

1. Can you tell me about yourself?

Answer: In my QA career, I have been working on various system platforms and operating systems like (example Windows 8.x, Windows 7, Windows XP and UNIX. I have tested applications developed in Java, C++, Visual Basic and so on, I have tested Web-based applications as well as client server applications.

As a QA person, I have written Test Plans, Test Cases, attended walkthrough meetings with the Business Analysts, Project Managers, Business Managers and QA Leads. Attended requirement review meetings and provided feedback to the Business Analysts. I have worked in different databases like Oracle and DB2, wrote SQL queries to retrieve data from the database.

As far as different types of testing is concerned, I have performed Smoke Testing, Functional Testing, Backend Testing, BlackBox Testing, Integration Testing, Regression Testing and UAT (User Acceptance Testing) Testing. I have participated in Load Testing and Stress Testing.

I have written defects as they are found using ClearQuest and TestDirector. Once the defects were fixed, retested them and if the passed, closed them. If the defects were not fixed, then reopened them. I have also attended the defect assessment meetings as necessary.

In the meantime, a continuous interaction with developers was necessary.

Tell some of your past relevant experience to business function of prospective employer and how your are one of the best suited candidate.

2. What did you do in your last project?

In my last project, the application was a web-based application developed in Java platform. As a QA Person, I wrote Test Plans from the requirement documents and Use Cases. I performed Smoke Testing, Functional Testing, Backend Testing, BlackBox Testing, Integration Testing, Regression Testing and UAT (User Acceptance Testing). I have participated in Load Testing and Stress Testing. I attended several walkthrough meetings for requirement reviews and provided feedback to the Business Analysts. Mostly, I was in the backend testing, which required writing SQL queries directly to the database.

Besides these, I wrote defects using ClearQuest. Once the defects were fixed, retested them and if the passed, closed them. If the defects were not fixed, then reopened them.

3. Have you written Test Plan? What is a Test Plan? What does it include?

Yes.

What is a Test Plan?

A Test Plan is a document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks and who will do each task (roles and responsibilities) and any risks and its solutions.

What does it include? A Test Plan includes Heading, Revision History, Table of Contents, Introduction, Scope, Approach, Overview, different types of testing that will be carried out, what software and hardware will be required, issues, risks, assumptions and sign off section.

Table of Contents	1
2. SCOPE AND OBJECTIVES	3
2.1. Scope of Test Approach - System Functions	3
2.1.1. INCLUSIONS	3
2.1.2. EXCLUSIONS	4
2.2. Testing Process	4
2.3. Testing Scope (Test Types that will be performed)	5
2.3.1. Functional Testing	5
2.3.2. Integration Testing	5
2.3.3 Beta Testing	5
2.3.4. User Acceptance Test (UAT)	6
	1

2.3.4. Stress Testing	6
2.3.5. Regression Testing	7
2.4. Test Entrance/Exit Criteria	7
2.4.1. Entrance Criteria	7
2.4.2. Exit Criteria	7

4. Have you written a Test Case?

Yes.

What is a Test Case? What does it include?

A Test Case is a document that describes step by step process how to test the application. A Test Case includes Test Case ID, Steps Description, Expected Output, Actual Output, Pass/Fail, Remarks.

Click here to see how a complete Test Case looks like.

5. How many Test Cases did you write in your last project?

Answer: I wrote about 1100 Test Cases in my last project. (The reasonable number of Test Cases varies from 500 to thousands. The number 1100 test cases can be completed in a 6 month project duration).

6. What document did you refer to write the Test Cases?

Requirement document. (NOTE: It can also be Use Cases, or Design Document)

(Note: It depends company to company. In some companies, they use Use Cases. In some companies, they use Requirement Documents and in some companies, they use Design Document. However, in practical scenario, most of the companies have requirement document at least).

Table of Contents..... **ii**

Revision History..... **ii**

1. Introduction..... **1**

1.1 Purpose 1

1.2 Document Conventions..... 1

1.3 Intended Audience and Reading Suggestions..... 2

1.4 Project Scope 2

1.5 References..... 2

2. Overall Description..... **3**

2.1 Product Perspective 3

2.2 Product Features 3

2.3 User Classes and Characteristics 4

2.4 Operating Environment..... 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3. System Features..... **5**

3.1 System Feature 1..... **Error! Bookmark not defined.**

3.2 System Feature 2 (and so on)..... **Error! Bookmark not defined.**

4. External Interface Requirements **5**

4.1 User Interfaces 5

4.2 Hardware Interfaces 6

4.3 Software Interfaces 6

4.4 Communications Interfaces 6

5. Other Nonfunctional Requirements..... **6**

5.1 Performance Requirements..... 6

5.2 Safety Requirements 6

5.3 Security Requirements 7

5.4 Software Quality Attributes..... 7

6. Other Requirements **7**

Appendix A: Glossary..... **7**

Appendix B: Analysis Models..... **7**

Appendix C: Issues List..... **7**

7. Did you have a situation where you did not have any documents (no requirement document, no Use Cases, or no Design Document) and you had to write the Test Cases? How did you write the Test Cases?

Yes. I have been to that kind of scenarios several times. There were companies where they had no documents at all. In that case, I had to discuss the application scenario and functionalities with the Business Analysts or developer. I kind of prepared a document in consultation with Business Analysts and Developers and then started writing Test Cases.

8. Have you worked with the Uses Cases before?

Yes. I have written Test Cases using Use Cases.

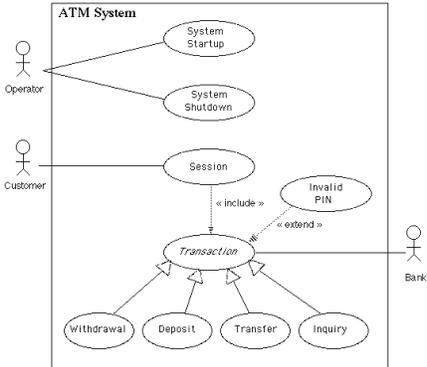
Can you tell me what a Use Case is?

A use case is a document that describes the user action and system response for a particular functionality. (you can also include, For example, in the Use Case given below, is a Use Case for login system for a company called Auto Parts One. This application is being developed by Digital Systems, Inc. The project name is Auto Parts One. However, the business owner (user) is a company called American Auto Parts of the North (imaginary name). Or

What is a Use Case and what does it include?

A Use Case is a document that describes the user action and system response for a particular functionality. It includes cover page, Revision History, Table of Contents, Floe of Events (normal flow and alternative flow), Exceptions, Special Requirements, Pre-conditions and Post-conditions.

This is how it looks (Next Page)



1 Feature Name (Example: ATM Transaction)

1.1 Feature Process Flow / Use Case Model

1.2 Use Case(s)

Use Case ID:	Enter a unique numeric identifier for the Use Case. e.g. UC-1.2.1
---------------------	---

Use Case Name:	Enter a short name for the Use Case using an active verb phrase. e.g. Withdraw Cash		
Created By:		Last Updated By:	
Date Created:		Last Revision Date:	
Actors:	[An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case (primary) and any other actors who will participate in completing the use case (secondary).]		
Description:	[Provide a brief description of the reason for and outcome of this use case.]		
Trigger:	[Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.]		
Preconditions:	[List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each pre-condition. e.g. 1. Customer has active deposit account with ATM privileges 2. Customer has an activated ATM card.]		

Postconditions:	<p>[Describe the state of the system at the conclusion of the use case execution. Should include both <i>minimal guarantees</i> (what must happen even if the actor's goal is not achieved) and the <i>success guarantees</i> (what happens when the actor's goal is achieved. Number each post-condition. e.g.</p> <ol style="list-style-type: none">1. Customer receives cash2. Customer account balance is reduced by the amount of the withdrawal and transaction fees]
Normal Flow:	<p>[Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description.</p> <ol style="list-style-type: none">1. Customer inserts ATM card2. Customer enters PIN3. System prompts customer to enter language performance English or Spanish4. System validates if customer is in the bank network5. System prompts user to select transaction type6. Customer selects Withdrawal From Checking7. System prompts user to enter withdrawal amount8. ...9. System ejects ATM card]

Now, Let us write Test Cases based on this Use Case. Remember, one Use Case can have many Test Cases. For example, look below:

Project Name: Test Case Template						
Test Case ID: Fun_10			Test Designed by: <Name>			
Test Priority (Low/Medium/High): Med			Test Designed date: <Date>			
Module Name: Google login screen			Test Executed by: <Name>			
Test Title: Verify login with valid username and password			Test Execution date: <Date>			
Description: Test the Google login page						
Pre-conditions: User has valid username and password						
Dependencies:						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= sample@gmail.com	User should be able to login	User is navigated to	Fail	
2	Provide valid username	Password: 1234		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					
Post-conditions: User is validated with database and successfully login to account. The account session details are logged in database.						

9. What is Software Development Life Cycle?

The systems (or software) development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

It includes the following different stages:

1. Requirement phase
2. Design phase
3. Coding (programming)
4. Testing
5. Release (Production)
6. Maintenance (Support)

10. What is Business Requirement Document (BRD)?

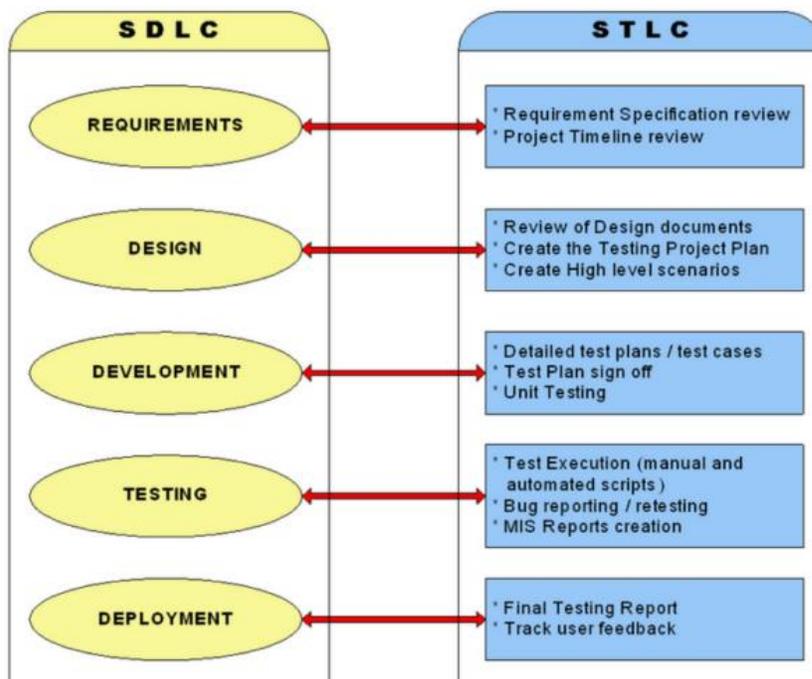
It is a document that describes the details of the application functionalities which is required by the user. This document is written by the Business Analysts.

What is Software Testing Life Cycle (STLC)?

The testing of software has its own life cycle. It starts with study and analyzing the requirements. Here is the software testing life cycle:

1. Requirement Study
2. Test Planning
3. Writing Test Cases
4. Review the Test Cases
5. Executing the Test Cases
6. Bug logging and tracking
7. Close or Reopen bugs

Software Development Life Cycle VS Software Testing Life Cycle



What is Business Design Document?

It is the document which describes the application functionalities of the user in detail. This document is the further details of the Business Requirement Document. This is a very crucial step in the SDLC. Sometimes the Business Requirement Document and Business Design Document can be lumped together to make only one Business Requirement Document.

What is Code Generation or Program?

Coding is the process of translating the Business Design Document into the machine readable form. If the design is done in detailed manner, the Code Generation can be done without much application. Programming tools like Compilers, Interpreters and Debuggers are used to generate the code thru different high level language like C, C++, Pascal, Java.

11. What is a Module?

A 'Module' is a software component that has a specific task. It can be a 'link' which can go inside to its component detail.

12. What is meant by Walk-thru meeting?

Before start working in a module and/or after accomplishing the testing of a module, the tester calls a meeting to disseminate his findings or to share his queries to other tester or leads of the company working on the same application that is called the Walk-thru meeting.

13. What is Build?

When each of the different modules of software is prepared, they are put in a single folder by the Configuration Management Team (CMT) and it is called the 'Build'. In other word, the developers put their code in the shared location (folder) and all those code (modules) are combined together so that it is a complete application that works.

What is meant by the Build Deployment?

When the Build so prepared by the CMT is sent to different Test Environments, it is called the Build Deployment.

14. What is Test Strategy?

A test strategy is an outline that describes the testing portion of the software development cycle. It is created to inform project managers, testers, and developers about some key issues of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the project, and the testing environment.

The test strategy describes how the product risks of the stakeholders are mitigated at the test-level, which types of test are to be performed, and which entry and exit criteria apply. (source: Wikipedia)

The test strategy is created based on development design documents.. It is written by the Test Manager or Lead.

The following are some of the components that the Test Strategy includes:

1 Test Levels. 2 Roles and Responsibilities. 3 Environment Requirements. 4 Testing Tools. 5 Risks and Mitigation. 6 Test Schedule. 7 Regression Test Approach. 8 Test Groups. 9 Test Priorities. 10 Test Status Collections and Reporting. 11 Test Records Maintenance. 12 Requirements traceability matrix. 13 Test Summary

Test Strategy Example

Prior to the start of Cycle 0 testing, key preparation activities will be conducted in order to properly prepare for the testing. Key activities include:

Project Team will get the tools for logging defects.

The Project build team is ready.

Business Scenario Test Cases are ready.

The data for testing are ready for all scenarios.

Define Security and Privileges requirements should be ready

The timeline below identifies these key items to complete prior to Date xxx xx, xxxx

Cycle 0 Dependencies: Test Cases completion; QA Environment Up Focus: Test preparation - project team preparation and logistical planning: System Configuration (Done by the deployment team) QA Environment should be ready Load QA "Box" Load Test Data Refine Test Cases Dry-Run Cycle Fix Errors Reload

Cycle 1 Dependencies: Interfaces/Conversions; Site Connectivity Focus: Business scenario – Business-User testing: Execute Test Scenarios Document Test Results Fix Errors Reload Run multiple iterations as necessary

Cycle 2 Dependencies: Reports/Security/Privileges, External Assessment Plan Focus: Security, Reports and External Process testing – Business-User and Technical lead testing: onfidential Execute Test Scenarios Test security/privileges Begin development of training and end user material Document Test Results Fix Errors Upon the completion of Cycle 2 testing, detailed results will be summarized and presented for review to Mercury Management, which should be scheduled for the appropriate date decided by the Mercury Management.

Are Test Plan and Test Strategy same type of document?

No. They are different documents. Test Plan is a document that collects and organizes test cases by functional areas and/or types of testing in a form that can be presented to the other teams and/or customer where as the Test Strategy is the documented approach to testing. Test Plan is prepared by the tester whereas the Test Strategy is prepared by the QA Manager or QA lead.

Both are important pieces of Quality Assurance processes since they help communicate the test approach scope and ensure test coverage while improving the efficiency of the testing effort.

15. What does the Test Strategy include?

It includes introduction, scope, resource and schedule for test activities, acceptance criteria, test environment, test tools, test priorities, test planning, executing a test pass and types of test to be performed.

16. What are different types of software testing?

Different types of testing carried out are:

- 1) Unit testing
- 2) Shakeout testing
- 3) Smoke testing (Ad-hoc testing)
- 4) Functional testing
- 5) Integration testing
- 6) Regression testing
- 7) System testing
- 8) Load testing
- 9) Stress testing
- 10) Performance testing
- 11) User acceptance testing
- 12) Black box testing
- 13) White box testing
- 14) Alpha testing
- 15) Beta testing

Note: Except the Shakeout testing and Unit testing which are respectively done by the CMT and Coder/Developer, all other testing are done by the QA Engineer (Tester).

- 1) **Unit testing:** It is a test to check the code whether it is properly working or not as per the requirement. It is done by the developers (Not testers).
- 2) **Shakeout testing:** This test is basically carried out to check the networking facility, database connectivity and the integration of modules. (It is done by the Configuration Team)
- 3) **Smoke testing:** It is an initial set of test to check whether the major functionalities are working or not and also to check the major breakdowns in the application. It is the preliminary test carried out by the SQA tester.
- 4) **Functional testing:** al It is a test to check whether each and every functionality of that application is working as per the requirement. It is major test where 80% of the tests are done. In this test, the Test Cases are 'executed'.
- 5) **Integration testing:** It is a test to check whether all the modules are combined together or not and working successfully as specified in the requirement document.

17. Testing Tool: The following tools should be provided to the QA Team prior to commencement of testing:

1. Defect Logging and Tracking tool: Quality Center or ClearQuest, or Bugzilla
2. QTP (Quick Test Professional): For Functional and Regression Testing
3. LoadRunner: For Load Testing.

18.. Overall Completion Criteria for Testing A critical aspect of monitoring the progress and acceptance of the end result of Testing for the Version 1.0 will be a clear understanding and agreement on completion criteria. One key aspect of this is the successful identification and resolution of issues/defects. Issues/defects are categorized as follows:

1. **CRITICAL** Causes system to crash or does not support critical functional requirements tied to Key Performance Parameters. No acceptable work-around. Unable to demonstrate completed business scenario.
2. **HIGH** Causes significant system degradation or a difficult work-around to support critical functional requirements tied to Key performance Parameters. Business scenario can be completed with difficulty.
3. **MEDIUM** Causes moderate system degradation or a work-around to support critical functional requirements tied to Key performance Parameters. Business scenario can be completed with moderate to minor impact to business process.
4. **LOW** Causes minor system degradation or minor work-around to support critical functional requirements tied to Key performance Parameters. Business scenario can be completed with minor to no impact to business process.

- Completion criteria for Pre-System Test have been established as following:

- Each business Test Case exercised and Each required conversion, interface and enhancement completed No CRITICAL unresolved issues/defects, may allow HIGH if necessary, but have an action plan.
- Action plans exist for remaining HIGH, MEDIUM and LOW issues/defects Security and Privileges(Profiles) established and tested
-
- Completion criteria for System Test have been established as following:
 - Each business Test Case exercised and Each required conversion, interface and enhancement completed No CRITICAL, HIGH and MEDIUM unresolved issues/defects remain
 - Action plans exist for remaining LOW issues/defects Security and Privileges (Profiles) established and tested
 -
- Completion criteria for Beta Test have been established as following:
 - Each business Test Case exercised and Each required conversion, interface and enhancement completed No CRITICAL, HIGH, MEDIUM and LOW unresolved issues/defects remain Security and Privileges (Profiles) established and tested Test

19. STATUS REPORTING

Test preparation and testing progress will be formally reported during a weekly Status Meeting to the Director of Product Development. A status report will be prepared by the Test Manager to facilitate this meeting. This report will contain the following information:-

1. Current Status v. Plan (Ahead/Behind/On Schedule)
2. Progress of tasks planned for previous week
3. Tasks planned for next week including tasks carried from previous week
4. Error Statistics from Error Measurement system
5. Issues/Risks
6. AOB (Any Other Business)