**MANUAL TESTING FAQ’s**

**What is bidirectional traceability?**

**Bidirectional traceability needs to be implemented both forward and backward (i.e., from requirements to end products and from end product back to requirements).   
When the requirements are managed well, traceability can be established from the source requirement to its lower level requirements and from the lower level requirements back to their source. Such bidirectional traceability helps determine that all source requirements have been completely addressed and that all lower level requirements can be traced to a valid source.**

**What is stub? Explain in testing point of view?**

**Stub is a dummy program or component, the code is not ready for testing, it's used for testing...that means, in a project if there are 4 modules and last is remaining and there is no time then we will use dummy program to complete that fourth module and we will run whole 4 modules also. The dummy program is also known as stub.**

**For Web Applications what type of tests are you going to do?**

**Web-based applications present new challenges, these challenges include:  
- Short release cycles;  
- Constantly Changing Technology;  
- Possible huge number of users during initial website launch;  
- Inability to control the user's running environment;  
- 24-hour availability of the web site.  
  
  
The quality of a website must be evident from the Onset. Any difficulty whether in response time, accuracy of information, or ease of use-will compel the user to click to a competitor's site. Such problems translate into lost of users, lost sales, and poor company image.  
  
  
To overcome these types of problems, use the following techniques:   
1. Functionality Testing  
Functionality testing involves making sure the features that most affect user interactions work properly. These include:   
· forms  
· searches  
· pop-up windows  
· shopping carts  
· online payments**

**2. Usability testing  
many users has low tolerance for anything that is difficult to use or that does not work. A user's first impression of the site is important, and many websites have become cluttered with an increasing number of features. For general-use websites frustrated users can easily click over a competitor's site.  
  
Usability testing involves following main steps  
· identify the website's purpose;   
· identify the indented users;   
· define tests and conduct the usability testing  
· analyze the acquired information  
  
3. Navigation Testing  
Good Navigation is an essential part of a website, especially those that are complex and provide a lot of information. Assessing navigation is a major part of usability testing.  
  
4. Forms Testing  
Websites that use forms need tests to ensure that each field works properly and that the forms posts all data as intended by the designer.  
  
5. Page Content Testing  
Each web page must be tested for correct content from the user perspective for correct content from the user perspective. These tests fall into two categories: ensuring that each component functions correctly and ensuring that the content of each is correct.  
  
6. Configuration and Compatibility testing  
A key challenge for web applications is ensuring that the user sees a web page as the designer intended. The user can select different browser software and browser options, use different network software and on-line service, and run other concurrent applications. We execute the application under every browser/platform combination to ensure the web sites work properly under various environments.  
  
7. Reliability and Availability testing  
A key requirement o a website is that it be available whenever the user requests it, after 24-hours a day, every day. The number of users accessing web site simultaneously may also affect the site's availability.  
  
8. Performance Testing  
Performance Testing, which evaluates System performance under normal and heavy usage, is crucial to success of any web application. A system that takes for long to respond may frustrate the user who can then quickly move to a competitor's site. Given enough time, every page request will eventually be delivered. Performance testing seeks to ensure that the website server responds to browser requests within defined parameters.  
  
9. Load testing  
the purpose of Load testing is to model real world experiences, typically by generating many simultaneous users accessing the website. We use automation tools to increases the ability to conduct a valid load test, because it emulates thousands of users by sending simultaneous requests to the application or the server.  
  
10. Stress Testing  
Stress Testing consists of subjecting the system to varying and maximum loads to evaluate the resulting performance. We use automated test tools to simulate loads on website and execute the tests continuously for several hours or days.  
  
11. Security Testing  
Security is a primary concern when communicating and conducting business- especially sensitive and business- critical transactions - over the internet. The user wants assurance that personal and financial information is secure. Finding the vulnerabilities in an application that would grant an unauthorized user access to the system is important.**

**Define Brain Stromming and Cause Effect Graphing?**

**BS:  
A learning technique involving open group discussion intended to expand the range of available ideas  
OR  
A meeting to generate creative ideas. At PEPSI Advertising, daily, weekly and bi-monthly brainstorming sessions are held by various work groups within the firm. Our monthly I-Power brainstorming meeting is attended by the entire agency staff.  
OR  
Brainstorming is a highly structured process to help generate ideas. It is based on the principle that you cannot generate and evaluate ideas at the same time. To use brainstorming, you must first gain agreement from the group to try brainstorming for a fixed interval (e.g. six minutes).   
  
CEG:  
A testing technique that aids in selecting, in a systematic way, a high-yield set of test cases that logically relates causes to effects to produce test cases. It has a beneficial side effect in pointing out incompleteness and ambiguities in specifications.**

**What is the maximum length of the test case we can write?**

**We can't say exactly test case length, it depending on functionality.**

**Password is having 6 digits alphanumeric then what are the possible input conditions?**

**Including special characters also Possible input conditions are:  
1) Input password as = 6abcde (i.e. number first)  
2) Input password as = abcde8 (i.e. character first)  
3) Input password as = 123456 (all numbers)  
4) Input password as = abcdef (all characters)  
5) Input password less than 6 digit  
6) Input password greater than 6 digits  
7) Input password as special characters  
8) Input password in CAPITAL i.e. uppercase  
9) Input password including space  
10) (SPACE) followed by alphabets /numerical /alphanumerical/**

**What is internationalization testing?**

**Software Internationalization is process of developing software products independent from cultural norms, language or other specific attributes of a market**

**If I give some thousand tests to execute in 2 days what do you do?**

**If possible, we will automate or else, execute only the test cases which are mandatory.**

**What does black-box testing mean at the unit, integration, and system levels?**

**Tests for each software requirement using Equivalence Class Partitioning, Boundary Value Testing, and more Test cases for system software requirements using the Trace Matrix, Cross-functional Testing, Decision Tables, and more Test cases for system integration for configurations, manual operations, etc**

**What is agile testing?**

**Agile testing is used whenever customer requirements are changing dynamically  
  
If we have no SRS, BRS but we have test cases does you execute the test cases blindly or do you follow any other process.  
  
Test case would have detail steps of what the application is supposed to do.   
1) Functionality of application.   
2) In addition you can refer to Backend, is mean look into the Database. To gain more knowledge of the application.**

**What is Bug life cycle?**

**New: when tester reports a defect  
Open: when developer accepts that it is a bug or if the developer rejects the defect, then the status is turned into "Rejected"  
Fixed: when developer make changes to the code to rectify the bug...  
Closed/Reopen: when tester tests it again. If the expected result shown up, it is turned into "Closed" and if the problem persists again, it's "Reopen".**

**What is deferred status in defect life cycle?**

**Deferred status means the developer accepted the bus, but it is scheduled to rectify in the next build.**

**Smoke test? Do you use any automation tool for smoke testing?**

**Testing the application whether it is performing its basic functionality properly or not there by the test team can go ahead with the application. Definitely can use.**

**Verification and validation?**

**Verification is static. No code is executed. Say, analysis of requirements etc.  
Validation is dynamic. Code is executed with scenarios present in test cases.**

**When a bug is found, what is the first action?**

**Report it in bug tracking tool.**

**What is test plan and explain its contents?**

**Test plan is a document which contains the scope for testing the application and what to be tested, when to be tested and who to test.**

**Advantages of automation compared to manual testing?**

**Time saving, resource and money**

**What is mean by release notes?**

**It's a document released along with the product which explains about the product. It also contains about the bugs that are in deferred status.**

**What is Testing environment in your company, means how testing process start?**

**Testing process is going as follows:  
Quality assurance unit  
Quality assurance manager  
Test lead  
Test engineer**

**Give an example of high priority and low severity, low priority and high severity?**

**Severity level:**

**The degree of impact the issue or problem has on the project. Severity 1 usually means the highest level requiring immediate attention. Severity 5 usually represents a documentation defect of minimal impact.**

**Severity is levels:**

* **Critical: the software will not run**
* **High: unexpected fatal errors (includes crashes and data corruption)**
* **Medium: a feature is malfunctioning**
* **Low: a cosmetic issue**

**Severity levels**

1. **Bug causes system crash or data loss.**
2. **Bug causes major functionality or other severe problems; product crashes in obscure cases.**
3. **Bug causes minor functionality problems, may affect "fit and finish".**
4. **Bug contains typos, unclear wording or error messages in low visibility fields.**

**Severity levels**

* **High: A major issue where a large piece of functionality or major system component is completely broken. There is no workaround and testing cannot continue.**
* **Medium: A major issue where a large piece of functionality or major system component is not working properly. There is a workaround, however, and testing can continue.**
* **Low: A minor issue that imposes some loss of functionality, but for which there is an acceptable and easily reproducible workaround. Testing can proceed without interruption.**

**Severity and Priority**

**Priority is Relative: the priority might change over time. Perhaps a bug initially deemed P1 becomes rated as P2 or even a P3 as the schedule draws closer to the release and as the test team finds even more heinous errors. Priority is a subjective evaluation of how important an issue is, given other tasks in the queue and the current schedule. It’s relative. It shifts over time. And it’s a business decision.**

**Severity is an absolute: it’s an assessment of the impact of the bug without regard to other work in the queue or the current schedule. The only reason severity should change is if we have new information that causes us to re-evaluate our assessment. If it was a high severity issue when I entered it, it’s still a high severity issue when it’s deferred to the next release. The severity hasn’t changed just because we’ve run out of time. The priority changed.**

**Severity Levels can be defined as follow:**

**S1 - Urgent/Showstopper. Like system crash or error message forcing to close the window.  
Tester's ability to operate the system either totally (System Down), or almost totally, affected. A major area of the users system is affected by the incident and it is significant to business processes.  
  
S2 - Medium/Workaround. Exist like when a problem is required in the specs but tester can go on with testing. Incident affects an area of functionality but there is a work-around which negates impact to business process. This is a problem that:   
a) affects a more isolated piece of functionality.  
b) Occurs only at certain boundary conditions.  
c) Has a workaround (where "don't do that" might be an acceptable answer to the user).  
d) Occurs only at one or two customers. Or is intermittent  
  
S3 - Low. This is for minor problems, such as failures at extreme boundary conditions that are unlikely to occur in normal use, or minor errors in  
layout/formatting. Problems do not impact use of the product in any substantive way. These are incidents that are cosmetic in nature and of no or very low impact to business processes.**

**What is Use case?**

**An user action and sub sequent reaction from the system in terms of pre conditions, post conditions, normal flows and exceptions. It is done by Team Lead/Test Lead/Tester.**

**Diff. between STLC and SDLC?**

**STLC is software test life cycle it starts with**

* **Preparing the test strategy.**
* **Preparing the test plan.**
* **Creating the test environment.**
* **Writing the test cases.**
* **Creating test scripts.**
* **Executing the test scripts.**
* **Analyzing the results and reporting the bugs.**
* **Doing regression testing.**
* **Test exiting.**

**SDLC is software or system development life cycle, phases are...**

* **Project initiation.**
* **Requirement gathering and documenting.**
* **Designing.**
* **Coding and unit testing.**
* **Integration testing.**
* **System testing.**
* **Installation and acceptance testing. “Support or maintenance.**

**How you are breaking down the project among team members?**

**It can be depend on these following cases----  
1) Number of modules  
2) Number of team members  
3) Complexity of the Project  
4) Time Duration of the project  
5) Team member's experience etc......**

**What is Test Data Collection?**

**Test data is the collection of input data taken for testing the application. Various types and size of input data will be taken for testing the applications. Sometimes in critical application the test data collection will be given by the client also.**

**What is Test Server?**

**The place where the developers put their development modules, which are accessed by the testers to test the functionality**

**What are non-functional requirements?**

**The non-functional requirements of a software product are: reliability, usability, efficiency, delivery time, software development environment, security requirements, standards to be followed etc.**

**What are the differences between these three words Error, Defect and Bug?**

**Error: The deviation from the required logic, syntax or standards/ethics is called as error.  
  
There are three types of error. They are:  
Syntax error (This is due to deviation from the syntax of the language what supposed to follow).   
Logical error (This is due to deviation from the logic of the program what supposed to follow)   
Execu\tion error (This is generally happens when you are executing the same program, that time you get it.)   
Defect: When an error found by the test engineer (testing department) then it is called defect  
  
Bug: if the defect is agreed by the developer then it converts into bug, which has to fix by the developer or post pond to next version.**

**Why we perform stress-testing, resolution-testing and cross- browser testing?**

**Stress Testing: - We need to check the performance of the application.   
Def: Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements   
Resolution Testing: - Sometimes developer created only for 1024 resolution, the same page displayed a horizontal scroll bar in 800 x 600 resolutions. Nobody can like the horizontal scroll appears in the screen. That is reason to test the Resolution testing.  
  
Cross-browser Testing: - This testing sometimes called compatibility testing. When we develop the pages in IE compatible, the same page is not working in Firefox or Netscape properly, because   
most of the scripts are not supporting two other than IE. So that we need to test the cross-browser Testing**

**There are two sand clocks(timers) one complete totally in 7 minutes and other in 9-minutes we have to calculate with this timers and bang the bell after completion of 11- minutes !plz give me the solution.**

**1. Start both clocks   
2. When 7 min clock complete, turn it so that it restarts.  
3. When 9 min clock finish, turn 7 min clocks (It has 2 mints only).  
4. When 7 min clock finishes, 11 min complete.**

**What are the minimum criteria for white box?**

**We should know the logic, code and the structure of the program or function. Internal knowledge of the application how the system works what's the logic behind it and structure how it should react to particular action**

**What are the technical reviews?**

**For each document, it should be reviewed. Technical Review in the sense, for each screen, developer will write a Technical Specification. It should be reviewed by developer and tester. There are functional specification review, unit test case review and code review etc.**

**n what basis you will write test cases?**

**I would write the Test cases based on Functional Specifications and BRDs and some more test cases using the Domain knowledge.**

**Explain ETVX concept?**

**E- Entry Criteria  
T- Task  
V- Validation  
X- Exit Criteria  
  
ENTRY CRITERIA: Input with 'condition' attached.  
e.g. Approved SRS document is the entry criteria for the design phase.  
  
TASK: Procedures.  
e.g. Preparation of HLD, LLD etc.  
  
VALIDATION: Building quality & Verification activities  
e.g. Technical reviews  
  
EXIT CRITERIA: Output with 'condition' attached.  
e.g. Approved design document  
It is important to follow ETVX concept for all phases in SDLC**

**What are the main key components in Web applications and client and Server applications? (Differences)**

**For Web Applications: Web application can be implemented using any kind of technology like Java, .NET, VB, ASP, CGI& PERL. Based on the technology, we can derive the components.   
  
Let's take Java Web Application. It can be implemented in 3 tier architecture. Presentation tier (jsp, html, dthml, servlets, struts). Business Tier (Java Beans, EJB, JMS) Data Tier(Databases like Oracle, SQL Server etc., )  
  
If you take .NET Application, Presentation (ASP, HTML, DHTML), Business Tier (DLL) & Data Tier ( Database like Oracle, SQL Server etc.,)  
  
Client Server Applications: It will have only 2 tiers. One is Presentation (Java, Swing) and Data Tier (Oracle, SQL Server). If it is client Server architecture, the entire application has to be installed on the client machine. Whenever you do any changes in your code, Again, It has to be installed on all the client machines. Where as in Web Applications, Core Application will reside on the server and client can be thin Client (browser). Whatever the changes you do, you have to install the application in the server. NO need to worry about the clients. Because, You will not install anything on the client machine.**

**If the client identified some bugs to whom did he reported?**

**He will report to the Project Manager. Project Manager will arrange a meeting with all the leads (Dev. Manager, Test Lead and Requirement Manager) then raise a Change Request and then, identify which all the screens are going to be impacted by the bug. They will take the code and correct it and send it to the Testing Team.**

**What is the formal technical review?**

**Technical review should be done by the team of members. The document, which is going to be reviewed, who has prepared and reviewers should sit together and do the review of that document. It is called Peer Review. If it is a technical document, It can be called as formal Technical review, I guess. It varies depends on the company policy.**

**At what phase tester role starts?**

**In SDLC after completion of FRS document the test lead prepare the use case document and test plan document, then the tester role is start.**

**Explain 'Software metrics'?**

**Measurement is fundamental to any engineering discipline  
Why Metrics?  
- We cannot control what we cannot measure!  
- Metrics helps to measure quality  
- Serves as dash-board  
  
the main metrics are: size, shedule, defects.In this there are main sub metrics.  
  
Test Coverage = Number of units (KLOC/FP) tested / total size of the system  
Test cost (in %) = Cost of testing / total cost \*100  
Cost to locate defect = Cost of testing / the number of defects located  
Defects detected in testing (in %) = Defects detected in testing / total system defects\*100  
Acceptance criteria tested = Acceptance criteria tested / total acceptance criteria**

**Actually how many positive and negative test cases will write for a module?**

**That depends on the module and complexity of logic. For every test case, we can identify +ve and -ve points. Based on the criteria, we will write the test cases, If it is crucial process or screen. We should check the screen, in all the boundary conditions.**

**What is Software reliability?**

**It is the probability that software will work without failure for a specified period of time in a specified environment. Reliability of software is measured in terms of Mean Time Between Failure (MTBF). For e.g. if MTBF = 10000 hours for an average software, then it should not fail for 10000 hours of continuous operation.**

**What are the main bugs which were identified by you and in that how many are considered as real bugs?**

**If you take one screen, let's say, it has got 50 Test conditions, out of which, I have identified 5 defects which are failed. I should give the description defect, severity and defect classification. All the defects will be considered.  
  
Defect Classification are:  
GRP : Graphical Representation  
LOG : Logical Error  
DSN : Design Error  
STD : Standard Error  
TST : Wrong Test case  
TYP : Typographical Error (Cosmo tic Error)**

**What the main use of preparing a traceability matrix?**

**Traceability matrix is prepared in order to cross check the test cases designed against each requirement, hence giving an opportunity to verify that all the requirements are covered in testing the application.  
(Or)  
To Cross verify the prepared test cases and test scripts with user requirements. To monitor the changes, enhance occurred during the development of the project.**

**What is six sigma? Explain.**

**Six Sigma:  
A quality discipline that focuses on product and service excellence to create a culture that demands perfection on target, every time.   
  
Six Sigma quality levels  
Produces 99.9997% accuracy, with only 3.4 defects per million opportunities.  
  
Six Sigma is designed to dramatically upgrade a company's performance, improving quality and productivity. Using existing products, processes, and service standards,  
They go for Six Sigma MAIC methodology to upgrade performance.   
  
  
MAIC is defined as follows:  
Measure: Gather the right data to accurately assess a problem.  
Analyze: Use statistical tools to correctly identify the root causes of a problem  
Improve: Correct the problem (not the symptom).  
Control: Put a plan in place to make sure problems stay fixed and sustain the gains.  
  
Key Roles and Responsibilities:  
  
The key roles in all Six Sigma efforts are as follows:  
Sponsor: Business executive leading the organization.  
Champion: Responsible for Six Sigma strategy, deployment, and vision.  
Process Owner: Owner of the process, product, or service being improved responsible for long-term sustainable gains.  
Master Black Belts: Coach Black belts expert in all statistical tools.  
Black Belts: Work on 3 to 5 $250,000-per-year projects; create $1 million per year in value.  
Green Belts: Work with black belt on projects.**

**What is TRM?**

**TRM means Test Responsibility Matrix.   
  
TRM: --- It indicates mapping between test factors and development stages...   
  
Test factors like:   
Ease of use, reliability, portability, authorization, access control, audit trail, ease of operates, maintainable... Like dat...   
Development stages...   
Requirement gathering, Analysis, design, coding, testing, and maintenance**

**What are cookies? Tell me the advantage and disadvantage of cookies?**

**Cookies are messages that web servers pass to your web browser when you visit Internet sites. Your browser stores each message in a small file. When you request another page from the server, your browser sends the cookie back to the server. These files typically contain information about your visit to the web page, as well as any information you've volunteered, such as your name and interests. Cookies are most commonly used to track web site activity. When you visit some sites, the server gives you a cookie that acts as your identification card. Upon each return visit to that site, your browser passes that cookie back to the server. In this way, a web server can gather information about which web pages are used the most, and which pages are gathering the most repeat hits. Only the web site that creates the cookie can read it. Additionally, web servers can only use information that you provide or choices that you make while visiting the web site as content in cookies. Accepting a cookie does not give a server access to your computer or any of your personal information. Servers can only read cookies that they have set, so other servers do not have access to your information. Also, it is not possible to execute code from a cookie, and not possible to use a cookie to deliver a virus.**

**What is the difference between Product-based Company and Projects-based Company?**

**Product based company develops the applications for Global clients i.e. there is no specific clients. Here requirements are gathered from market and analyzed with experts.   
Project based company develops the applications for the specific client. The requirements are gathered from the client and analyzed with the client.**

**http://www.klariti.com/**

**Principles of Testing**

**There are seven principles of** [**testing**](http://istqbexamcertification.com/what-is-a-software-testing/)**. They are as follows:**

**1) Testing shows presence of defects: Testing can show the** [**defects**](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/) **are present, but cannot prove that there are no defects. Even after testing the application or product thoroughly we cannot say that the product is 100% defect free. Testing always reduces the number of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.**

**2) Exhaustive testing is impossible: Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.**

**3) Early testing: In the** [**software development life cycle**](http://istqbexamcertification.com/what-are-the-software-development-life-cycle-phases/) **testing activities should start as early as possible and should be focused on defined objectives.**

**4) Defect clustering: A small number of modules contains most of the defects discovered during pre-release testing or shows the most operational failures.**

**5) Pesticide paradox: If the same kinds of tests are repeated again and again, eventually the same set of test cases will no longer be able to find any new bugs. To overcome this “Pesticide Paradox”, it is really very important to review the test cases regularly and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.**

**6) Testing is context depending: Testing is basically context dependent. Different kinds of sites are tested differently. For example, safety – critical software is tested differently from an e-commerce site.**

**7) Absence – of – errors fallacy: If the system built is unusable and does not fulfill the user’s needs and expectations then finding and fixing defects does not help.**

# System Testing: What? Why? & How?

##### ****What is System Testing?****

**System testing is the testing of behavior of a complete and fully integrated software product based on the software requirements specification (SRS) document. In main focus of this testing is to evaluate Business / Functional / End-user requirements.**

**This is black box type of testing where external working of the software is evaluated with the help of requirement documents & it is totally based on Users point of view. For this type of testing do not required knowledge of internal design or structure or code.**

**This testing is to be carried out only after System Integration Testing is completed where both**[**Functional & Non-Functional requirements**](http://www.softwaretestingclass.com/)**are verified.**

**In the integration testing testers are concentrated on finding bugs/defects on integrated modules. But in the *Software System Testing* testers are concentrated on finding bugs/defects based on software application behavior, software design and expectation of end user.**

##### ****Why system testing is important:****

**a) In Software Development Life Cycle the System Testing is perform as the first level of testing where the System is tested as a whole.**

**b) In this step of testing check if system meets functional requirement or not.**

**c) System Testing enables you to test, validate and verify both the Application Architecture and Business requirements.**

**d) The application/System is tested in an environment that particularly resembles the effective production environment where the application/software will be lastly deployed.**

**Generally, a separate and dedicated team is responsible for system testing. And,System Testing is performed on staging server which is similar to production server. So this means you are testing software application as good as production environment**

|  |  |
| --- | --- |
| **Regression Testing** | **Retesting** |
| **Regression testing is a type of software testing that intends to ensure that changes like defect fixes or enhancements to the module or application have not affecting unchanged part.** | **Retesting is done to make sure that the tests cases which failed in last execution are passing after the defects against those failures are fixed.** |
| **Regression testing is not carried out on specific defect fixes. It is planned as specific area or full regression testing.** | **Retesting is carried out based on the defect fixes.** |
| **In Regression testing, you can include the test cases which passed earlier. We can say that check the functionality which was working earlier.** | **In Retesting, you can include the test cases which failed earlier. We can say that check the functionality which was failed in earlier build.** |
| **Regression test cases we use are derived from the functional specification, the user manuals, user tutorials, and defect reports in relation to corrected problems.** | **Test cases for Retesting cannot be prepared before start testing.** |
| **Automation is the key for regression testing. Manual regression testing tends to get more expensive with each new release. Regression testing is right time to start automating test cases.** | **You cannot automate the test cases for Retesting.** |
| **Defect verification is not comes under Regression testing.** | **Defect verification is comes under Retesting.** |
| **Based on the availability of resources the Regression testing can be carried out parallel with Retesting.** | **Priority of Retesting over Regression testing is higher, so it is carried out before regression testing.** |

### What is a Traceability Matrix?

**The focus of any testing engagement is and should be maximum test coverage. By coverage, it simply means that we need to test everything there is to be tested. The aim of any testing project should be 100% test coverage.**

**Requirements Traceability Matrix to begin with, establishes a way to make sure we place checks on the coverage aspect.  It helps in creating a snap shot to identify coverage gaps.**

### How to Create a Traceability Matrix?

**To being with we need to know exactly what is it that needs to be tracked or traced.**

**Testers start writing their**[**test scenarios**](http://www.softwaretestinghelp.com/sample-test-cases-testing-web-desktop-applications/)**/objectives and eventually the test cases based on some input documents – Business requirements document,**[**Functional Specifications document**](http://www.softwaretestinghelp.com/how-to-test-software-requirements-specification-srs/)**and Technical design document (optional).**

### Important Points to Note About Traceability Matrix

**The following are the important points to note about this version of the Traceability Matrix:**

**1) The execution status is also displayed.  During execution, it gives a consolidated snapshot of how work is progressing.**

**2) Defects: When this column is used to establish the backward traceability we can tell that the “New user” functionality is the most flawed. Instead of reporting that so and so test cases failed, TM provides a transparency back to the business requirement that has most defects thus show casing the Quality in terms of what the client desires.**

**3) As a further step, you can color code the defect ID to represent their states. For example, defect ID in red can mean it is still Open, in green can mean it is closed. When this is done, the TM works as a health check report displaying the status of the defects corresponding to a certain BRD or FSD functionality is being open or closed.**

**4) If there is a technical design document or use cases or any other artifacts that you would like to track you can always expand the above created document to suit your needs by adding additional columns.**

**To sum up, a requirements traceability Matrix helps in:**

1. **Ensuring 100% test coverage**
2. **Showing requirement/document inconsistencies**
3. **Displaying the overall defect/execution status with focus on business requirements.**
4. **If a certain business and/or functional requirement were to change, a TM helps estimate or analyze the impact on the QA team’s work in terms of revisiting/reworking on the test cases.**

***Additionally,***

1. **A TM is not a manual testing specific tool, it can be used for automation projects as well. For an automation project, the test case ID can indicate the automation test script name.**
2. **It is also not a tool that can be used just by the QAs. The development team can use the same to map BRD/FSD requirements to blocks/units/conditions of code created to make sure all the requirements are developed.**
3. [**Test management tools**](http://www.softwaretestinghelp.com/category/test-management-tools/)**like**[**HP ALM**](http://www.softwaretestinghelp.com/learn-hp-quality-center-qc-in-4-days/)**come with the inbuilt traceability feature.**

**An important point to note is that, the way you maintain and update your**[**Traceability Matrix**](http://www.softwaretestinghelp.com/requirements-traceability-matrix/)**determines the effectiveness of its use. If not updated often or updated incorrectly the tool is a burden instead of being a help and creates the impression that the tool by itself it not worthy of using.**

## Release Notes - Table of Contents TOC

**1. Introduction   
1.1. Changes to the release notes  
1.2. Scope   
  
2. System Requirements  
2.1. Operating systems supported  
2.2. System requirements  
  
3. New Features  
  
4. Dropped Features  
  
5. Fixed Issues  
5.1. Included Hotfixes  
5.2. Included maintenance fixes  
  
6. Known Issues, Limitations & Restrictions  
6.1. Known Issues   
6.2. Limitations  
6.3. Restrictions  
  
7. Caveats  
  
8. Deployment Instructions  
8.1. <Item 1> – Deployment Instructions  
8.2. <Item 2> – Deployment Instructions  
8.3. <Item 3> – Deployment Instructions  
  
9. Related Documentation   
9.1. Obtaining Documentation  
  
  
How to use this Release Notes template**

**1. How to use this template  
1.1. How to add your logo to the cover page   
1.2. How to change [COMPANY] to your company name   
1.3. How to add your logo to the header   
1.4. Updating the Table of Contents   
1.5. Adding Notes   
1.6. Adding Warnings   
1.7. Adding Comments   
1.8. Bullet Lists   
1.9. Numbered List   
1.10. Sample Tables**

**Finally, *summary* of STLC along with Entry and Exit Criteria**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STLC Stage** | **Entry Criteria** | **Activity** | **Exit Criteria** | **Deliverables** |
| **Test Planning** | **Requirements Documents**  **Requirement Traceability matrix.**  **Test automation feasibility document.** | **Analyze various testing approaches available**  **Finalize on the best suited approach**  **Preparation of test plan/strategy document for various types of testing**  **Test tool selection**  **Test effort estimation**  **Resource planning and determining roles and responsibilities.** | **Approved test plan/strategy document.**  **Effort estimation document signed off.** | **Test plan/strategy document.**  **Effort estimation document.** |
| **Requirement Analysis** | **Requirements Document available (both functional and non functional)**  **Acceptance criteria defined.**  **Application architectural document available.** | **Analyse business functionality to know the business modules and module specific functionalities.**  **Identify all transactions in the modules. Identify all the user profiles.**  **Gather user interface/authentication, geographic spread requirements.**  **Identify types of tests to be performed.**  **Gather details about testing priorities and focus.**  **Prepare Requirement Traceability Matrix (RTM).**  **Identify test environment details where testing is supposed to be carried out.**  **Automation feasibility analysis (if required).** | **Signed off RTM**  **Test automation feasibility report signed off by the client** | **RTM**  **Automation feasibility report (if applicable)** |
|  |  |  |  |  |
| **Test case development** | **Requirements Documents**  **RTM and test plan**  **Automation analysis report** | **Create test cases, automation scripts (where applicable)**  **Review and baseline test cases and scripts**  **Create test data** | **Reviewed and signed test Cases/scripts**  **Reviewed and signed test data** | **Test cases/scripts**  **Test data** |
| **Test Environment setup** | **System Design and architecture documents are available**  **Environment set-up plan is available** | **Understand the required architecture, environment set-up**  **Prepare hardware and software requirement list**  **Finalize connectivity requirements**  **Prepare environment setup checklist**  **Setup test Environment and test data**  **Perform smoke test on the build**  **Accept/reject the build depending on smoke test result** | **Environment setup is working as per the plan and checklist**  **Test data setup is complete**  **Smoke test is successful** | **Environment ready with test data set up**  **Smoke Test Results.** |
| **Test Execution** | **Baselined RTM, Test Plan , Test case/scripts are available**  **Test environment is ready**  **Test data set up is done**  **Unit/Integration test report for the build to be tested is available** | **Execute tests as per plan**  **Document test results, and log defects for failed cases**  **Update test plans/test cases, if necessary**  **Map defects to test cases in RTM**  **Retest the defect fixes**  **Regression testing of application**  **Track the defects to closure** | **All tests planned are executed**  **Defects logged and tracked to closure** | **Completed RTM with execution status**  **Test cases updated with results**  **Defect reports** |
| **Test Cycle closure** | **Testing has been completed**  **Test results are available**  **Defect logs are available** | **Evaluate cycle completion criteria based on - Time, Test coverage , Cost , Software Quality , Critical Business Objectives**  **Prepare test metrics based on the above parameters.**  **Document the learning out of the project**  **Prepare Test closure report**  **Qualitative and quantitative reporting of quality of the work product to the customer.**  **Test result analysis to find out the defect distribution by type and severity** | **Test Closure report signed off by client** | **Test Closure report**  **Test metrics** |