SOFTWARE TESTING QUESTION BANK

Software Testing Definitions

Black box testing – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

White box testing – This testing is based on knowledge of the internal logic of an application's code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Unit testing – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

Incremental integration testing – Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. done by programmers or by testers.

Integration testing – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

Functional testing – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

System testing – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

End-to-end testing – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

Sanity testing - Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix.

Regression testing – Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

Acceptance testing -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

Load testing – Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.

Stress testing – System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

Performance testing – Term often used interchangeably with 'stress' and 'load' testing. To check whether system meets performance requirements. Used different performance and load tools to do this.

Usability testing – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

Install/uninstall testing - Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

Recovery testing – Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

Security testing – Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks.

Compatibility testing – Testing how well software performs in a particular hardware/software/operating system/network environment and different combination s of above.

Comparison testing – Comparison of product strengths and weaknesses with previous versions or other similar products.

Alpha testing – In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

Beta testing – Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

FREQUENTLY ASKED QUESTIONS

1. What is the MAIN benefit of designing tests early in the life cycle?

It helps prevent defects from being introduced into the code.

2. What is risk-based testing?

Risk-based testing is the term used for an approach to creating a test strategy that is based on prioritizing tests by risk. The basis of the approach is a detailed risk analysis and prioritizing of risks by risk level. Tests to address each risk are then specified, starting with the highest risk first.

3. A wholesaler sells printer cartridges. The minimum order quantity is 5. There is a 20% discount for orders of 100 or more printer cartridges. You have been asked to prepare test cases using various values for the number of printer cartridges ordered. Which of the following groups contain three test inputs that would be generated using Boundary Value Analysis?

4, 5, 99

4. What is the KEY difference between preventative and reactive approaches to testing?

Preventative tests are designed early; reactive tests are designed after the software has been produced.

5. What is the purpose of exit criteria?

To define when a test level is complete.

6. What determines the level of risk?

The likelihood of an adverse event and the impact of the event

7. When is used Decision table testing?

Decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combinations. In a decision table the inputs are listed in a column, with the outputs in the same column but below the inputs. The remainder of the table explores combinations of inputs to define the outputs produced.

Learn More About Decision Table Testing Technique in the Video Tutorial here

8. What is the MAIN objective when reviewing a software deliverable?

To identify defects in any software work product.

9. Which of the following defines the expected results of a test? Test case specification or test design specification.

Test case specification.

10. Which is a benefit of test independence?

It avoids author bias in defining effective tests.

11. As part of which test process do you determine the exit criteria?

Test planning.

12. What is beta testing?

Testing performed by potential customers at their own locations.

13. Given the following fragment of code, how many tests are required for 100% decision coverage?

```
if width > length
  then biggest_dimension = width
    if height > width
         then biggest_dimension = height
    end_if
else biggest_dimension = length
        if height > length
           then biggest_dimension = height
      end_if
```

end_if

4

14. You have designed test cases to provide 100% statement and 100% decision coverage for the following fragment of code. if width > length then biggest_dimension = width else biggest_dimension = length end_if The following has been added to the bottom of the code fragment above. print "Biggest dimension is " & biggest_dimension print "Width: " & width print "Length: " & length How many more test cases are required?

None, existing test cases can be used.

15. Rapid Application Development ?

Rapid Application Development (RAD) is formally a parallel development of functions and subsequent integration. Components/functions are developed in parallel as if they were mini projects, the developments are time-boxed, delivered, and then assembled into a working prototype. This can very quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. Rapid change and development of the product is possible using this methodology. However the product specification will need to be developed for the product at some point, and the project will need to be placed under more formal controls prior to going into production.

16. What is the difference between Testing Techniques and Testing Tools?

Testing technique: – Is a process for ensuring that some aspects of the application system or unit functions properly there may be few techniques but many tools.

Testing Tools: – Is a vehicle for performing a test process. The tool is a resource to the tester, but itself is insufficient to conduct testing

Learn More About Testing Tools here

17. We use the output of the requirement analysis, the requirement specification as the input for writing …

User Acceptance Test Cases

18. Repeated Testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software component:

Regression Testing

19. What is component testing ?

Component testing, also known as unit, module and program testing, searches for defects in, and verifies the functioning of software (e.g. modules, programs, objects, classes, etc.) that are separately testable. Component testing may be done in isolation from the rest of the system depend-ing on the context of the development life cycle and the system. Most often stubs and drivers are used to replace the missing software and simulate the interface between the software components in a simple manner. A stub is called from the software component to be tested; a driver calls a component to be tested.

20. What is functional system testing ?

Testing the end to end functionality of the system as a whole.

21. What is the benefits of Independent Testing

Independent testers see other and different defects and are unbiased.

22. In a REACTIVE approach to testing when would you expect the bulk of the test design work to be begun?

After the software or system has been produced.

23. What are the different Methodologies in Agile Development Model?

There are currently seven different Agile methodologies that I am aware of:

1. Extreme Programming (XP)
2. Scrum
3. Lean Software Development
4. Feature-Driven Development
5. Agile Unified Process
6. Crystal
7. Dynamic Systems Development Model (DSDM)

24. Which activity in the fundamental test process includes evaluation of the testability of the requirements and system?

A Test analysis and design.

25. What is typically the MOST important reason to use risk to drive testing efforts?

Because testing everything is not feasible.

26. Which is the MOST important advantage of independence in testing?

An independent tester may be more effective at finding defects missed by the person who wrote the software.

27. Which of the following are valid objectives for incident reports?

i. Provide developers and other parties with feedback about the problem to enable identification, isolation and correction as necessary.

ii. Provide ideas for test process improvement.

iii. Provide a vehicle for assessing tester competence.

iv. Provide testers with a means of tracking the quality of the system under test.

i. Provide developers and other parties with feedback about the problem to enable identification, isolation and correction as necessary,

ii.Provide ideas for test process improvement,

iv.Provide testers with a means of tracking the quality of the system under test

28. Consider the following techniques. Which are static and which are dynamic techniques?

i. Equivalence Partitioning.

ii. Use Case Testing.

iii.Data Flow Analysis.

iv.Exploratory Testing.

v. Decision Testing.

vi. Inspections.

Data Flow Analysis and Inspections are static, Equivalence Partitioning, Use Case Testing, Exploratory Testing and Decision Testing are dynamic.

29. Why are static testing and dynamic testing described as complementary?

Because they share the aim of identifying defects but differ in the types of defect they find.

30. What are the phases of a formal review ?

In contrast to informal reviews, formal reviews follow a formal process. A typical formal review process consists of six main steps:

1. Planning
2. Kick-off
3. Preparation
4. Review meeting
5. Rework
6. Follow-up.

31. What is the role of moderator in review process?

The moderator (or review leader) leads the review process. He or she deter-mines, in co-operation with the author, the type of review, approach and the composition of the review team. The moderator performs the entry check and the follow-up on the rework, in order to control the quality of the input and output of the review process. The moderator also schedules the meeting, disseminates documents before the meeting, coaches other team members, paces the meeting, leads possible discussions and stores the data that is collected.

32. What is an equivalence partition (also known as an equivalence class)?

An input or output range of values such that only one value in the range becomes a test case.

33. When should configuration management procedures be implemented?

During test planning.

34. A Type of functional Testing, which investigates the functions relating to detection of threats, such as virus from malicious outsiders.

Security Testing

35. Testing where in we subject the target of the test , to varying workloads to measure and evaluate the performance behaviors and ability of the target and of the test to continue to function properly under these different workloads.

Load Testing

36. Testing activity which is performed to expose defects in the interfaces and in the interaction between integrated components is:

Integration Level Testing

37. What are the Structure-based (white-box) testing techniques ?

Structure-based testing techniques (which are also dynamic rather than static) use the internal structure of the software to derive test cases. They are com-monly called 'white-box' or 'glass-box' techniques (implying you can see into the system) since they require knowledge of how the software is implemented, that is, how it works. For example, a structural technique may be concerned with exercising loops in the software. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.

38. When should Regression testing be performed ?

After the software has changed or when the environment has changed

39. When should testing be stopped?

It depends on the risks for the system being tested

40. What is the purpose of a test completion criterion?

To determine when to stop testing

41. What can static analysis NOT find?

For example memory leaks

42. What is the difference between re-testing and regression testing?

Re-testing ensures the original fault has been removed; regression testing looks for unexpected sideeffects

43. What are the Experience-based testing techniques?

In experience-based techniques, people's knowledge, skills and background are a prime contributor to the test conditions and test cases. The experience of both technical and business people is important, as they bring different perspectives to the test analysis and

design process. Due to previous experience with similar systems, they may have insights into what could go wrong, which is very useful for testing.

44. What type of review requires formal entry and exit criteria, including metrics?
Inspection

45. Could reviews or inspections be considered part of testing?

Yes, because both help detect faults and improve quality

46. An input field takes the year of birth between 1900 and 2004 What are the boundary values for testing this field ?

1899,1900,2004,2005

47. Which of the following tools would be involved in the automation of regression test?

a. Data tester b. Boundary tester c. Capture/Playback d. Output comparator.

d. Output comparator

48. To test a function,what has to write a programmer, which calls the function to be tested and passes it test data.

Driver

49. What is the one Key reason why developers have difficulty testing their own work?

Lack of Objectivity

50.“How much testing is enough?”

The answer depends on the risk for your industry, contract and special requirements.

51. When should testing be stopped?

It depends on the risks for the system being tested.

52. Which of the following is the main purpose of the integration strategy for integration testing in the small?

To specify which modules to combine when, and how many at once.

53. What is the purpose of a test completion criterion?

To determine when to stop testing

54. Given the following code, which statement is true about the minimum number of test cases required for full statement and branch coverage?

```
Read p
Read q
IF p+q> 100
    THEN Print "Large"
ENDIF
IF p > 50
    THEN Print "p Large"
ENDIF
```

1 test for statement coverage, 2 for branch coverage

55. What is the difference between re-testing and regression testing?

Re-testing ensures the original fault has been removed; regression testing looks for unexpected side-effects.

56. Which review is normally used to evaluate a product to determine its suitability for intended use and to identify discrepancies?

Technical Review.

57. Why we use decision tables?.

The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs. However, if different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface. The other two specification-based tech-niques, decision tables and state transition testing are more focused on business logic or business rules. A decision table is a good way to deal with combinations of things (e.g. inputs). This technique is sometimes also referred to as a 'cause-effect' table. The reason for this is that there is an associated

logic diagramming technique called 'cause-effect graphing' which was sometimes used to help derive the decision table.

58. Faults found should be originally documented by who?

By testers.

59. Which is the current formal world-wide recognized documentation standard?

There isn't one.

60. Which of the following is the review participant who has created the item to be reviewed?

Author

61. A number of critical bugs are fixed in software. All the bugs are in one module, related to reports. The test manager decides to do regression testing only on the reports module.

Regression testing should be done on other modules as well because fixing one module may affect other modules.

62. Why does the boundary value analysis provide good test cases?

Because errors are frequently made during programming of the different cases near the 'edges' of the range of values.

63. What makes an inspection different from other review types?

It is led by a trained leader, uses formal entry and exit criteria and checklists.

64. Why can be tester dependent on configuration management?

Because configuration management assures that we know the exact version of the testware and the test object.

65. What is a V-Model ?

A software development model that illustrates how testing activities integrate with software development phases

66. What is maintenance testing?

Triggered by modifications, migration or retirement of existing software.

67. What is test coverage?

Test coverage measures in some specific way the amount of testing performed by a set of tests (derived in some other way, e.g. using specification-based techniques). Wherever we can count things and can tell whether or not each of those things has been tested by some test, then we can measure coverage.

68. Why is incremental integration preferred over "big bang" integration?

Because incremental integration has better early defects screening and isolation ability

69. When do we prepare RTM (Requirement traceability matrix), is it before test case designing or after test case designing?

The would be before. Requirements should already be traceable from Review activities since you should have traceability in the Test Plan already. This question also would depend on the organisation. If the organisation do test after development started then requirements must be already traceable to their source. To make life simpler use a tool to manage requirements.

70. What is called the process starting with the terminal modules ?

Bottom-up integration

71. During which test activity could faults be found most cost effectively?

During test planning

72. The purpose of requirement phase is

To freeze requirements, to understand user needs, to define the scope of testing

73. How much testing is enough?

The answer depends on the risks for your industry, contract and special requirements

74. Why we split testing into distinct stages?

Each test stage has a different purpose.

75. Which of the following is likely to benefit most from the use of test tools providing test capture and replay facilities? a) Regression testing b) Integration testing c) System testing d) User acceptance testing

Regression testing

76. How would you estimate the amount of re-testing likely to be required?

Metrics from previous similar projects and discussions with the development team

77. What studies data flow analysis ?

The use of data on paths through the code.

78. What is Alpha testing?

Pre-release testing by end user representatives at the developer's site.

79. What is a failure?

Failure is a departure from specified behaviour.

80. What are Test comparators ?

Is it really a test if you put some inputs into some software, but never look to see whether the software produces the correct result? The essence of testing is to check whether the software produces the correct result, and to do that, we must compare what the software produces to what it should produce. A test comparator helps to automate aspects of that comparison.

81. Who is responsible for document all the issues, problems and open point that were identified during the review meeting

Scribe

82. What is the main purpose of Informal review

Inexpensive way to get some benefit

83. What is the purpose of test design technique?

Identifying test conditions and Identifying test cases

84. When testing a grade calculation system, a tester determines that all scores from 90 to 100 will yield a grade of A, but scores below 90 will not. This analysis is known as:

Equivalence partitioning

85. A test manager wants to use the resources available for the automated testing of a web application. The best choice is

Tester, test automater, web specialist, DBA

86. During the testing of a module tester 'X' finds a bug and assigned it to developer. But developer rejects the same, saying that it's not a bug. What 'X' should do?

Send to the detailed information of the bug encountered and check the reproducibility

87. A type of integration testing in which software elements, hardware elements, or both are combined all at once into a component or an overall system, rather than in stages.

Big-Bang Testing

88. In practice, which Life Cycle model may have more, fewer or different levels of development and testing, depending on the project and the software product. For example, there may be component integration testing after component testing, and system integration testing after system testing.

V-Model

89. Which technique can be used to achieve input and output coverage? It can be applied to human input, input via interfaces to a system, or interface parameters in integration testing.

Equivalence partitioning

90. "This life cycle model is basically driven by schedule and budget risks" This statement is best suited for…

V-Model

91. In which order should tests be run?

The most important tests first

92. The later in the development life cycle a fault is discovered, the more expensive it is to fix. why?

The fault has been built into more documentation,code,tests, etc

93. What is Coverage measurement?

It is a partial measure of test thoroughness.

94. What is Boundary value testing?

Test boundary conditions on, below and above the edges of input and output equivalence classes.

95. What is Fault Masking ?

Error condition hiding another error condition.

96. What does COTS represent?

Commercial Off The Shelf.

97.The purpose of wich is allow specific tests to be carried out on a system or network that resembles as closely as possible the environment where the item under test will be used upon release?

Test Environment

98. What can be though of as being based on the project plan, but with greater amounts of detail?

Phase Test Plan

99. What is exploratory testing?

Exploratory testing is a hands-on approach in which testers are involved in minimum planning and maximum test execution. The planning involves the cre-ation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort, the objectives and possible approaches to be used. The test design and test execution activities are performed in parallel typi-cally without formally documenting the test conditions, test cases or test scripts. This does not mean that other, more formal testing techniques will not be used. For example, the tester may decide to use boundary value analysis but will think through and test the most important boundary values without

necessarily writing them down. Some notes will be written during the exploratory-testing session, so that a report can be produced afterwards.

100. What is failure?

Deviation from expected result to actual result

Multiple Choice Questions

1) The approach/document used to make sure all the requirements are covered when writing test cases

a) Test Matrix

b) Checklist

c) Test bed

d) Traceability Matrix

---

2)Unit Testing will be done by

a) Testers

b) End Users

c) Customer

d) Developers

---

3) To check whether we are developing the right product according to the customer requirements are not.
It is a static process

a) Validation

b) Verification

c) Quality Assurance

d) Quality Control

---

4) To check whether we have developed the product according to the customer requirements r not.
It is a Dynamic process.

a) Validation

b) Verification

c) Quality Assurance

d) Quality Control

---

5) It is a set of levels that defines a testing maturity hierarchy

a) TIM (Testing Improving Model)

b) TMM (Testing Maturity Model)

c) TQM(Total Quality Management)

---

6) A Non-Functional Software testing done to check if the user interface is easy to use and understand

a) Usability Testing

b) Security Testing

c) Unit testing

d) Block Box Testing

---

7)Beta testing will be done at

a) User place

b) Developers place

---

8) Retesting of a single program or component after a change has been made?

a) Full Regression Testing

b) Unit Regression

c) Regional Regression

d) Retesting

---

9) The testing which is done by going thro' the code is known as,

a) Unit Testing

b) Black box testing

c) White box Testing

d) Regression testing

---

10) This type of testing method attempts to find incorrect or missing functions, errors in data structures or externa Performance errors and initialization and Termination errors. It is called as

a) White Box Testing

b) Grey Box Testing

c) Black Box Testing

d) Open Box Testing

---

11) Phase Definition. It will come under

a) CMM Level 1

b) CMM Level 2

c) None

---

12) Software testing which is done without planning and Documentation is known as

a) adhoc Testing

b) Unit Testing

c) Regression testing

d) Functional testing.

---

13) Acceptance testing is known as

a) Beta Testing

b) Grey box testing

c) Test Automation

d) White box testing

---

14) Retesting the entire application after a change has been made called as?

a) Full Regression Testing

b) Unit Regression

c) Regional Regression

d) Retesting

---

15) Boundary value analysis belongs to which testing method?

a) Black Box testing

b) White Box testing

---

16)--------- means under what test environment(Hardware, software set up) the application will run smoothly

a) Test Bed

b) Checkpoint

c) Code Walk through

d) Checklist

17) What are the Types of Integration Testing?

a) Big Bang Testing

b) Bottom Up Testing

c) Top Down Testing

d) All the above

---

18) Which is non-functional software testing?

a) Unit Testing

b) Block box testing

c) Performance Testing

d) Regression testing

---

19) The process that deals with the technical and management issues of software development called as?

a) Delivery Process

b) Testing Process

c) Software Process

---

20) Executing the same test case on a modified build called as

a) Regression Testing

b) Retesting

c) Ad hoc Testing

d) Sanity Testing

---

21) Which is Black-Box Testing method?

a) equivalence partitioning

b) code coverage

c) fault injection

---

22) Informing to the developer which bug to be fix first is called as

a) Severity

b) Priority

c) Fix ability

d) Traceability

23) An Important metric is the number of defects found in internal testing compared to the defects found in customer tests, Status of test activities against the plan, Test coverage achieved so far, comes under

a) Process Metric

b) Product Metric

c) Test Metric

---

24) SPICE Means

a) Software Process Improvement and Capability Determination

b) Software Process Improvement and Compatibility Determination.

c) Software Process Invention and Compatibility Determination.

d) Software Process Improvement and Control Determination

---

25) Requirements Specification, Planning, Test case Bug Reporting & Maintenance This Life Cycle comes Under

a) SDLC

b) STLC

c) SQLC

d) BLC

---

26) Standards and procedures for managing changes in an evolving software product is called?

a) Confirmation Management

b) Confederation Management

c) Configuration Management

d) Comparability Management

---

27) What is correct Software Process Cycle?

a) Plan(P)------>Check(C)------>Act(A)----->Do(D)

b) Plan(P)------>Do(D)------>Check(C)----->Act(A)

c) Plan(P)------>Do(D)------>Act(A)----->Check(C)

---

28) TQM represents

a) Tool Quality Management

b) Test Quality Manager

c) Total Quality Management

d) Total Quality Manager

---

29) Optimization, Defect Prevention, and Quality Control. Its come under the

a) CMM Level 2

b) CMM Level 3

c) CMM Level 4

d) CMM Level5

---

30) Defects generally fall into the following categories?

a) WRONG

b) MISSING

c) EXTRA

d) All the above

ANSWERS:

1) D	11) B	21) A
2) D	12) A	22) B
3) B	13) A	23) C
4) A	14) A	24) A
5) B	15) A	25) B
6) A	16) A	26) C
7) A	17) D	27) B
8) B	18) C	28) C
9) C	19) C	29) D
10) C	20) A	30) D

SOFTWARE QUALITY ASSURANCE

1. Define software errors.
2. Define software faults
3. Define software failures.
4. List out the various causes of software errors.
5. Define Software Quality.
6. Define software Engineering.
7. Define quality control.
8. Show the relationship between failure, fault and error.
9. Define correctness.
10. Define portability
11. Define reusability
12. Define maintainability
13. Define tangibility
14. Define assurance
15. Define efficiency.
16. Define integrity
17. Define software testing
18. What are the three categories belonging to McCall's factor model?
19. Relate Quality assurance with Quality control. Justify your answer that QA is not QC.
20. Define Software Quality Assurance.
21. What are the three categories belonging to Product Transition Software Quality Factors?
22. Outline the different components of SQA Architecture.
23. Find the two contract review stages.
24. Label all the components in SQA architecture.
25. Show the process of object oriented model in flow chart format.
26. What are the main issues in the project development plan?

27. What are the main issues in the project's quality plan?
28. Which are main components of project life cycle components?
29. Who performs a contract review?
30. List out the different types of BBT.
31. List out the main components of software maintenance.
32. Define Software quality cost
33. Tell the way to conduct contract review process?

Short Question

1. Compare software errors, software faults and software failures.
2. Compare QC and QA.
3. Outline the Product view of software quality.
4. Demonstrate the Manufacturing view of software quality.
5. Outline the user view of software quality.
6. Outline the value based view of user quality.

7. Summarize the Transcendental view of manufacturing quality.
8. Compare quality control and quality assurance
9. Relate SQA activities in Software development (process-oriented)
10. Relate SQA activities in software maintenance (product oriented):
11. Summarize product operation factors
12. Summarize about product revision factors.
13. Summarize about product transition factors.
14. Classify SQA system.
15. Compare testing and debugging.
16. Compare black box and white box testing.

Discriptive questions

1. Explain the three categories belonging to McCall's factor model with examples.
2. Classify the Mccall's factor model and extend its components.
3. Classify SQA system components and explain at least two major components in detail.
4. Explain in detail about five views of software quality and objectives of SQA.
5. Explain in detail about pre project quality components.
6. Explain in detail about SQA Architecture.
7. Extend the SQA system.
8. Outline the major components of SQA and explain in detail.
9. Demonstrate SQA activities in software development and software maintenance.
10. Illustrate McCall's factor model with an example.
11. Extend the objectives of quality factors based on quality category.
12. Show the architecture of SQA and explain its components.