w
2000000 → S/w

③ Volume testing

Testing the stability and execution time of the application by transfering huge volume of data through it is called as volume testing.

④ Endurance / Soak testing

→ Applying load to any application equivalent to the defined load for a very long amount of time is called as endurance/soak testing.

→ eg: the app. with defined load 1000000 user then we will apply load for 1000000 continuesly for 1 week is called endurance/soak testing.

Scanned by CamScanner

Static and Dynamic

Testing
- Dynamic testing
  - Functional
  - Nonfunctional
- Static testing
  - Review, etc

Static	Dynamic
① It can be done without executing code	① It can be done by executing the code.
② can be done manually and with some static analysis tools.	② can be done by giving set of inputs, examined it's output and compared as per expected.
③ can be started early stage / at before the development cycle starts. eg:- By verifying the user requirements.	③ can be done after development of the software components.
④ Types of defects find in static testing are: Missing requirements, design defects, syntax errors, etc.	④ Types of defects find in dynamic testing are: variables are not constant, deviation in actual and expected o/p's.
⑤ Type: Review, Inspection, walkthrough	⑤ Types: Unit testing, Integration testing, system testing, Acceptance testing.
⑥ static testing find bug before you compile.	⑥ finds bug after compilation, linking.
⑦ static testing is about prevention (bugs)	⑦ Dynamic testing is about cure (finding and fixing of the bugs).
⑧ More cost effective compared to dynamic.	⑧ Not much cost effective compared to static
⑨ Done in verification stage	⑨ Done in validation stage.
⑩ Gives 100% statements coverage	⑩ Doesn't give 100% statement coverage.

Scanned by CamScanner

6/11/17

Accesibility Testing / ADA (508 compliance Testing)

↳ American Disability Activity

→ Testing the application from the physically challenged person's point of view. This testing is done as per american disability act 508.

Characteristics

→ For every action there should be proper response.

→ All the features should be accessible through keyboard.

→ Red and green colour shouldn't be used.

Test Plan

(O - S - T_m - A - R - M_p - A - R_R - S - D_t - T_E E_E - T_A - D - R_N - T = 16)

It is document which contains the details of feature testing activity.

① **Objective**

→ It is the aim for preparing the test plan and the entire testing activities which are going to be conduct further.

→ In this section we will mension all the goals which should met at the end of testing.

② **Scope**

→ Here we mesion 2 things,

ⓐ features to be tested

ⓑ features not to be tested.

How do we decide which features we shouldn't test?

→ If there is any document, FAQ, Licences, etc. created by the technical writing team. That means it is created by technical team and verified/tested by another technical writer. Then in such case we shouldn't verify those documents.

→ If there is any external module added to the application then we will not test the external module but we will test the integration of that external module with the application.

→ If there is any external link present in the application then we will just test the functionality of the link but not the application to which this link takes.

eg:- Help file

FAQ

Facebook link

twitter link

Google+ link.

external link eg:- fb, G+, twitter

③ Testing Methodologies

Based on my application we will check which testing methodology we should use and which we shouldn't used.

ⓐ standalone app.
ⓑ Web-based app.
ⓒ Mobile based app.

Functional, Integration, System, Smoke, Regression, sanity, compatibility, usability, Ad hoc, performance, LION, ISN, Accessibility, Interrupt based, etc.

④ Assumptions

→ For any project there will be some assumptions made related to technology, finance and ~~manpower~~ resources.

→ In such cases we assume these 3 things will not ~~the change~~ constant (or) not changing as per the needs of ~~team~~ team as well as the project.

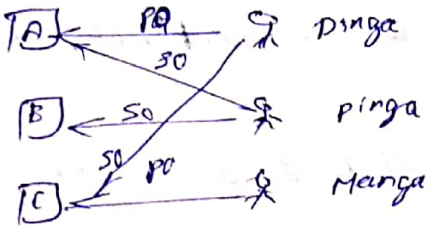
⑤ Risks

→ Risk is a -ve event which has not happed and may not happed in future also and if it happens it will have negative impact on the bussiness of app.

→ When ever we make some assumptions related to finance, technology and resources, these may fails. ~~in terms of risk~~ and hence these always are of risk.

⑥ Contingency plan / Mitigation plan

→ It is a backup plan if risks become reality in such case the backup plan is supposed to avoid or minimize the loose



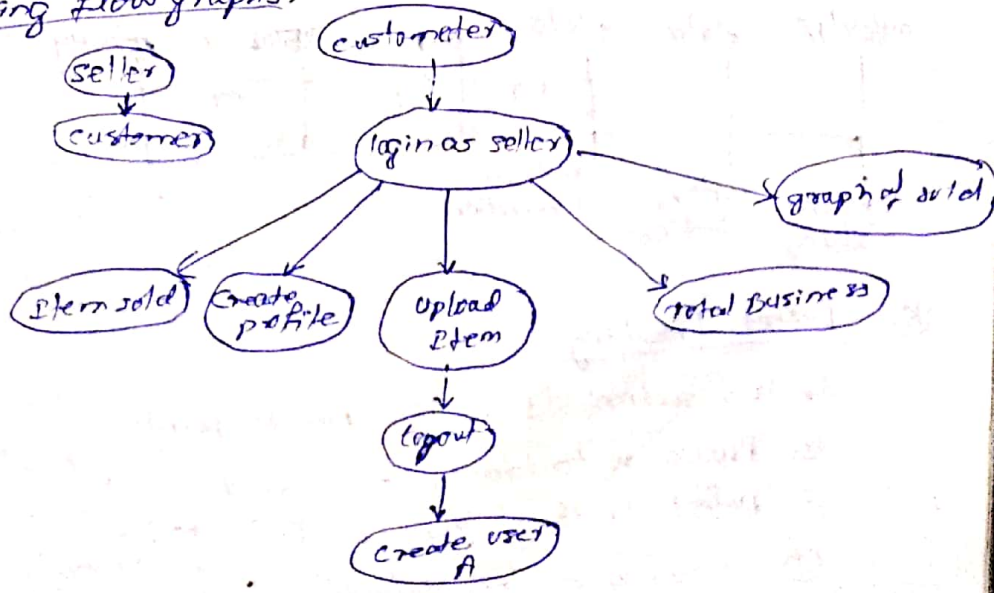
3/11/17

⑦ Approach

→ In this section of the test plan we will mension the approach in which the application should be tested.

→ This is done in two ways,

ⓐ writing flow graphs.



Scanned by CamScanner

⑦ Highlevel scenarios

→ create seller account
→ Login as seller
→ upload Items into panel
→ Logout from sellers A/C
→ Go to App Home page
→ Search Item
→ Add Item to cart
→ Goto checkout
→ Make payment
→ Check Invoice
→ Login as seller
→ check for sold Item
→ Logout.

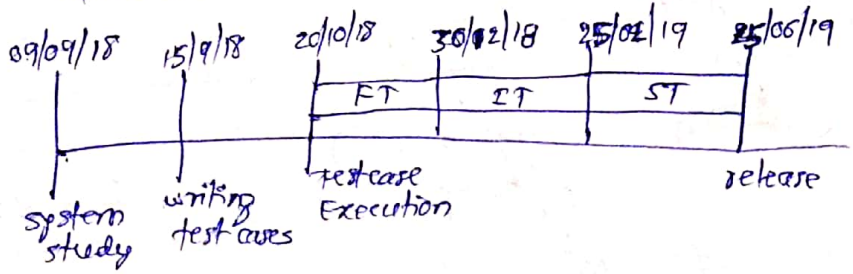
⑧ Roles and Responsibilities

→ In this section the role and responsibility for each and every team member mentioned for Test Manages, Test Leads, Sr.T.E, T.E, freshers, all people of testing team will be mentioned with their roles and responsibilities

→ Their roles and responsibilities will be decided as per experience and expertize.

⑨ Schedules

→ In this section when exactly each and every activity should start and end will be mentioned and date will be specified.



⑩ Defect Tracking

In this section there will be 4 points,

① Procedure to tract the defect – Defect/Bug Life cycle
② Defect tracking tool → Jira, Mantis, Bugzilla, ALM, TFS
③ Severity of the defects
④ Priority of the defects.

Scanned by CamScanner

(11) Test Environment

(a) Hardware Environment
- RAM
- Harddisk
- Motherboard
- Processor
- Graphics / VGA

(b) Software Environment
- OS
- Browsers
- Database

(c) Procedure to Install the S/W

5/11/17

(12) Entry and Exit Criteria

Entry Criteria

The minimum conditions required to be meet before starting testing the application is called as entry criteria.

Entry criteria for functional Testing

→ Minimum functionalities should be ready.
→ Application should be installed with proper environment.
→ White box testing should be done.
→ Test cases should be ready.
→ Test data should be ready.
→ Resources should be available

Entry criteria for Integration testing

→ Both modules should be ready and stable. ✓
→ Application should be installed with proper environment.
→ White box testing should be done.
→ Test cases should be ready
→ Test data should be ready
→ Resources should be available
→ Functional testing should be passed ✓

Entry criteria for system testing

→ All the modules should be ready. and stable ✓
→ Test environment is similar to production environment. ✓
→ White box testing should be done
→ Test cases should be ready
→ Test data should be ready
→ Resources should be available
→ Functional testing should be passed ✓
→ Integration testing should be passed. ✓

Scanned by CamScanner

Exit Criteria

Minimum conditions required to stop a type of testing is called as exit criteria.

Generic
× Exitentas for Testing

- Min. no. of bugs :- no app. is bug free hence, the min. no. of bugs which can be tolerated has to be defined. This is done as per the weightage of bugs.

1 critical	= 3 major
	= 5 minor

Budjeet – If the budjet for testing activity is over then we should stop testing.

Release Date :- If the application is released and no further enhancements are going to be added then we should stop testing.

Note:

→ For functional testing min 75% of test cases should be passed.

→ For Integration testing min 90% of test cases should be passed

→ For system testing min 95% – 98% of test cases should be passed.

(13) Test Automation

(a) Features to Automate

→ All the features which are working properly under stable will be tested for smoke and Regression from the second release.

→ Due to repetetive nature of testing those features will be automated.

(b) Features Not to automate

→ the new ~~and~~ & unstable features tested manually because through manual testing more care can be given.

(c) Automation tool

Depending up on the type of application we will choose like, selenium, UFT(QTP), selendroid, Appium, Test Complete, etc.

(d) Automation framework

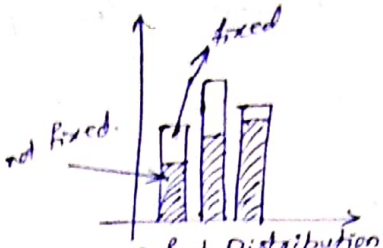
Data driven, Hybrid, keyword driven, etc.

Scanned by CamScanner

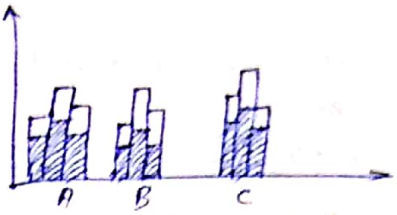
(14) Deliverables

- Deliverables are the documents given to the customer, following are diff. type of deliverables.

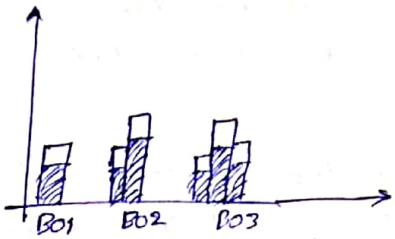
(a) Test cases
(b) Traceability matrix
(c) Test Execution Report
(d) Defect Report
(e) Release Note
(f) Test scripts
(g) Graphs and matrix.



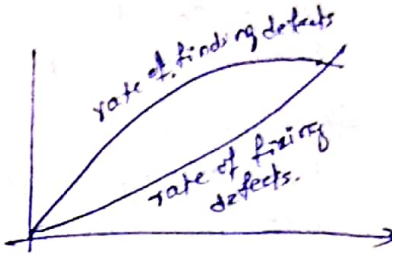
Defect Distribution graph



module wise graph



Build-wise graph



Defect Trend Analysis

Module	critical		Major		Minor	
	fix	not fixed	fix	not fixed	fix	not fixed
A	30	19	30	13	6	17
B	15	20	15	2	5	18
C	15	5	13	1	7	12

Module wise Matrix

Test Engineer	critical		Major		Minor	
	fixed	not fixed	fixed	not fixed	fixed	not fixed
Visal	5	20	12	35	31	37
vinod	13	31	22	12	30	15
Sangini	15	25	12	36	13	21

performance Matrix

(15) Release Note

The release note contains,

(a) List of pending or open bugs
(b) List of features which are added/ modified/ deleted.
(c) Platform in which the application is tested.
(d) List of bugs fixed in current release which were found in last release.
(e) procedure to install the software
(f) Version of the software.

(16) Templates

→ The outlines for all the diff. documents will be created as template.
→ The diff. types of templates required,

(a) Test case template
(b) Traceability matrix template
(c) Test execution Report template
(d) Defect tracking template.
(e) Release note template
(f) Graphs and Matrices template.

Scanned by CamScanner

9/11/17 User acceptance Testing / Acceptance testing

→ Testing the end to end bussiness of the application with real time data and real time scenarios inorder to ensure that the s/w is able to work in real time environment and can efficiently handle the bussiness.

→ Why should we perform UAT?

A s/w is created to handle a bussiness. If the s/w is not able to handle the real time scenarios then it can be a huge loss to the bussiness and the s/w might fail. Hence we perform UAT.

Types of UAT

Alpha testing: Testing the end to end bussiness of the application with real time data and with real time scenarios before installing installing into the production is called alpha testing.

→ In short Alpha testing is performing is done before launching

→ It can be done by, IT division of client, employees of customer who are good at bussiness of s/w, test engg. of company working under the supervision of the customer.

→ Alpha testing can be performed for all types of s/w. eg: Gaming app, social n/w app., security related s/w.

Beta testing: Performing UAT after launching the application into production done by the end users is called as Beta testing.

→ Why Beta testing is necessary?

It is done for those applications which are to be used by many people or many end users like, communication realated app., social n/w app., gaming apps, online booking apps, etc.

→ As these types of apps used by many users and end user openion & feed back is necessary Hence beta testing is important.

How to log a bug with Defect Tracking tool

① Enter the valid url for the application – Log in page should be displayed

② Enter valid user id & pwd for app –

③ click on report issue

④ Enter category of project name.

⑤ select reproducibility.

⑥ select severity – feature, trivial, text, tweak (overlap), major, minor crash.

⑦ select priority – none, low, normal, high, urgent, immediate, etc

⑧ platform – RAM – 5GB
OS –
OS version –

Scanned by CamScanner

(9) Enter summary — Project.Name/FeatureName/SubfeatureName/.

(10) Description — when customer is trying to login with valid data then he is receiving a blank page.

(11) steps to Reproduce —

(12) additional info — test data

(13) upload file —

(14) select public / private

(15) submit Report

How will you avoid duplicate bugs?

(1) click on view issues

(2) Enter: project name | feature name | subfeature name

→ all the bugs related to the sub feature will be displayed and from here if some body has already logged the bug then we will not log the bug & avoid creating duplicate bugs.

Scanned by CamScanner

14/11

White Box Testing (WBT)

"Testing the bussiness logic of the code is called as WBT."

→ In WBT the actual implimentation of the code has to be tested hence we will test all the logics that are being implemented in the code while performing WBT.

→ To perform WBT we should have,

① Programming knowledge.

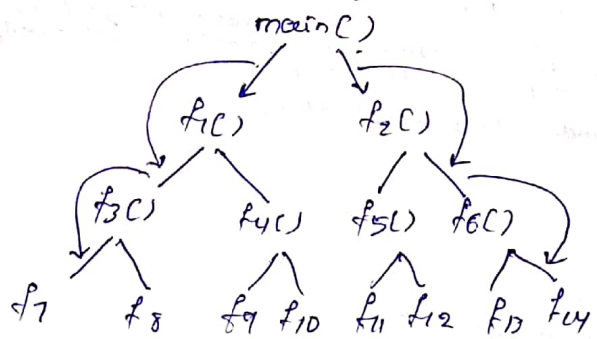
② knowledge of internal implementation of code.

→ white box testing can be done in two ways.

WBT — Manual
— Automation.

① Path Testing

Testing all the independent paths present in application is called as path testing.



→ We conduct path testing by,

ⓐ creating flow graph

ⓑ Testing the flow of program in individual paths.

② ~~Loop~~ Condition Testing

Testing all the if & else conditions is called as condition testing

```
if (condition 1) {
    st. 1
}
else { st 2 }
```

→ for the code above first we will pass true value we will check statement 1 should be executed. then we will pass false value and we will check st 2 should be executed.

Scanned by CamScanner

```
if (cond.1) {
    st.1 (script)
}
else if (cond 2) {
    st 2
}
else { st 3 }
```

→ For the code above we will first pass true value for condition, then we will check script, should be executed.
→ After the we'll pass true for condition 2 and we'll check script 2 should be executed.
→ Then we'll pass false value for all the conditions script 3 should be executed.

```
switch (condition) {
    case 1 : st 1;
             break;
    case2 : st 2;
             break;
    case3 : st3;
             break;
    default : st 4;
}
```

→ For the switch case first we'll pass true value for case 1 then we'll check st 1 should be executed.
→ Then we'll case 2, st 2 should be executed
→ " case 3, st 3 "
→ Then we will pass false value for all case 1, case 2, case 3. default st. should be executed.

③ Loop Testing

ⓐ While
ⓑ do-while
ⓒ for
ⓓ Enhanced for

Entry criteria →
```
while (condition) {
    script 1
}
```
exit criteria

→ In loop testing we'll test the loop for each and every cycle for the condition given in addition to the entry criteria and exit criteria of the loop.

→ for the above loop untill the condition true script 1 will be executed hence for each and every cycle the loop should be tested.

→ In addition to that we also have to check by providing a true condition so that the flow of program enters the loop and also the neccesary conditions will be achieved inorder to exit from loop.

④ Testing from memory point of view (WBT)

What are the diff. reasons because of which code ~~size~~ size is increased?

ⓐ due to unoptimized code

```
class A5
public void (String a) {
    int a;
    int b;
    int c;
    a=10;
    b= 5;
    c= a+b;
sop (c); }}
```

```
sop (10+5);
```

(b) Not using functions/methods

→ functions are used to reuse piece of code again and again no. of times.

→ If this is not used code size will be high.

```
class A {
  psvm(S[] a) {
  int a, b, b=2, c;
      c = a+b;
  int p=3; q=6; r;
      r = p+q;
  int m=15, n=10, o;
      o = m+n;
  int x=25, y=17, z;
      z = x+y;
  }
}
```

optimized →

```
class B {
    psvm(S[] a) {
  Sop( B.add(3,6));
  Sop( B.add(5,10));
  Sop( B.add(15,16));
  Sop( B.add(25,17));
  }
  public static int add (int a, int b){
      return (a+b);
  }
}
```

(∵ we can call directly add(2,3) because add method present in same prg.)

(c) Not using inbuilt functions/methods

→ In every programming language there will be some sort(), sear(), etc. If the programmer is not using the inbuilt code and he is trying to create own methods, this can also will increase the size of code.

(5) WBT from performance point of view

(a) what are the diff. reasons due to which the code can be slow?

(a) using if-else-if in place of switch

```
if (cond 1)
    ≡ S1
else if (cond 2)
    ≡ S2
else if (cond 3)
    ≡ S3
else
    ≡ S4
```

```
switch (cond)
  case 1 : ≡ S1
           break;
  case 2 : ≡ S2
           break;
  case 3 : ≡ S3
           break;
  default : ≡ S4
            break;
```

(b) while working with local AND and OR.

OR :- If there are two options one has higher probability of being selected then that option should be kept at first. Because in OR logical statements if any one logical i/p true o/p will be true.

Scanned by CamScanner

```
if (A||B){
    ≡ script
}
```

A	B	R
T	T	T
T	F	T
F	T	T
F	F	F

→ For example 80% of times A is supposed to be true and for 20% of time B should be true then A should kept first in conditional statements.

<u>AND</u> :- while working with logical AND if there are two options then the option which is most likely to be <u>false</u> should be kept <u>first</u>.

→ for eg. if 80% of times option A is supposed to be true and 20% of time option B is supposed to be true then option B should be kept at first.

```
if [ B && A) {
    ≡ script
}
```

A	B	R
T	T	T
T	F	F
F	T	F
F	F	F

Scanned by CamScanner