**Software Testing Questions**

Manual Testing

### What is software?

* The programs and other operating information used by the computer.

### What is testing?

* Testing is finding out how well something works.

### What is Software Testing?

* Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product, meets the business and technical requirements that guided it’s design and development.

### Who does testing?

* Software tester
* Software Developer
* Project lead/Manager
* End user

### Why Testing is needed?

* Software testing is very important because of the following reasons: Software testing is really required to point out the defects and errors that were made during the development phases. It’s essential since it makes sure of the Customer’s reliability and their satisfaction in the application.

### Objective of Testing

* Software testing has different goals and objectives. The major objectives of Software testing are as follows:

1. Finding Defects which may get created by the programmer while developing the software.
2. Gaining confidence in and providing information about the level of quality.
3. To prevent defects.
4. To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
5. To gain the confidence of the customers by providing them a quality product.

### 7 fundamental principles of testing

1. testing shows presence of defects

* testing can show the defects are present, but cannot prove that there is no defects.

1. exhaustive testing is impossible

* testing everything including all combinations of inputs and preconditions is not possible.

1. early testing

* in the SDLC testing activities should start as early as possible and should be focused on defined objectives.

1. defect clustering

* a small number of modules contains most of the defects discovered during pre-release testing or shows the most operational failures.

1. pesticide paradox

* if the same kind of tests are repeated again and again, eventually the same set of test cases will no longer be able to find any new bugs.

1. testing is context depending

* testing is basically context dependent. Different kinds of sites are tested differently.

1. absence-of-errors fallacy

* if the system built is unusable and does not fulfil the user’s needs and expectations then finding and fixing defects does not help.

### fundamental test process.

1. planning and control
2. analysis and design
3. implementation and execution
4. evaluating exit criteria and reporting
5. test closure activities

### SDLC

* **Software development life cycle**

1. Requirement gathering and analysis
2. Design
3. Implementation or coding
4. Testing
5. Deployment
6. Maintenance

### SDLC Models

1. Waterfall model
2. V model
3. Agile model
4. Incremental model
5. RAD model
6. Iterative model
7. Spiral model

### Waterfall Model

1. In waterfall model, each phase must be completed fully before the next phase can begin.
2. This type of model is basically used for the project which is small and there are no uncertain requirements.
3. At the end of each phase, a review takes place to determine if the project is on the right path or not, to continue or discard the project.
4. In the model the testing starts only after the development is complete.
5. In waterfall model phase, do not overlap.

* Diagram of Waterfall model:



### V Model

V- model means Verification and Validation model. Just like the [**waterfall model**](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/), the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins.  Testing of the product is planned in parallel with a corresponding phase of development in **V-model**.

**Diagram of V-model:**



The various phases of the V-model are as follows:

**Requirements** like BRS and SRS begin the life cycle model just like the waterfall model. But, in this model before development is started, a [**system test**](http://istqbexamcertification.com/what-is-system-testing/) plan is created.  The test plan focuses on meeting5 the functionality specified in the requirements gathering.

**The high-level design (HLD)** phase focuses on system architecture and design. It provides overview of solution, platform, system, product and service/process. An [**integration test**](http://istqbexamcertification.com/what-is-integration-testing/) plan is created in this phase as well in order to test the pieces of the software systems ability to work together.

**The low-level design** **(LLD)** phase is where the actual software components are designed. It defines the actual logic for each and every component of the system. Class diagram with all the methods and relation between classes comes under LLD. Component tests are created in this phase as well.

**The implementation** phase is, again, where all coding takes place. Once coding is complete, the path of execution continues up the right side of the V where the test plans developed earlier are now put to use.

**Coding:** This is at the bottom of the V-Shape model. Module design is converted into code by developers.

**Advantages of V-model:**

* Simple and easy to use.
* Testing activities like planning, [**test designing**](http://istqbexamcertification.com/what-is-test-design-or-how-to-specify-test-cases/) happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
* Proactive defect tracking – that is defects are found at early stage.
* Avoids the downward flow of the defects.
* Works well for small projects where requirements are easily understood.

**Disadvantages of V-model:**

* Very rigid and least flexible.
* Software is developed during the implementation phase, so no early prototypes of the software are produced.
* If any changes happen in midway, then the test documents along with requirement documents has to be updated.

**When to use the V-model:**

* The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
* The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.

High confidence of customer is required for choosing the V-Shaped model approach. Since, no prototypes are produced, there is a very high risk involved in meeting customer expectations.

### Agile Model

**Agile development model** is also a type of [**Incremental model**](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/). Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly [**tested**](http://istqbexamcertification.com/why-is-testing-necessary/) to ensure [**software quality**](http://istqbexamcertification.com/what-is-software-quality/) is maintained. It is used for time critical applications.  Extreme Programming (XP) is currently one of the most well known agile [**development life cycle model**](http://istqbexamcertification.com/what-are-the-software-development-models/).

**Diagram of Agile model:**



Advantages of Agile model:

* Customer satisfaction by rapid, continuous delivery of useful software.
* People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
* Working software is delivered frequently (weeks rather than months).
* Face-to-face conversation is the best form of communication.
* Close, daily cooperation between business people and developers.
* Continuous attention to technical excellence and good design.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed

Disadvantages of Agile model:

* In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
* There is lack of emphasis on necessary designing and documentation.
* The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
* Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

When to use Agile model:

* When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
* To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
* Unlike the [waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/) in agile model very limited [planning](http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/) is required to get started with the project. Agile assumes that the end users’ needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.
* Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

**What is a Template & CheckList & Review Process**

Template is the structure of the document which needs to be followed, while preparing the document.

CheckList is the guidelines that need to be followed while preparing the document. Checklist makes sure that we are not missing any important points.

Using the template and checklist the deliverable will be prepared. The deliverable can be source code, document etc.

After the deliverable is prepared, it will be reviewed. All the review comments needs to be added into the deliverable and then the deliverable is finally delivered.

**Reviews are of two types**

* Internal review [informal review]: Internal review is done by the internal team member within the team.
* External review [formal review]: external review is done by the external team member outside the team.

**What is Static & Dynamic Testing**

Static Testing: The review process is called as the static testing

Dynamic Testing: The actual testing of the product is called as dynamic testing.

**What is Traceability Matrix**

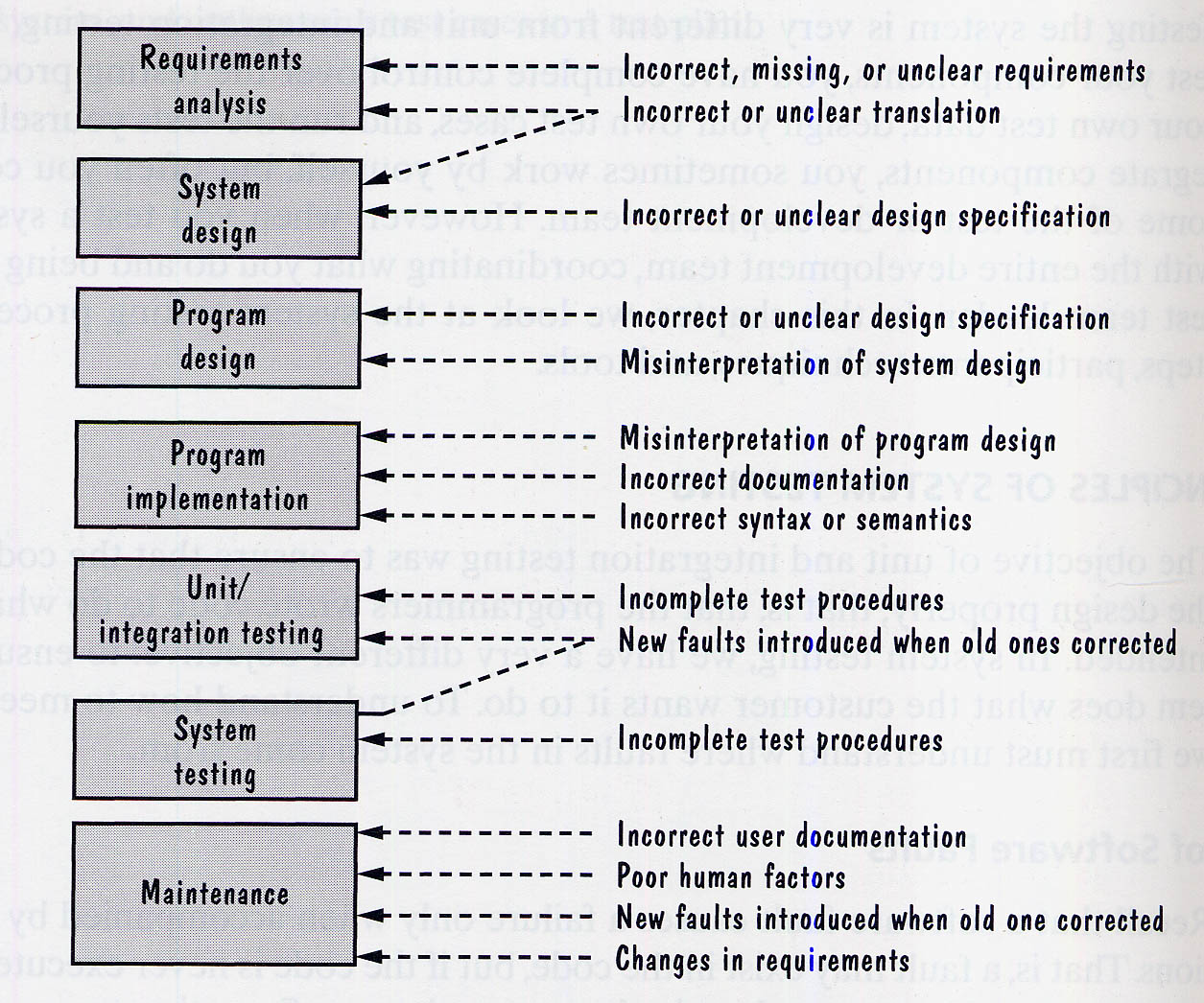
Traceability matrix is the review process which is done by the testing team to check that we have covered all the test cases as per the requirement. The review process will let us know whether we have covered all the requirements in the test cases or not.

**Overview of Templates, Checklist, Review & Usage in SDLC Deliverables**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Templates | Checklist | Review | Deliverable |
| List of items to be covered for that deliverable. Fonts, size, page no | Whether the important points that needs to be covered for that phase or task have been covered or not.  This is a reminder of the points to be covered for that deliverable.  The missing points needs to be updated. | While doing reviews also the reviewer will have a checklist of the points to be covered. | This all quality process will make this a better deliverable. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SDLC PHASES | Description | Person | Deliverable |  |  |
| Requirement | The details of what needs to be done needs to document. | Business **Analyst** | **SRS (System Requirement Specification).** | Business Analyst will meet the client and will prepare the SRS | Signed off or app  roved then only the next phase will start. |
| Design | Understand the requirement and prepare the architecture. | Architects | HLD(High Level Document)  LLD (Low Level Document) | The architect will be meeting the BA or going the SRS and prepare the deliverable. | Signed off |
| Development | Developer will understand the code and then will starting preparing. | Software Developers | Code | Developers with understand the design and requirements and prepare the code |  |
| Testing | Tester will take the code output and test the system. | Software Testers | Test Cases & Reports | Testers will do the testing | Signoff, |
| Implementation |  |  |  |  |  |

**Why Projects Fails**



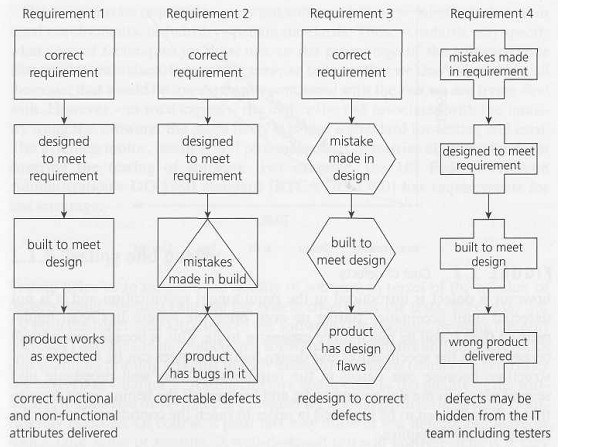
Requirement is not understood properly. Lacking of domain expertise.

**Miscommunication :**

Assumption : The requirement people will assume something.

The developer will assume something else.

The tester will assume something else



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

It helps prevent defects from being introduced into the code.

**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.

**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.

**What is the purpose of exit criteria?**

The purpose of exit criteria is to define when a test level is completed.

**What determines the level of risk?**

 The likelihood of an adverse event and the impact of the event determine the level of risk.

**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.

**What is the MAIN objective when reviewing a software deliverable?**

To identify defects in any software work product.

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

Test case specification defines the expected results of a test.

**What is the benefit of test independence?**

It avoids author bias in defining effective tests.

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

The exit criteria is determined on the bases of 'Test Planning'.

**What is beta testing?**

Testing performed by potential customers at their own locations.

**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.

**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

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

User Acceptance Test Cases

**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 depending 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 is defined as a functional system testing.

**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)

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

A 'Test Analysis' and 'Design' includes evaluation of the testability of the requirements and system.

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

Because testing everything is not feasible.

**What is random/monkey testing? When it is used?**

Random testing often known as monkey testing. In such type of testing data is generated randomly often using a tool or automated mechanism. With this randomly generated input the system is tested and results are analysed accordingly. These testing are less reliable; hence it is normally used by the beginners and to see whether the system will hold up under adverse effects.

**Which of the following are valid objectives for incident reports?**

1. Provide developers and other parties with feedback about the problem to enable identification, isolation and correction as necessary.
2. Provide ideas for test process improvement.
3. Provide a vehicle for assessing tester competence.
4. Provide testers with a means of tracking the quality of the system under test.

**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.

**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.

**What is the role of moderator in review process?**

The moderator (or review leader) leads the review process. He or she determines, 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.

**When should configuration management procedures be implemented?**

During test planning.

**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 commonly 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.

**When "Regression Testing" should be performed?**

After the software has changed or when the environment has changed Regression testing should be performed.

**What is negative and positive testing?**

A negative test is when you put in an invalid input and receives errors. While a positive testing, is when you put in a valid input and expect some action to be completed in accordance with the specification.

**What is the purpose of a test completion criterion?**

The purpose of test completion criterion is to determine when to stop testing

**What can static analysis NOT find?**

For example memory leaks.

**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.

**Could reviews or inspections be considered part of testing?**

Yes, because both help detect faults and improve quality.

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

 Driver

**When should testing be stopped**

It depends on the risks for the system being tested. There are some criteria bases on which you can stop testing.

1. Deadlines (Testing, Release)
2. Test budget has been depleted
3. Bug rate fall below certain level
4. Test cases completed with certain percentage passed
5. Alpha or beta periods for testing ends
6. Coverage of code, functionality or requirements are met to a specified point

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

Technical Review.

**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.

**Why is incremental integration preferred over "big bang" integration?**

Because incremental integration has better early defects screening and isolation ability

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

It would be before test case designing. 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 organisations do test after development started then requirements must be already traceable to their source. To make life simpler use a tool to manage requirements.

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

During test planning

**The purpose of requirement phase is**

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

**Why we split testing into distinct stages?**

We split testing into distinct stages because of following reasons,

1. Each test stage has a different purpose
2. It is easier to manage testing in stages
3. We can run different test into different environments
4. Performance and quality of the testing is improved using phased testing

**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

**What is Alpha testing?**

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

**What is a failure?**

Failure is a departure from specified behaviour.

**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**

**What is Boundary value testing**

Test boundary conditions on, below and above the edges of input and output equivalence classes. For instance, let say a bank application where you can withdraw maximum Rs.20,000 and a minimum of Rs.100, so in boundary value testing we test only the exact boundaries, rather than hitting in the middle.  That means we test above the maximum limit and below the minimum limit.

**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 creation 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 typically 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.

**What is the difference between STLC (  Software Testing Life Cycle) and SDLC ( Software Development Life  Cycle) ?**

The complete Verification and Validation of software is done in SDLC, while STLC only does Validation of the system. SDLC is a part of STLC.

**What is traceability matrix?**

The relationship between test cases and requirements is shown with the help of a document. This document is known as traceability matrix.

**What is the difference between static and dynamic testing?**

Static testing: During Static testing method, the code is not executed and it is performed using the software documentation.

Dynamic testing:  To perform this testing the code is required to be in an executable form.

**What is verification and validation?**

Verification is a process of evaluating software  at development phase and to decide whether the product of a given  application satisfies the specified requirements. Validation is the process of evaluating software at the end of the development process and to check whether it meets the customer requirements.

**What are different test levels?**

There are four test levels

1. Unit/component/program/module testing
2. Integration testing
3. System testing
4. Acceptance testing

**109. What is Integration testing**

Integration testing is a level of software testing process, where individual units of an application are combined and tested. It is usually performed after unit and functional testing.

**What are the tables in testplans?**

Test design, scope, test strategies , approach are various details that Test plan document consists of.

1. Test case identifier
2. Scope
3. Features to be tested
4. Features not to be tested
5. Test strategy & Test approach
6. Test deliverables
7. Responsibilities
8. Staffing and training
9. Risk and Contingencies

**What is the difference between UAT (User Acceptance Testing) and System testing?**

System Testing: System testing is finding defects when the system under goes testing as a whole, it is also known as end to end testing. In such type of testing, the application undergoes from beginning till the end.

UAT: User Acceptance Testing (UAT) involves running a product through a series of specific  tests  which determines whether the product will meet the needs of its users.

**Mention the difference between Data Driven Testing and Retesting?**

**Retesting:**  It is a process of checking bugs that are actioned by development team to verify that they are actually fixed.

**Data Driven Testing (DDT):**In data driven testing process, application is tested with multiple test data. Application is tested with different set of values.

**What are the valuable steps to resolve issues while testing?**

* Record : Log and handle any problems which has happened
* Report: Report the issues to higher level manager
* Control: Define the issue management process

**What is the difference between test scenarios, test cases and test script?**

Difference between test scenarios and test cases is that

**Test Scenarios:**  Test scenario is prepared before the actual testing starts, it includes plans for testing product, number of team members, environmental condition, making test cases, making test plans and all the features that are to be tested for the product.

**Test Cases:**  It is a document that contains the steps that has to be executed, it has been planned earlier.

**Test Script:** It is written in a programming language and it's a short program used to test part of functionality of the software system. In other words a written set of steps that should be performed manually.

**What is Latent defect?**

**Latent defect:** This defect is an existing defect in the system which does not cause any failure as the exact set of conditions has never been met

**Explain what is Test Deliverables**

Test Deliverables are set of documents, tools and other components that has to be developed and maintained in support of testing.

There are different test deliverables at every phase of the software development lifecycle

* Before Testing
* During Testing
* After the Testing

**What is mutation testing**

Mutation testing is a technique to identify if a set of test data or test case is useful by intentionally introducing various code changes (bugs) and retesting with original test data/ cases to determine if the bugs are detected.

**How will you conduct Risk Analysis?**

For the risk analysis following steps need to be implemented

a)      Finding the score of the risk

b)      Making a profile for the risk

c)       Changing the risk properties

d)      Deploy the resources of that test risk

e)      Making a database of risk

**122. What are the categories of debugging?**

Categories for debugging

a)      Brute force debugging

b)      Backtracking

c)       Cause elimination

d)      Program slicing

e)      Fault tree analysis

**What is fault masking explain with example?**

When presence of one defect hides the presence of another defect in the system is known as fault masking.

Example : If the "Negative Value" cause a firing of unhandled system exception,  the developer will prevent the negative values inpu. This will resolve the issue and hide the defect of unhandled exception firing.

**Explain what is Test Plan ? What are the information that should be covered in Test Plan ?**

A test plan can be defined as a document describing the scope, approach, resources and schedule of testing activities and a test plan should cover the following details.

* Test Strategy
* Test Objective
* Exit/Suspension Criteria
* Resource Planning
* Test Deliverables

**How you can eliminate the product risk in your project ?**

To eliminate product risk in your project, there is simple yet crucial step that can reduce the product risk in your project.

* Investigate the specification documents
* Have discussions about the project with all stakeholders including the developer
* As a real user walk around the website

**What are the common risk that leads to the project failure?**

The common risk that leads to a project failure are

* Not having enough human resource
* Testing Environment may not be set up properly
* Limited Budget
* Time Limitations

**On what basis you can arrive to an estimation for your project?**

To estimate your project , you have to consider following points

* Divide the whole project into a smallest tasks
* Allocate each task to team members
* Estimate the effort required to complete each task
* Validate the estimation

**Explain how you would allocate task to team members ?**

|  |  |
| --- | --- |
| **Task** | **Member** |
| * Analyze software requirement specification | * All the members |
| * Create the test specification | * Tester/Test Analyst |
| * Build up the test environment | * Test administrator |
| * Execute the test cases | * Tester, Test administrator |
| * Report defects | * Tester |

**Explain what is testing type and what are the commonly used testing type ?**

To get an expected test outcome a standard procedure is followed which is referred as Testing Type.

Commonly used testing types are

* Unit Testing:  Test the smallest code of an application
* API Testing: Testing API created for the application
* Integration Testing: Individual software modules are combined and tested
* System Testing: Complete testing of system
* Install/UnInstall Testing: Testing done from the point of client/customer view
* [Agile Testing](http://www.guru99.com/agile-scrum-extreme-testing.html): Testing through Agile technique

**While monitoring your project what all things you have to consider ?**

The things that has to be taken in considerations are

* Is you project on schedule
* Are you over budget
* Are you working towards the same career goal
* Have you got enough resources
* Are there any warning signs of impending problems
* Is there any pressure from management to complete the project sooner

**What are the common mistakes which creates issues ?**

* Matching resources to wrong projects
* Test manager lack of skills
* Not listening to others
* Poor Scheduling
* Underestimating
* Ignoring the small problems
* Not following the process

**What does a typical test report contains? What are the benefits of test reports?**

A test report contains following things:

* Project Information
* Test Objective
* Test Summary
* Defect

The benefits of test reports are:

* Current status of project and quality of product are informed
* If required, stake holder and customer can take corrective action
* A final document helps to decide whether the product is ready for release

**What is** [**test management**](http://www.guru99.com/test-management.html) **review and why it is important?**

Management review is also referred as Software Quality Assurance or SQA. SQA focusses more on the software process rather than the software work products.  It is a set of activities designed to make sure that the project manager follows the standard process.  SQA helps test manager to benchmark the project against the set standards.

**What are the best practices for software quality assurance?**

The best practices for an effective SQA implementation is

* Continuous Improvement
* Documentation
* Tool Usage
* Metrics
* Responsibility by team members
* Experienced SQA auditors

**What is difference between Test matrix and Traceability matrix?**

**Test Matrix**:  Test matrix is used to capture actual quality, effort, the plan, resources and time required to capture all phases of software testing

**Traceability Matrix**:Mapping between test cases and customer requirements is known as Traceability Matrix

**In manual testing what are stubs and drivers?**

Both stubs and drivers are part of incremental testing.  In incremental testing there are two approaches namely bottom up and top down approach. Drivers are used in bottom up testing and stub is used for top down approach. In order to test the main module, stub is used, whuich is a dummy code or program .

**Explain what is "Test Plan Driven" or "Key Word Driven" method of testing?**

This technique uses the actual test case document developed by testers using a spread sheet containing special "key Words". The key words control the processing.

**What is Fuzz testing and when it is used?**

Fuzz testing is used to detect security loopholes and coding errors in software.  In this technique random data is added to the system in attempt to crash the system.  If vulnerability persists, a tool called fuzz tester is used to determine potential causes. This technique is more useful for bigger projects but only detects major fault.

**Mention what are the main advantages of statement coverage metric of software testing?**

The benefit of statement coverage metric is that

a)      It does not require processing source code and can be applied directly to object code

b)      Bugs are distributed evenly through code, due to which percentage of executable statements covered reflects the percentage of faults discovered

**How to generate test cases for replace string method?**

a)      If characters in new string > characters in previous string.  None of the characters should get truncated

b)      If characters in new string< characters in previous string.  Junk characters should not be added

c)       Spaces after and before the string should not be deleted

d)      String should be replaced only for the first occurrence of the string

**Mention what are the categories of defects?**

Mainly there are three defect categories

* **Wrong**: When requirement is implemented incorrectly
* **Missing**: It is a variance from the specification, an indication that a specification was not implemented or a requirement of the customer is not met
* **Extra**: A requirement incorporated into the product that was not given by the end customer. It is considered as a defect because it is a variance from the existing requirements

**Explain how does a test coverage tool works?**

The code coverage testing tool runs parallel while performing testing on the actual product. The code coverage tool monitors the executed statements of the source code. When the final testing is done we get a complete report of the pending statements and also get the coverage percentage.

**Mention what is the difference between a "defect" and a "failure" in software testing?**

In simple terms when a defect reaches the end customer it is called a failure while the defect is identified internally and resolved then it is referred as defect.

**Explain how to test documents in a project that span across the software development lifecycle?**

The project span across the software development lifecycle in following manner

* Central/Project test plan: It is the main test plan that outlines the complete test strategy of the project. This plan is used till the end of the software development lifecycle
* Acceptance test plan: This document begins during the requirement phase and is completed at final delivery
* System test plan: This plan starts during the design plan and proceeds until the end of the project
* Integration and Unit test plan: Both these test plans start during the execution phase and last until the final delivery

**Explain which test cases are written first black boxes or white boxes?**

Black box test cases are written first as to write black box test cases; it requires project plan and requirement document all these documents are easily available at the beginning of the project. While writing white box test cases requires more architectural understanding and is not available at the start of the project.

**Explain what is the difference between latent and masked defects?**

* **Latent defect:** A latent defect is an existing defect that has not caused a failure because the sets of conditions were never met
* **Masked defect:** It is an existing defect that has not caused a failure because another defect has prevented that part of the code from being executed

**Mention what are the different types of test coverage techniques?**

Different types of test coverage techniques include

* **Statement Coverage:** It verifies that each line of source code has been executed and tested
* **Decision Coverage:** It ensures that every decision in the source code is executed and tested
* **Path Coverage:** It ensures that every possible route through a given part of code is executed and tested

**Mention what is the meaning of breadth testing?**

Breadth testing is a test suite that exercises the full functionality of a product but does not test features in detail

**Mention what is the difference between Pilot and Beta testing?**

The difference between pilot and beta testing is that pilot testing is actually done using the product by the group of user before the final deployment and in beta testing we do not input real data, but it is installed at the end customer to validate if the product can be used in production.

**Mention what are the basic components of defect report format?**

The basic components of defect report format includes

* Project Name
* Module Name
* Defect detected on
* Defect detected by
* Defect ID and Name
* Snapshot of the defect
* Priority and Severity status
* Defect resolved by
* Defect resolved on

**Mention what is the purpose behind doing end-to-end testing?**

End-to end testing is done after functional testing. The purpose behind doing end-to-end testing is that

* To validate the software requirements and integration with external interfaces
* Testing application in real world environment scenario
* Testing of interaction between application and database