**Why is software testing necessary?**

[Software Testing](http://tryqa.com/what-is-a-software-testing/) is necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong – [humans make mistakes all the time](http://tryqa.com/when-do-defects-in-software-testing-arise/).

Ideally, we should get someone else to check our work because another person is more likely to spot the flaws.

There are several reasons which clearly tells us as why Software Testing is important and what are the major things that we should consider while testing of any product or application.

Software testing is very important because of the following reasons:

* Software testing is really required to point out the [defects](http://tryqa.com/what-is-defect-or-bugs-or-faults-in-software-testing/)and errors that were made during the [development phases](http://tryqa.com/what-are-the-software-development-life-cycle-sdlc-phases/).
* It’s essential since it makes sure that the customer finds the organization reliable and their satisfaction in the application is maintained.
* It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence.
* Testing is necessary in order to provide the facilities to the customers like the delivery of high-quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
* Testing is required for an effective performance of software application or product.
* It’s important to ensure that the application should not result into any [failures](http://tryqa.com/what-is-a-failure-in-software-testing/)because it can be very expensive in the future or in the later stages of the development.
* It’s required to stay in the business.

**Software Testing Life Cycle (STLC)**

Software Testing Life Cycle (STLC) is defined as a sequence of activities conducted to perform Software Testing.

Contrary to popular belief, Software Testing is not a just a single activity. It consists of a series of activities carried out methodologically to help certify your software product.

### Different Phases of the STLC Model



## 1. Requirement Analysis

During this phase, test team studies the requirements from a testing point of view to identify the testable requirements.

The QA team may interact with various stakeholders (Client, Business Analyst, Technical Leads, System Architects etc) to understand the requirements in detail.

Requirements could be either Functional (defining what the software must do) or Non Functional (defining system performance /security availability )

Automation feasibility for the given testing project is also done in this stage.

**Activities**

* Identify types of tests to be performed.
* Gather details about testing priorities and focus.
* Prepare [Requirement Traceability Matrix (RTM)](https://www.guru99.com/traceability-matrix.html).
* Identify test environment details where testing is supposed to be carried out.
* Automation feasibility analysis (if required).

**Deliverables**

* RTM
* Automation feasibility report. (if applicable)

## 2. Test Planning

Typically, in this stage, a Senior QA manager will determine effort and cost estimates for the project and would prepare and finalize the Test Plan. In this phase, Test Strategy is also determined.

**Activities**

* Preparation of test plan/strategy document for various types of testing
* Test tool selection
* Test effort estimation
* Resource planning and determining roles and responsibilities.
* Training requirement

**Deliverables**

* [Test plan](https://www.guru99.com/test-plan.html) /strategy document.
* [Effort estimation](https://www.guru99.com/testing-estimation.html) document.

## 3. Test Case Development

This phase involves the creation, verification and rework of test cases & test scripts. [Test data](https://www.guru99.com/software-testing-test-data.html), is identified/created and is reviewed and then reworked as well.

**Activities**

* Create test cases, automation scripts (if applicable)
* Review and baseline test cases and scripts
* Create test data (If Test Environment is available)

**Deliverables**

* Test cases/scripts
* Test data

## 4. Test Environment Setup

Test environment decides the software and hardware conditions under which a work product is tested. Test environment set-up is one of the critical aspects of testing process and **can be done in parallel with Test Case Development Stage**.

**Activities**

* Understand the required architecture, environment set-up and prepare hardware and software requirement list for the Test Environment.
* Setup test Environment and test data
* Perform smoke test on the build

**Deliverables**

* Environment ready with test data set up
* Smoke Test Results.

## 5. Test Execution

During this phase, the testers will carry out the testing based on the test plans and the test cases prepared. Bugs will be reported back to the development team for correction and retesting will be performed.

**Activities**

* Execute tests as per plan
* Document test results, and log defects for failed cases
* Map defects to test cases in RTM
* Retest the Defect fixes
* Track the defects to closure

**Deliverables**

* Completed RTM with the execution status
* Test cases updated with results
* Defect reports

## 6. Test Cycle Closure

Testing team will meet, discuss and analyse testing artifacts to identify strategies that must be implemented in the future, taking lessons from the current test cycle. The idea is to remove the process bottlenecks for future test cycles and share best practices for any similar projects in the future.

**Activities**

* Evaluate cycle completion criteria based on Time, Test coverage, Cost, Software, Critical Business Objectives, Quality
* 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.

**Deliverables**

* Test Closure report
* Test metric

### What is Verification?

The verifying process includes checking documents, design, code, and program.

### What is Validation?

Validation is a dynamic mechanism of Software testing and validates the actual product.

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| * The verifying process includes checking documents, design, code, and program | * It is a dynamic mechanism of testing and validating the actual product |
| * It does **not** involve executing the code | * It always involves executing the code |
| * Verification uses methods like reviews, walkthroughs, inspections, and desk- checking etc. | * It uses methods like Black Box Testing, White Box Testing, and non-functional testing |
| * Whether the software conforms to specification is checked | * It checks whether the software meets the requirements and expectations of a customer |
| * It finds bugs early in the development cycle | * It can find bugs that the verification process cannot catch |
| * Target is application and software architecture, specification, complete design, high level, and database design etc. | * Target is an actual product |
| * QA team does verification and make sure that the software is as per the requirement in the SRS document. | * With the involvement of testing team validation is executed on software code. |
| * It comes before validation | * It comes after verification |

**Desktop application:**  
  
A native application that executes on a user's local machine.  This application may or may not have a network component, although most desktops have network component these days, even if it's just to update itself online.  To keep things simple, let's put it this way: if you need to update the application, an update needs to be downloaded locally.  
An example of a desktop app would be MS Word, Adobe Photoshop, a web browser.

**Mobile application:**  
  
An application built to run natively on a mobile device.  The most common devices these days are either iOS or Android based, but there is a small population of Windows mobile users and a few folks still clinging to their Blackberries.  Mobile applications, like desktop apps may or may not have a network component -- but with the exception of the calculator on my phone, I can't think of any that don't.  Just like a desktop app, if you need to make an update to this app, something needs to be downloaded and installed.  
An example of a mobile app would be Angry Birds, Harvest for iOS (sorry, shameless plug) or the SMS app.  
  
**Web applications:**  
  
Applications that run 100% within a browser. This is where the waters get muddy, because there are a group of apps on both the desktop and on mobile that are just web apps, running within an app-specific browser.  An example of one of these would have been the early versions of the Facebook app -- those were just a bunch of web views running within a browser window.  Sometimes those are hard to identify, but a sure-fire tell is if your app updates without you needing to do anything from the App Store/Play Store.