

Description of QA work at Zesium

1. General Overview of Software Testing terms

Software Quality

Quality level of a software product is determined by the extent to which the product meets its specifications and satisfy user needs and expectations.

Validation is the process of evaluating the final software product to check whether the software meets the business needs.

Verification is a process of evaluating the intermediary work products of a software development lifecycle to check if we are in the right track of creating the final product.

Quality assurance (QA) is any systematic process of determining whether a software product or service meets specified requirements.

Quality control (QC) is a set of procedures intended to ensure that a manufactured software product fulfills defined set of quality criteria or meets the requirements of the client or customer.

Testing plays an important role in achieving the quality of a software product.

Software Testing

Software testing is a process of executing a program or application with the intent of finding the defects in software products. Software testing includes execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or by using automated tools.

Main goal of software testing is to check whether the actual results match the expected results and to ensure that the software system is bug free.

Software bug is an error, flaw or failure in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

Seven testing principles that are widely practiced in software industry

Test Principles help us to create an effective Test Strategy, that suits the needs of our business.

1. Exhaustive testing is not possible – instead, we need the optimal amount of testing based on the risk assessment of the application.
2. Defect Clustering – which states that a small number of modules contain most of the defects detected.
3. Pesticide Paradox – if the same set of repetitive tests are conducted, the method will be useless for discovering new defects. To overcome this, the test cases need to be regularly reviewed and revised,

adding new and different test cases to help us find more defects.

4. Testing shows presence of defects – software testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

5. Absence of Error – it is possible that software which is 99% bug-free is still unusable. Software testing is not mere finding defects, but also to check that software addresses the business needs and requirements.

6. Early Testing – testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in early stages of testing.

7. Testing is context dependent – it's necessary to use a different approach, methodologies, techniques and types of testing depending upon the application type.

Software Testing Life Cycle

Software Testing Life Cycle (STLC) is defined as a sequence of activities conducted to perform software testing. It consists of series of activities carried out methodologically to help certify your software product.

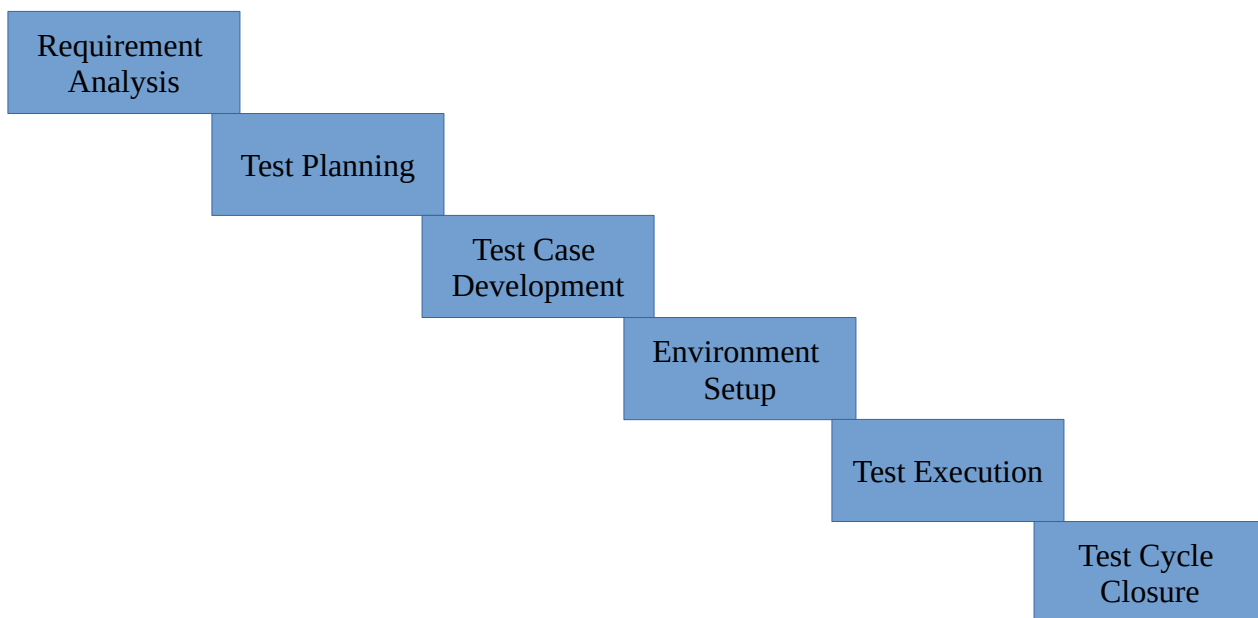


Diagram of STLC

For each of this stages in STLC is necessary: to define Entry and Exit criteria, Test Activities and to generate Reports about conducted tests and gained results.

2. Mobile Applications Testing

Testing applications on mobile devices is more challenging than testing Web applications on desktop. All testing principals are applicable for testing mobile applications, but while testing mobile applications, testers must be aware of the challenges and different expectations that users have in regards to web applications.

Main challenges in Mobile Application Testing:

- Different variants of mobile devices (smartphone, tablet), with different screen sizes and hardware configurations like a hard keypad, virtual keypad (touch screen) etc.
- Different manufacturers of mobile devices,
- Different versions and generations of operating systems (Android, iOS, Windows).
- Compatibility of mobile app's with different sensors that exists on mobile device.
- Different mobile network operators and different characteristics of the network.
- Communication with wireless networks in different environments.

The majority of reports and surveys state that nearly 80% of users delete an app after using it for the first time. The top four reasons for deletion are always bad design, poor usability, slow loading time, and crashes immediately after installation.

Points that developers and testers should keep in mind when developing and testing a mobile app are: to gather information about possible target customer group, to ask customers about their needs because that should solve a problem for the user, pay attention to usability because an app needs to be reliable and robust, also app performance is really important, and an app should have good design.

In order to handle the fast pace of mobile release cycles, a tester should keep the following things in mind: to monitor the mobile device and software market, to know when new phones will be rolled out, to find out about the new features of the operating systems, to keep an eye on your target customer group to see if new devices are showing up in your statistics, to think twice before updating a phone to the latest operating system version, to buy new phones with the latest operating system version and if buying is not an option, to rent the devices.

There are 3 types of mobile apps: a mobile web apps, hybrid apps and native apps.

1. A mobile Web app is a Web site that can be accessed from the device's Web browser. Such Web sites are optimized for mobile browser usage and are independent of the mobile platform. Mobile Web apps are developed with Web technologies such as HTML and JavaScript, particularly with HTML5, CSS3, and JavaScript.
2. Hybrid apps, are apps that consist of different Web technologies such as HTML or JavaScript. Once the Web part has been built, developers are able to compile this code base to the different native formats: Android, iOS, Windows Phone.
3. Native apps are programmed with a specific programming language for the specific mobile platform. For example, Android apps are developed in Java, whereas iOS apps are written in Objective-C or Swift. Native apps generally exhibit excellent performance and are optimized for mobile platforms. In most cases, native apps look and feel great and are able to support every possible gesture on the touchscreen.

Pros: Native apps have full access to platform-specific hardware and software features, they have good performance because they are optimized for the specific mobile platform; they have a good look-and-feel; they offer good usability if the platform UI guidelines are met; they have full access to all touch gestures (if implemented). Native app distribution is easy. Users can search for your app. Native apps can store data offline.

Cons: The amount of development work increases with each supported platform because each platform needs its own code base. The approval process can be quite long. Updating a released app may take some time (which is annoying when it comes to urgent bug fixes). Development costs can be higher.

A few major points in testing mobile apps:

- The installation process is the first impression a mobile customer has of an app. If the installation fails due to errors or problems, the customer will not try to download the app again and will move on to another one. Besides installing and uninstalling an app, users are also able to update the app from one version to another.
- The speed and availability of Internet connection is very important and it could vary in different regions and environments. What happens if a user has a bad or no Internet connection while using the app? Will it crash or will it still work? What happens if the mobile device changes network providers while the app is being used? All of these questions are very hard to answer when testing an app in a lab. A mobile tester needs to be mobile and connected to data networks while testing apps so it's important to test application in real-life environments. Communication infrastructure must be available. The mobile infrastructure is currently in its fourth generation known as 4G or LTE (Long Term Evolution).
- Different OS versions: a tester should always be aware of the OS version he/she is testing. It's not unusual to find some bugs only on specific versions.
- Different screen resolutions: a tester should install and test an app on as many as possible different mobile devices. Since it's impossible to have so many devices in development environment, a tester should use an emulator/simulator to check compability. Especially when it comes to Android world which is not that simple.
- When it comes to the Usability of the application, it is important that an app be easy to use, otherwise it might end up with low ratings. Is the user able to see the elements with which he or she can interact? Is every element on the screen clear and understandable? This aspect of usability testing gets more difficult over time because the more you work with the app, the more blind you will become to things that are not self-explanatory.
- Application must allow users to make mistakes. If a user, for example, misses a required input field, the app should provide friendly and useful error messages. A tester should check that the error messages are not too generic and that they describe the error in a way that people without a technical background can easily understand.
- Buttons, labels, and other screen elements should be checked if they are big or small enough to be used. Screen division is important, so the app should be checked on smaller and bigger screens. A tester should keep an eye on visual transitions such as animations or elements that fade in or out and if the main navigation and sub-navigation of the app are easy to use. Are there any unnecessary taps that the user must perform in order to reach his or her goal?

- The text used within the app should be clear and easy to understand. As a tester if you're not sure, ask different people if they understand the wording and the feature behind it. If an app supports more than one language, be sure the text fits into every UI element and that the translation is correct. The UI elements must have the same look-and-feel, the same text, spacings, colors, and images.
- It is necessary to test battery consumption during use. If an app consumes too much power, users will delete it and move on to another one.
- It is useful to think out of the box, to be creative in testing. Testers need to think of edge cases that are most likely to happen in the real world when lots of people will be using the mobile app in a totally different way from what development team expect. For example tester could rotate or flip the device, use more than one hand to interact with the app, and press buttons as fast as possible. While doing that, it's necessary to watch out for any unexpected behavior, crashes, freezes, error messages that make no sense, and any other strange things.

Most common tools and frameworks used for Mobile applications testing on today's market are: Appium, Espresso, Monkey Runner, Katalon Studio, etc.

3. Web applications Testing

Software Testing that focuses on testing the web applications is known as Web testing. This type of testing is focused on testing the applications that are hosted on web for potential bugs before its made live or before code is moved into the production environment.

Web Testing focuses on testing of the application interfaces and logical functionality that are implemented in accordance with the requirements of clients.

Most common web applications are websites and they are essentially client/server applications.

While testing Web applications, consideration should be given to the interactions between HTML pages, TCP/IP communications, Internet connections, firewalls, applications that run in web pages (javascript, plug-in applications) and applications that run on the server side (database interfaces).

When testing web applications, the following tests must be performed:

- Functionality testing – means testing links, forms on all pages, cookies, HTML/CSS, databases,
- Usability testing – means testing ease of use, navigation on page, user satisfaction, appearance,
- Interface testing – is performed to verify the interface and the correct data flow from one system to other. Meaning testing of web server and application server interface, application server and database server interface,
- Compatibility testing – means testing Browser compatibilty, OS compatibility, compatibility for mobile browsing,
- Performance testing – means conducting the Web Load Testing, Web Stress Testing, Test application performance on different internet connection speed.
- Security testing – Primary reason for testing the security of a web is to identify potential vulnerabilities. Security testing involves: Network Scanning, Vulnerability Scanning, Password Cracking, Log Review, Integrity Checkers, Virus Detection.

Most common tools used for Web applications Testing on today's market are: Selenium, Ranorex, Katalon Studio, etc.

4. Our Practice and Solutions

Testing Process of Zesium QA Team consists of following activities:

1. Participation during software planing process,
2. Conducting test planing and creating Test Plans,
3. Tool selection and defining Test Environment,
4. Selection of whitch components of application need to be tested,
5. Writing Test Scenarios and Test Cases,
6. Creating Test documentation,
7. Executing Tests,
8. Generating Test Report documentation,
9. Providing detail defect tracking and analysis,
10. Communication with the development team,
11. Participation in defect removal,
12. Conduct Regression Testing.

Most commonly used testing Types/Methods/Approaches in Testing Process of Zesium QA Team:

- Manual testing – is a type of where tester manually conducts a test by following a set of predefined test cases. In this process a tester uses functions and features of an application under test as an end-user in order to verify that software is working as required. Manual testing is typically implemented in testing mobile applications.,
- Black Box testing – is a method where a tester doesn't have access to source code of the application under test. Tester tests an application in order to examine functionality of an application without insight into internal structures of application. This testing method is often used for testing mobile applications.
- White Box testing – is a method where the internal structure of the software is known to the tester. Testers in White Box testing have access to the source code of application under test. White box testing is used in earlier stages of software development.
- Stress testing – is a type of intense testing witch is used to determine the stability of a given system or entity, or breaking points or safe usage limits of application under test. This testing is done in the final stages of software development.
- Exploratory testing – is an approach in testing used to discover unknown issues during and after the software development process. Exploratory tests must be run manually. Experienced QA testers or developers are generally required for exploratory testing.
- Regression testing – is a testing type that is used to make sure that a change in software like adding new features or correcting the discovered defects didn't broke or corrupt existing functionality. Main goal of Regression testing is to catch bugs that may have been accidentally introduced into a new build or release candidate, end to ensure that previously eradicated bugs are permanently removed.

The most commonly used Tools and Frameworks by Zesium QA Team in software testing process:

- JIRA – is a project management tool used by agile teams for project management, issue tracking, bug tracking, team communication and reporting in software development.,
- JMeter – is an open source software designed for testing Web applications. Usually JMeter is used for Load testing and measuring performance of web applications. JMeter can simulate heavy load on a server, group of servers or network, to test its strength or to analyze overall performance under different load types.,
- Monkey Runner – is a tool primarily designed to test applications and devices at the functional/framework level and for running unit test suites. Monkey Runner is tool that provides API for writing programs that control Android device or emulator from outside of Android code. With Monkey Runner is possible to write test script that install Android application, runs it, sends keystrokes to it, takes screenshots of user interface and stores it.
- Espresso – is a testing framework for Android. Espresso is used for creating user interface tests for Android devices,
- Layout inspector – is used in Android Studio for comparison application layout with design mockups and prepared design resources. Its allows tester to display a magnified view of app under test and to examine details of its layout at runtime,
- Selenium – is a software testing framework for web applications. Selenium provides a playback tool for generating test scripts. It supports different programming languages. Tests created with Selenium can be run against most modern web browsers.,
- Katalon Studio – is an automated testing tool for testing Web applications, Mobile applications (Android and iOS) and for API testing on all operating systems. Its based on Selenium and Appium and can be integrated with JIRA, GIT and Jenkins with appropriate plugins.

Final remarks on work that Zesium's QA Team does:

Main goal or testing process is to reduce the risks appearing during software development. Therefore, it's very important to find as many bugs as possible so that they can be brought down to acceptable level. With this objective reached Test Team can provide all necessary information about software product that is being developed by Zesium.

References:

- [1] Daniel Knott, “Hands on Mobile App Testing”, Crawfordville, United States, 2015.
- [2] Kshirasager N., Priyadarshi T., “Software Testing and Quality Assurance Theory and Practice”, New Jersey, United States, 2008.
- [3] “Software Testing Tutorial”, www.guru99.com
- [4] “QA Training”, www.softwaretestinghelp.com