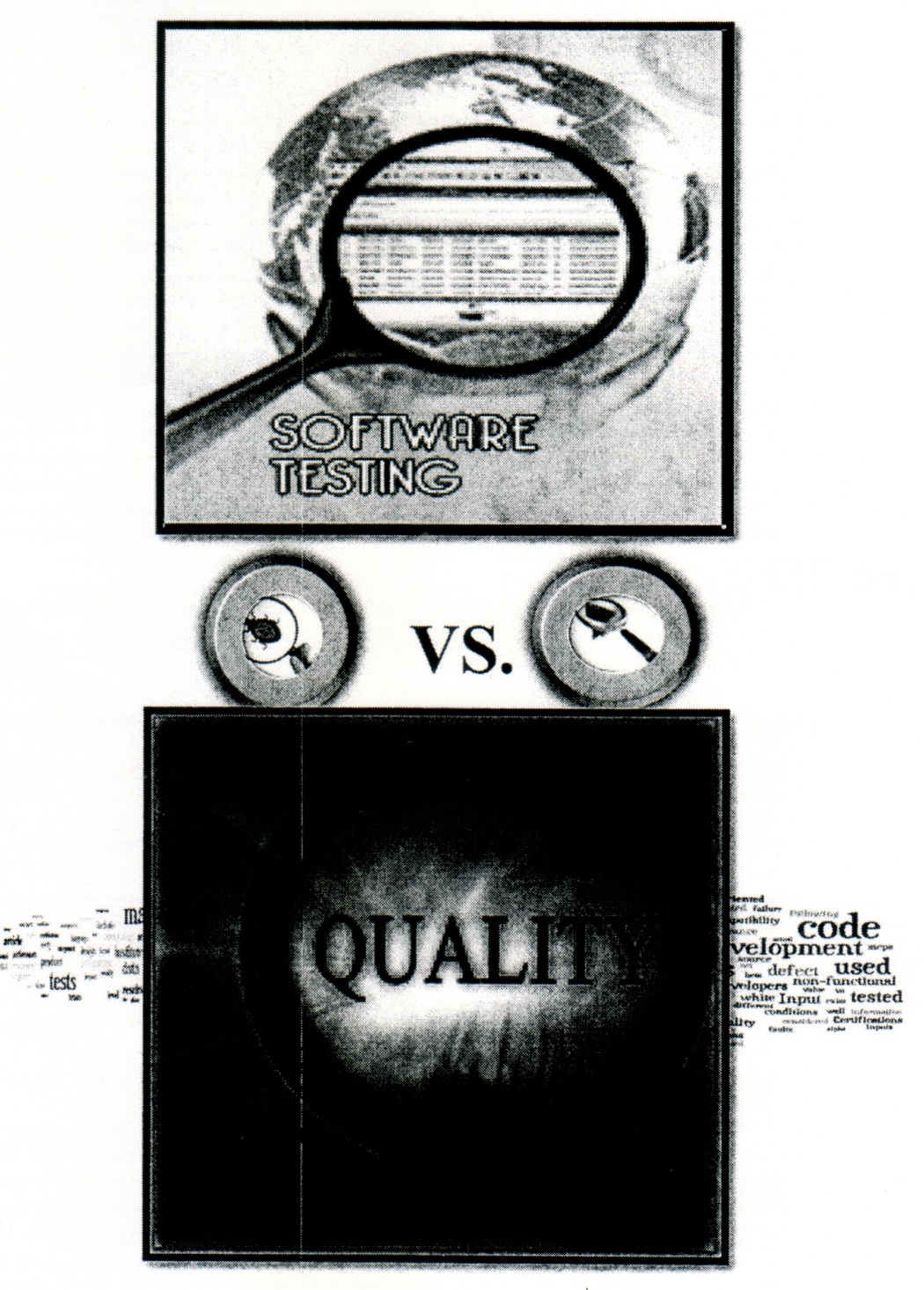
**SOFTWARE TESTING**

****

**QUALITY ASSURANCE**

**Objective**

The object of this tutoring is to provide information about the software testing industry. You will learn about software testing process, techniques, technologies, tools, terminologies, and more other information to guide you to be comfortable with the QA industry.

You will learn about Software Development Life Cycle, Software Methodologies, Software Testing Life Cycle, Bug Life Cycle, Business Requirement Document, Requirements, Use Case, Test Cases, Testing Artifacts, and Test Management Tools. We will also learn Automation Tools and many other documents and information. ***It is your job to study and learn and master the process.*** Good luck.

**Contents**

1. Software Testing & Why

2. Key Terms and Involvement in developing of software

3. Fundamental of Testing

4. Manual Testing vs. Automated Testing

5. Software Testing Terminology

6. Requirement

7. Test Case

8. Test Plan

9. Software Development Methodologies

10. Software Development Life Cycle (SDLC)

11. QA involvement in SDLC

12. Software Testing Type

13. Software Testing Life Cycle (STLC)

14. Software Testing Techniques

15. Testing Documents

16. Use Case

17. Bug Life Cycle

18. Software Release Life Cycle

19. HP QC - Quality' Center

20. Questions vs. Answers

21. Tools and Certificates

**History of Software Bugs**

A software bug is an error, flaw, mistake, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended ways. Most bugs arise from mistakes and errors made by people in either a program's source code or its design, and a few are caused by compilers producing incorrect code. A program that contains a large number of bugs, and/or bugs that seriously interfere with its functionality, is said to be buggy. Reports detailing bugs in a program are commonly known as bug reports, fault reports, problem reports, trouble reports, change requests, and so forth.

Bugs trigger errors that can in turn have a wide variety of ripple effects, with varying levels of inconvenience to the user of the program. Some bugs have only a subtle effect on the program's functionality, and may thus lie undetected for a long time. More serious bugs may cause the program to crash or freeze. Others qualify as security bugs and might for example enable a malicious user to bypass access controls in order to obtain unauthorized privileges.

The results of bugs may be extremely serious. Software bugs can potentially cause monetary and human loss, history is full of such examples:

• Bugs in the code controlling the Therac-25 radiation therapy machine were directly responsible for some patient deaths in the 1980s.

• In 1996, the European Space Agency's US$1 billion prototype Ariane 5 rocket was destroyed less than a minute after launch, due to a bug in the on-bard guidance computer program.

• China Airlines Airbus A300 crashing due to a software bug on April 26, 1994 killing 264 innocent lives

• In April of 1999, a software bug caused the failure of a $1.2 billion military' satellite launch, the costliest accident in history

• In May of 1996, a software bug caused the bank accounts of 823 customers of a major U.S. bank to be credited with 920 million US dollars

As you see, **testing is important because software bugs could be expensive or even dangerous**

(Note: Memorize at least two examples for interview questions)

In 2002, a study commissioned by the US Department of Commerce' National Institute of Standards and Technology concluded that "software bugs, or errors, arc so prevalent and so detrimental that they cost the US economy an estimated $59 billion annually, or about 0.6 percent of the gross domestic product”. (Wikipedia)

**Quality Assurance (QA)**

To assure quality of an application or software by testing its functionality, performance, and security throughout the SDLC (Software Development Life Cycle), to provide a bug and defect free software, and making sure it is built as the client expects and needs.

**Software Quality**

Software Quality is a process of meeting customer expectations and customer requirements. The features of Software Quality are described below:

* Meet customer requirements
* Meet customer expectation
* Time to Market
* Cost of product

**Software Testing**

* Process of finding the bugs in a software and help in verifying and validating the software is working properly
* To detect software failure so that defect may be covered and corrected
* Software does what it suppose to do
* To detect the defect.

**Why Testing is important?**

* It is important to deliver a good, reliable, bug and defect free software.
* To meet client Requirement and Expectation, what client desire from the software.
* If you fail to deliver quality software, client will not be happy and may lose his or her business and could be bad reputation for business.
* Extremely important for Tester to check the entire system in/out to make sure software is 99.9% defect free and fix them on time.

**What Could Happen Without Testing?**

As we know and have experience with software that did not work as we expected. Software that does not work can have a large impact on an organization or company. It can lead to many issues including:

* **Damage to Business Reputation** - if company fail to provide quality software , customer will lose their confidence in the organization and may take their business somewhere else
* **Loss of Time-**bynot testing the company will have to spend more time to fix the defects while the product is in the market.
* **Loss of Money-**theclient may return the software or may ask to re-develop the software again.
* **Injury or death**-specialty with medical or some safety, the critical system could result in injury or death if it does not work properly.

**Key Terms and Involvement in Developing Quality Software**

**Important People:**

**Project Manager (PM)**

-Managing the project and Prepare the project plan/ charter

-Responsible for consistent reporting, risk mitigation, timeline, and cost control

**Business Analyst (BA)**

- Middle man between the client and the company

- Gather and collect all the information about the software needed to be built.

- Creating all the requirements into BRD (Business Requirement Document)

**Product Manager / SME (Subject Matter Expert)**

- Person who is expert about application (Software)

- Knows everything about the software and how it should be working

- If QA not sure about functionally of the application and could come to SME and ask question about the application and functionality.

**Developer / Programmer**

- Responsible for develop the software by writing computer code

- Responsible for unit and integration testing

-Responsible for fixing the defect of the application

**QA Tester**

- Test the application by testing its **functionality, performance** and **security**

**QA Tester/ Analyst-**testthe application throughout the STLC

**QA Automation Engineer** - Person who automates the testing process using Automation tools (Like QTP/UFT, Selenium, LoadRunner, Ranorex etc.)

**QA Lead** - QA Tester leader and reports to QA Manager

**QA Manager** -Manager of QA Team and Reports to Director

**UAT Tester** - User Acceptance Tester

**Important Terms:**

**BRD (Business Requirement Document) -** BA creates

- This document defines customer requirements to be developed as software. It is also known as Customer Requirement Specification or User Requirement Specification(**CRS / URS**).

**TRD (Technical Requirement Document) -** Developer creates

**SRS (Software Requirement Specification) -** BA creates

- BRS and TRD could make up SRS document

**BDD (Business Design Document) -** BA creates

**DDD (Detail Design Document) -** Developer creates

* + **GUI** - Graphic User Interface
  + **AUT** - Application under test
  + **SCR** - Service change request (to make suggestion)
  + **RTM** - Requirement Traceability Matrix (Mapping between Test Case & Requirement.
  + **SOP**- Standard Operating Procedure used in a variety of different contexts.
  + **SoapUI**- Open source web service testing tool for testing application functionality, covers web service inspection, invoking, development, simulation and mocking
  + **Test Data** –Information used in testing (Real Data: Actual Data and Dummy Data: Fake data used during testing.)
  + **ETL**( Extract Transform Load) three database migrate data from one database to another
  + **DBA** (Database Administrator)-Manage the Database, QA will ask DBA for test data.
* **Artifacts** - any document that goes into project (Test Strategy, Test Plan & Etc.
* **Landing Page -** Home Page
* **Demo** - Demonstrate or Presentation of complete work before going live
* **Showstopper** - Prevent you from going to future testing
* **Hot Fix** - Bug should be fix right away
* **Defect** - Bug in application or crush all the sudden -Fault in System.

**The Fundamentals of Testing Process**

**I. Test Plan** **(Test Lead or Test Manager)**

- is the document that describes the scope, approach, resource scheduling, and responsibilities and other testing activities goes into the testing effort.

a. Initial test plan/rough estimation

b. Master test plan (Final)

**II. Test Cases Specification** **<Test Engineer>**

Specify the Test Case (Type ofTechnique)

**BVA**

**ECP**

**Error Handling**

**III.** **Test Case Execution**

Execute each test case

**IV.** **Test Recording- Bug Reporting**

**a. Analysis and Preparation of reports**

Compare with the expected result and actual result.

**V.** **Checking for test completion - Exit Criteria**

Checking the test against the test completion criteria

**Manual Testing vs Automated Testing**

**Manual Testing:**

- *Manual testing is the process of testing the software or application to find the defects without the help of Automation testing tool.* Tester write test plan and test cases to ensure completeness of testing. Manual testing is the oldest and most rigorous type of software testing.

**Process of Manual Tester:**

**1** **Understand the requirement (BRD)**

**2** **Specify the test cases**

**3** **Prepare the test environment**

**4** **Execute the test case manually**

**5** **Record the test result Pass/Fail**

**6** **Check for test completion**

**Automation Testing:**

- *Automation* *testing is the process of testing the software with the help of Automation testing tool.* Tester develop automation test script (write some computer programs). Once the test has been automated, they can be run over and over again.

**Process of Automation Tester:**

**1** **Prepare to record**

**2** **Recording**

a. Normal recording

b. Analog recording

c. Low level recording

**3** **Edit Script**

Checkpoints (expected), Parameterization, Conditions

**4** **Debug**

a. Breakpoint

b. Step into

c. Step over

d. Step out

**5 Run**

Execute the automation test script

**6 Analyze results**

It compares expected with actual values

**Advantages of Automation Testing:**

1) **Fast:** Automation testing tools run the tests significantly faster than human users.

2) **Reliable:** Eliminating human error and perform the same operation every time.

3) **Reusable**: Reuse tests on different versions of an application, even if the user interface

(UI) changes.

4) **Repeatable:** Same test can be repeated over and over.

5) **Programmable:** Can program sophisticated test to test the application.

6) **Comprehensive:** Can build a suite of tests that covers every feature in your application.

7) **Cost** **Reduction:** Reduce the resources (Human user)

**8) Better Quality Software:** The application can be tested many times by executing the

automation test case over and over, which ensure the quality of the application.

**Disadvantages of Automated Testing:**

1) Proficiency is required to write the automation test script.

2) Debugging the test script is major issue. If any error is present in the test Script, sometimes it may lead to deadly consequences.

3) Maintenance is costly. (Note: If the application is dynamic, AUT object changes property and value you need to edit your automation test script every time before execute the script)

**Types of testing can be automated:**

• **Functional**: Testing the functionality of the application & making sure that it is working properly as it expected.

• **Regression:** Retesting the application and to make sure that the behavior of the system has not changed.

**•** **Exception or Negative:** Forcing error conditions in the system.

**•** **Stress:** Determining the absolute capacities of the application.

• **Performance**: Testing the application that it can meet the demand during peak and off peak time.

**•** **Load:** Determining the points at which the capacity and performance of the system become degraded to the situation that hardware or software upgrades would be required.

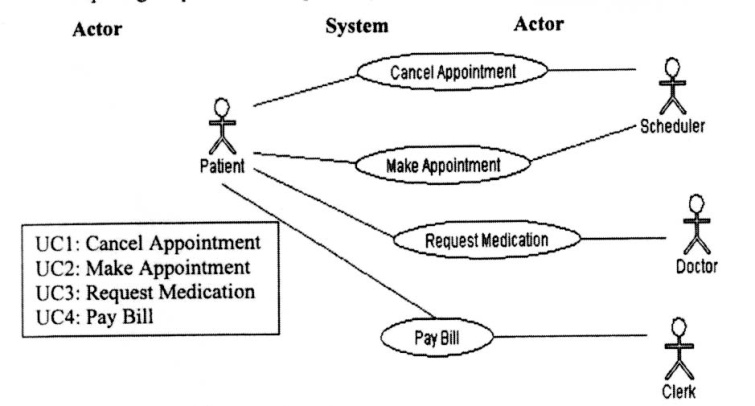
**Software Testing Terminology**

**Use Stories (US)**

*It's a conversation between BA and Client. BA take notes on an index card.* Story of how the application will need to be work. Use stories answer WHO (I), WHAT (something) and WHY (benefit)? BA writes the Use Story.

Example: 1 (who) want to purchase a computer (what) for my school (why).

**Use Case (UC)**

*It's a step by step sequence of actions between the user (In pic. Actor) and the system.* It's a document that describes the user action and system response for a particular functionality. It is helpful in exposing Requirement and planning the project. BA is responsible for creating this document.

**Content of Use Case**

1) General Information: UC names, ID, Project Name

2) Change Log: Who is making change and what is it.

3) Assumption: Pre-Condition and Post-Condition.

4) Data Source: Data needed, where it is coming from. Client, DBA, BA.

5) Description of Process: What? How? What?

6) Primary Flow: Very common or normal way will do.

7) Alternative Flow: Other ways of accomplish.

**Requirement**

*A requirement is what client want or desire from a system.* What client wants the system to do? All requirements should be in SMART. One of the biggest challenges in QA industry is unclear or poor requirements. Requirements should be clear, complete, reasonably detailed, attainable and testable.

**Specific**

• Does it address a real business problem?

**Measurable**

• Are we able to measure the problem, establish a baseline, and set targets for

improvement?

**Attainable**

• Is the goal achievable? Is the project completion date realistic?

**Relevant**

• Does it relate to a business objective?

**Time Bound**

• Have we set a date for completion**?**

**Non-Requirement**

*Some non functional attributes that may be or may not be mention in the document (BRD, BRS) are called Non-Requirement*

1. Performance: How fast

2. Reliability: How often

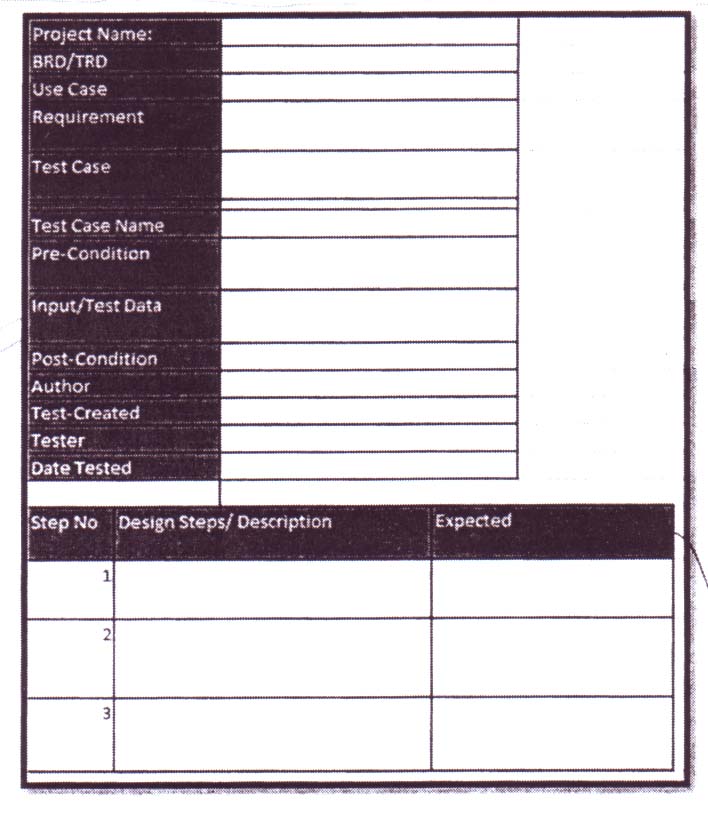
3. Security: Access level (Unauthorized user)

4. Supportability: Modify application (Updates, Edit)

5. Usability: User friendly (Easy to use)

**Test Case**

*A Test Case is a document that describes the step by step process of “how to test single behavior or function” of the application.* ***A Test Case includes Test Case Name, Test Case ID, Pre-Condition, Post-Condition, Steps Number, Steps Description, Expected Output, Actual Output, Pass/Fail and Remarks***.



**Test Script**

*A script is a collection of statements which are written using a scripting language. Scripts are developed to test the functionality of the application.*