INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

Examinations Control Office

Examination	B TECH VI SEMESTER END EXAMINATIONS REGULAR JUNE 2025 REG UG20		
Month & Year	1-Jun	**Date**	23/06/2025
Course Name	SOFTWARE QUALITY ASSURANCE AND TESTING		
Course Code	ACIC02	**E-Code**	3487

Instructions to Evaluators

- Evaluators should spend at least 3-5 minutes on one answer booklet during the evaluation.
- Evaluators should cross check that marks are allotted for all the attempted questions.
- The marks should be assigned fairly according to the mark distribution specified in the scheme of evaluation.
- For questions that were attempted incorrectly, evaluators are required to award zero marks.
- The evaluator must give a proper justification in case of any mistakes identified in the marks provided.

START WRITING FROM HERE

Q.No.

1.b. Differences between Verification and validation (V & V) in the context of Software Testing :-

(i) Verification :- In the content of software Testing "Verification" can refer to "how we are doing, Rather than what we are doing". In Verification we check if we are doing it correctly accordingly and what we are doing is meeting the requirements of the client.

* Mainly check that we are doing what client asked., and Methods we used

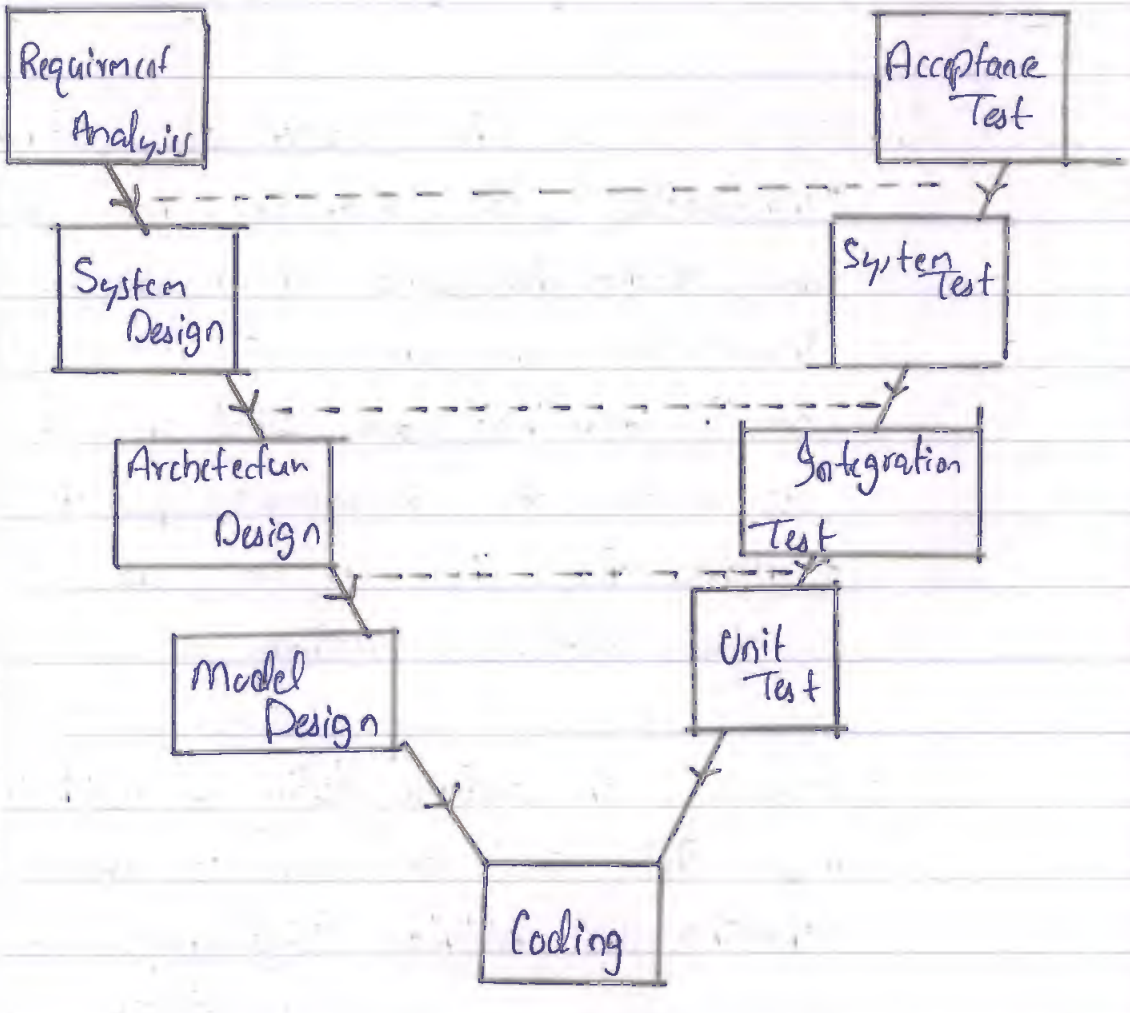
(ii) Validation :- In software testing, validation will imply that what we did is correct or not. checking for mistakes, fixing bugs and valida-ting based on users specifications. Validation means "what we are doing", and is correct according to the clients specifications

* Mainly check for any bugs and defects in the, software product.

⇒ Now, let us see the V-model and look into phases with difference

Q.No.

Verification and Validation Model :-



In verification and validation Modal The design (planning) and validation (Testing) are done parallelly. The defects in design can be found in early stage preventing costly change later.

Q.No.

* In Verification, The requirements are analyzed and validated in Acceptance testing and the assumption and our understanding of Requirement is checked

* In next phase, System design, the layout and compone t are checked and Tested in System Testing

* Archetecture design is tested using Integration testing to check the modules are interacting properly

* Modules designe is tested using Unit testing which include functionality, logic, reliability Testes.

In Software Testing the Verification and validation are two different goal goals but work togeather in hand in hand. This is required imported due to proactive check rather than reactive checks.

(ii) Role of Test Automation :-

In the modern software Testing, Manual Testing alone is not practical, . So Many

Q.No.

Softwave are developed to we a tools in Test Management, executing Tests, performce Testing and also for automation. These Tools help us to do work efficiently while minimizing mistakes.

Test Automation :-

The Test automation play an important role in our modern Software Testing activities. They help automate the repetative task and improve accuracy. Test Automation can improve the Test coverage and minimize the testing errors. Not all Tests need to be automate. The test automation initial stage is time taking and need skilled testers and developers to automate the testing. And the automation need to changed and modified according to the UI-change in the software. There are many automation tools available in the market making the Testing reliable and efficient

Automation Tools :-

Selenium, JUnit, JIRA and PractiTest etc,

Q.No.

1.a Primary Objectives of Software Testing :-

Software testing is used to check the functional and non-functional requirement of a software product. And verify if it is ~~Satisfying~~ satisfying the client-specifications. There are some primary objectives of Software Testing

- Validate the user specifications
- Find bugs and defective modules
- Gain Trust and Confidance with the client.
- Improve the software Reliability.
- Enhance the useability, operability of the software
- Find the Threshold or limit or breaking point of the software
- Minimize the error rate during the production
- Maintain Standards while following the Quality Assurance factors.
- Fix and Report bug during the Testing phase

The main Objectives of the Software

Q.No.

testing is to ensure that the software product is robust, reliable, efficient and finally meet the client requirement and standard while maintaing the product Quality.

* Here are some software testing action that are used to complect the Testing Objectives;

(i) Test Planning :- Identify scope, stategies, allocate resources and manage team.

(ii) Test Design :- Understand and write the Test Cases

(iii) Test environment Setup :- Setup the producti into the testing environment

(iv) Test Execution :- Run the product and check the test cases

(v) Defect Reporting :- Document the defect into a report.

(vi) Test Closure :- End or terminate / exit the testing phase after checking the actions.

Q.No.

Power of Testing :-

Testing plays an important role in the Software development lifecycle (SDLC) to ensure there aren't any defects, bugs and the software is reliable and meet the users expectations. The Testing help to gain confidance and the customer trust. Proper Testing will ensure that the software product meets the user specification and higher the Software quality the more the customer Satisfaction.

Testing the functional requirements and non-functional requirement like Maintainability security, efficiency, usability, Robustness, reliability Accessability, understandability and Operability will satisfy the customer if it is properly executed.

Q.No.

2.a Testing a function in Context :-

The concept of "Testing a function in Context" refers to testing a specific module in a system. Which is similar to the unit Testing. Where a particular functionality / feature or Module is seperated and tested from a System.

Example :-

→ In a e-commerce application, testing a specific module like payment gateway.

Incrimental integration :-

Incrimental integration is an integration approach where modules are integrated incrementally one after the other and also tested parallelly.

→ It is a simple approach widely used by one-manned teams

→ Example :- Integrate a module and test its interaction with other module

Q.No.

Top-down Integration :-

Top-down integration is an integration approach where the higher level Modules are first integrated and the moved down-wards to add the other module based on their level of hierarchy in the architecture.

* In top-down integration testing, "studs" are used in the place of missing values.
* The top-level module are tested first and low/bottom-level modules are Tested late

Bottom-up Integration :-

Bottom-up integration is an integration approach where the lower level Modules are tested first & integrated and they move upwards to add higher level modules based on their level of hierarchy in the architecture.

* In Bottom-up integration testing, "drivers" are used in the place of Missing value as the placeholders.
* In Bottom-up approach, lower-level Module are tafer first and higher/Top-level Module are tested late

Q.No.

Sandwich Approach :-

Sandwich integration is an integration approch where both top-level and bottom level Module are integrated simultaneously layer by layer. It is the combination of both top-down integration approch and Bottom-up integration approch. which gives the effect of layering a sandwich.

* In sandwich Integration testing both "Stubs" and "drivers" are used as placeholder in the place of Missing Modules

Big-Bang Integration :-

The Big-Bang integration a complex integration approch where all the Modules are integrated all at once making it Chaotic and more complex. All modules are integrated and tested at once.

Q.No.

2.b Boundary value analysis (BVA) and decision table :-

The Boundary value and decision table are methods used in Black Box testing which help improve the Test coverage in the System testing.

(i) Boundary Value analysis (BVA) :-

BVA is a Black Box testing technique that is used to identify the Boundary case ie, edge cases in a Software test. The edge case, play an important role in covering the more testing rounds. There are mainly three da Boundaries, Minimimum Boundary : When the testcase for the minimum value is calculated. Maximum Boundary, where the maximum Input that the software can take is calculated. Finally the null -value case where the zero or empty string is given as the input.

Example :-

let us see, the BVA Testcase for voting eligibility

Q.No.

: Condition, age >= 18, nationality = Indian

Id	age	nationality	expected	Actual
1	10	IN	F	F
2	18	IN	T	T
3	0	IN	F	F
4	100	IN	T	T

(ii) Decision Table :-

It is a structure testing approch which is a part of Black Box testing. The data is tabulated. It is used to check the multiple ifelse chain.

Example :- loan approval System, Exam pass verification, spam detection etc.,

Q.No.

Factors that Influence Software Reliability :-

→ Code Complexity, The complex codes make it harder to keep track of every Module

→ Experience, Proper balanced team with freshers and sen experienced people is needed.

→ Defect Reduction Efficiency (DRE), should be more for a reliable software

→ Lines of Code (LOC), More the code more the risk of failure

→ Quality Assurance Metrics, Measures like "Mean time fault, repair" etc effect the software reliability

→ Team Organization, The team should be properly manage, and developers and testers should Co-oridate with each other for building a reliable software

Q.No.

3.a Factors that influence system design test:-

(i) Improper understanding of requirements:
developer and tester should have proper understanding of the requirements and client consideration.

(ii) Lack of Co-ordination :
With out coordination and proper tracking of the design Test the Test could fail.

(iii) Communication :-
The different team should be able to communicate with the other team, to make the Task effective and Complete the System design Test

(iv) Hardware design :-
The hardware should be compatable with the software that is provided and the software and hardware should work properly.

Q.No.

(V) Planning :-

The phase should be planned, defined scope, startegies, resources and goal and Objective, are needed. And accountabilty should be verified.

(vi) Feedback and Acceptance :-

Continous user or client feedback and acceptance are essential for the success of the system design Test.

(vii) Hardware Software Integration :-

In system testing, the hardware and software are tested as whole and they need to be properly integrated and should co-ordinate as expected.

edral other factors like Reliability, Security, Manageability, Interface, Stress Environment, load, Stability, GUI, etc will impact the sfo system design test.

Q.No.

Metrics for Measuring testcase design effectiveness:

(i) Total Testcases :- Total number of Testcases designed.

(ii) Testcases Passed :- Total number of Testcase that passed the Test.

(iii) Testcases failed :- Total number of Testcase that failed the test.

(iv) Testcases Denied :- The total number of test-case that crashed the system

(v) Defect density :- How many defects are found with in a unit.

(vi) Fault Tollerency :- How many bug can be Managed by the software

and others like, Mean Time for defect Mean time for repair, Deffect recovery efficiency can also be used for Measuring the effectiveness of the testcase design

Q.No.

3.b. (i) Finate State Machine :-

finate State Machine is a mathematical representation of a software system. which can be used for generating testcases

→ A FSM have three components

- States
- Events / Input
- Transisions.

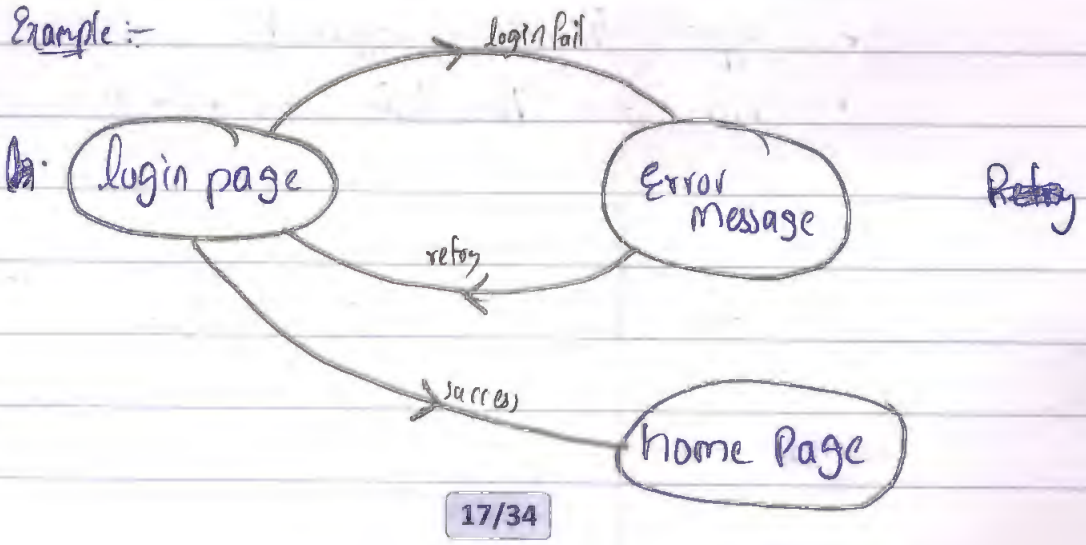
→ The software system is always in any one of the finete States.

→ Based on the events and Input the state is change

→ The changing of state based on the event or input is known as Transision.

→ Using these Transicions, we can generate the Testcases for the software.

Example :-



Q.No.

States : Login, Error, home

Events : success, failure, retry

Transition :-

→ Login $\xrightarrow{\text{fail}}$ error

→ Error $\xrightarrow{\text{retry}}$ login

→ login $\xrightarrow{\text{success}}$ home.

Using, above data we can controll a table for test cases

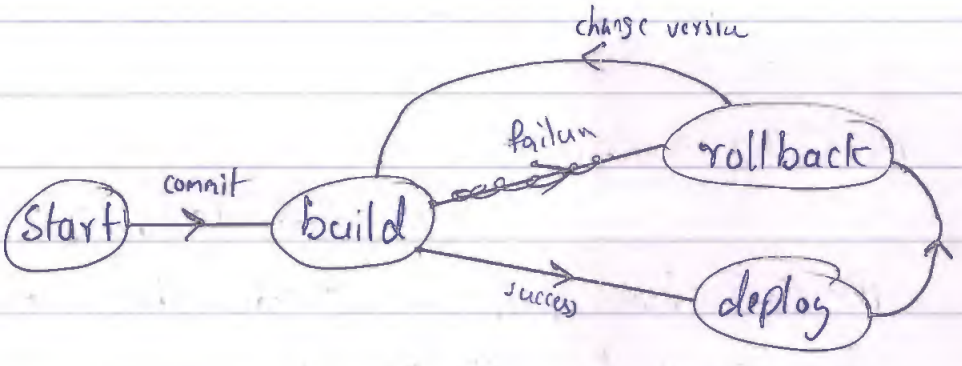
(iii) Transision Tour Method :-

The transision tour Method is a technique for finite state Machine, where the software is been in each state atleast once. This techniqu ensures the coverage of the states.

- helps to defect any error or Bug early.
- Simplify the Test approach
- Covers all the states avilable in a software.

let we look into the example,

Q.No.



Here to visite all nodes / state atleat once we write testcase in such a way that the system has been in each state atleast one

Q.No.	
5.a	McCall's quality factors :- In early 1970's James McCall have proposed factor that can be used to maintain the software quality. There based on three perspective and all to-gether have 11- factor. ii) Product Operations :- This perspective focuses on the operatability of the product like its functionality., Reliability and how close it is to the client requirments ⇒ factor :- 1, Reliability : how reliable it is. 2, Effency : Good use of resources. 3, Correctness : how close to client vision 4, Usability : Easy to use 5, Integrity : Security and authenticatio Where factors that are part of the product operation criteria, mainly focused on functionality.

Q.No.

(ii) Product Revision :-

The product revision refers to how easily the product can be modified and maintained easily.

Factors;

1. Maintainability :-

Easy to Maintain and change

2. Flexibility :-

flexible to changes.

3. Testability :-

Ability to be tested easily and cover more test.

(iii) Product Transition :-

Product Transition refers to the portability and reusability of the product factors :-

1. Portability :-

Used between different environments

2) Reusability :- easy to update and use

3) Interoprability :- Work with external Libraries and 3rd Party tools.

Q.No.

5b Software Quality Assurance :-

Software Quality is not just about testing. It is responsible for the overall end product and how close it is to the client's vision. And Quality Assurance function as a function in Agile and DevOps environment. It is used in every software ~~dep~~ development Models not only in the devops or Agile. It has been a norm which is inherited from the Manufacturing. The Software Quality assurance is responsible for Make the software product more efficient reliable and meet the client requirements. In the agile development Methodologies the the requirement are always changing so there complete Quality assurance teams are deployed to Make shure that the final product Meet the requirment of the client.

In Agile development, each ~~~~ iteration have its ~~one~~ requirements

Q.No.

and specification is the Software Quality Assurance plays as a function in the Agile development to maintain the Product Quality

Coming to the devOps, the Test Automation, Tracking and CICD pipelines we the Quality Assurance tools like Selenium, Junit, LoadRunner, JMeter, PractiTest TestNG, Bugzilla, Mantis, JIRA and for the CICD pipline, they we github action and build tools like Jenkins. So we can say that Software quality Assurance Work as a function in the devops also

Q.No.

7.b (i) Software fault Tolerance :-

Software fault tolerance can be explained as the ability of the software to work as usual even after encountering few defects or bug with crashing.

There two methods commonly used to improve the fault Tollerance

(a) Using duplicate or different Modules that perform the same task with different structure, Though one fails, the other will give the output.

(b) Having N- number of version of software working simultaneously. And the output is taken based on the voting method. Most generated output is the final Output.

Q.No.

(ii) Software Safty Assurance :-

Safty assurance include the factors like

⇒ Security
⇒ Reliability
→ Stress environment
→ and Scalability,

these factors make the software robust and fail proof. And prevent unauthorized access and data Breaches.

(iii) Failure Containment :-

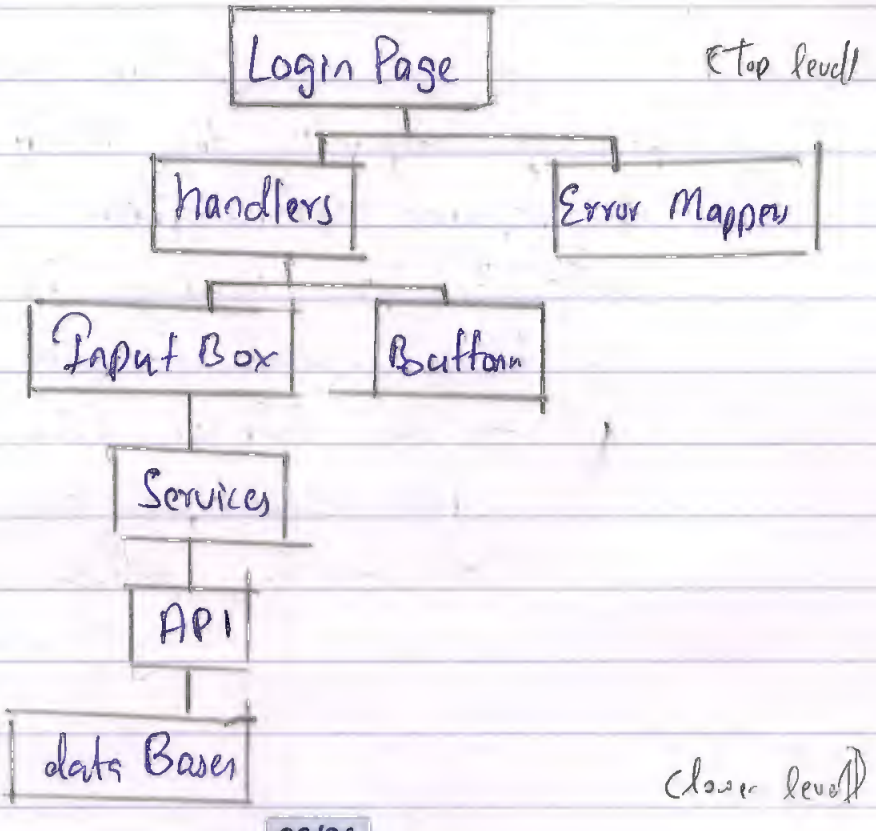
Failure containment is the process of seperating the failure or bugs in a software system. This will help the reliability by ensuring the failure are not being spread to other Modules.

⇒ The techniques like failure Containment Region tracking and error handling are used to perform failure Containment

Q.No.

7.a. Root Cause Analysis (RCA) :-

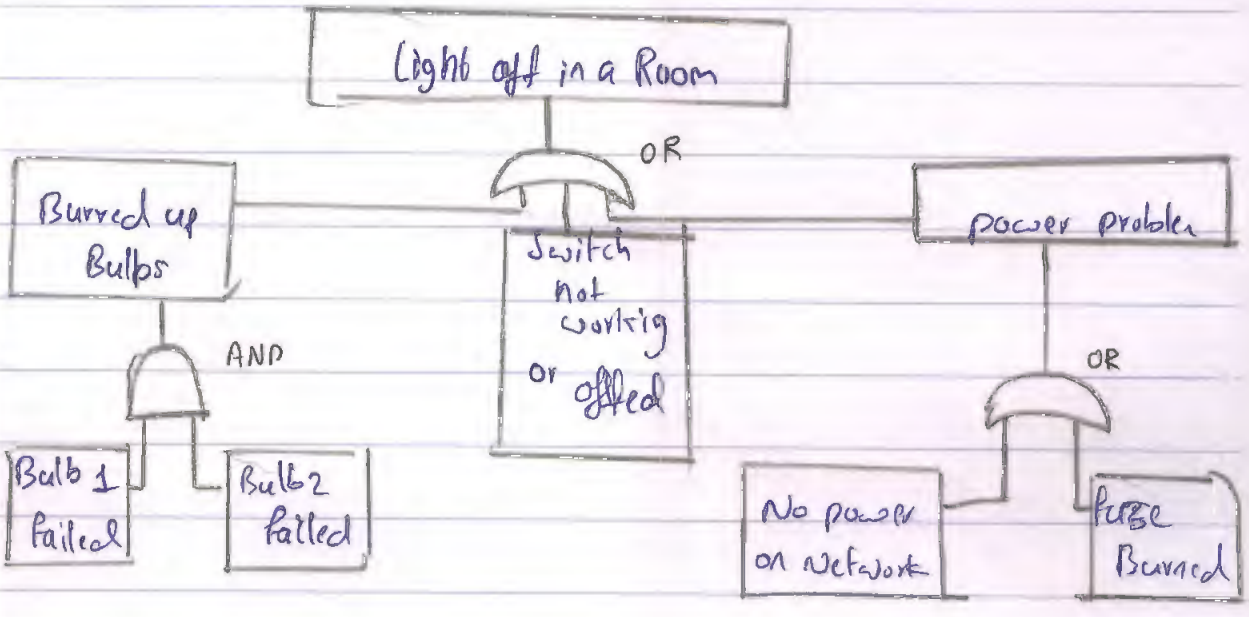
Root cause Analysis is an systematic approach used to find the root cause or origin of the Bug in the software. Techniques like FTA, and decision Trees are used to find the errors. It is a Top down ~~appro~~ deductive approach. ~~Ster~~ We Traverse from higher level Modules to the lower level dependent Modules that are under the effect of the error. And check each module. It is a mixture of top down - integration test and unit testing.



Q.No.

After the error is defected, then it is fixed and defect is prevented from going to the production.

Example of Using FTA :-



fault Tree analysis for "RCA"

Q.No.	

Q.No.	

Q.No.	

Q.No.	

Q.No.	

Q.No.	

ROUGH WORK

Content written here will not be considered for valuation