Share on your Social Media

Software Testing Tutorial for Beginners

Published On: October 8, 2024

Software Testing Tutorial for Beginners

Software testing is the process of assessing and confirming the functionality of an application or software product. Learn effective testing that has the advantage of reducing errors through this software testing tutorial.

[Download Software Testing Tutorial PDF](#)

Introduction to Software Testing

Software testing is a great approach for freshers to enter into IT jobs because it's a reasonably straightforward job to get into and a pretty simple position to do. We cover the following in this software testing tutorial:

- Overview of Software Testing
- Software Testing Principles
- Types of Software Testing
- Software Testing Levels
- Software Testing Life Cycle
- Test Environment
- Test Strategy
- Testing Techniques
- Advantages of Software Testing
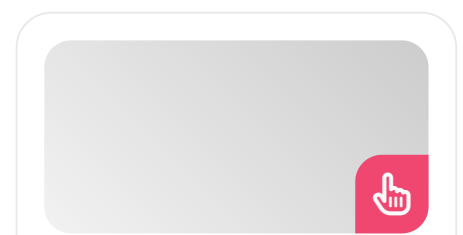
Related Courses at SLA

- Software Testing Training in OMR
- Software Testing Training in Chennai
- Software Testing Online Training

Related Posts

Overview of Software Testing

Testing is essentially a group of techniques used to assess whether an application is appropriate for usage with a script that has been pre-programmed. Finding application bugs is the main objective of testing so that they can be found and fixed. Achieve your dream career easily by enrolling in your desired course from our wide range of **software training courses**.

- To find any errors or bugs in the software product's code, software testing entails running a program.
- This procedure considers every facet of the software, such as its usability, scalability, portability, reusability, and dependability.
- Software testing's primary objective is to confirm that the system and all of its parts function accurately and meet the requirements as stated.

Manual Testing

Verifying an application's functionality following customer requirements by manual testing eliminates the need for automated methods.

To perform manual testing on any application, we don't need to be highly versed in testing tools; instead, we just need to have a solid understanding of the product to swiftly write the test document.

Three categories of manual testing are as follows:

- **White box testing:** This method is used to check the software's internal logic, flow, and coding structure.
  - It is carried out by testers with technical know-how and a thorough comprehension of the source code.
  - It offers a comprehensive testing method and eliminates unnecessary lines of code to optimize the code.

- This form of testing detects flaws from the early phases of the SDLC.
- It finds the bugs that other testing methodologies fail to pick up on.
- **Black box testing:** When testing if a piece of software's features and functionalities meet user requirements, testers who are not familiar with the inner workings of the program utilize this technique.
  - Even testers without programming experience can complete it.
  - It is best suited for testing large systems and is done from the perspective of the client.
  - The detection of differences in the functional requirements is another benefit of this kind of testing.
- **Grey box testing:** The concepts of the white box testing and black box testing methodologies are combined in this method.
  - The testers' inside knowledge of the code is limited.
  - Complete access to the internal data structures and code is granted while creating the test cases.
  - It is presented from the perspective of the client and raises the software's general caliber.

[Software Testing Interview Questions and Answers](#)

Automation Testing

Any manual test cases can be turned into test scripts by using automation tools or an automation testing programming language.

With automated testing, we can execute our tests more quickly because it eliminates the requirement for human labor. Test scripts must be written and executed.

Introduction A Software Tester, or QA Tester, ensures software applications function correctly by planning and...

Explore what our **software testing and QA architect job seeker program** has in store for you.

Software Testing Principles

Software testing is the process of using a program or piece of software to look for bugs or errors.

To make sure that the end product is flawless, there are rules that we must follow when testing software or apps. The test engineers can test the program more effectively as a result, saving them time and effort.

Testing Reveals the Existence of Errors

- Software testing aims to make the program fail.
- Software testing lowers the likelihood of flaws.

- Software testing just discusses the existence of defects and leaves out any discussion of their absence.
- Software testing can confirm the existence of problems in the program, but it cannot demonstrate that the program is flawless.
- Software can never be completely bug-free, not even with numerous tests.
- While not all flaws may be eliminated, testing can help minimize their number.

Complete Testing is Not Feasible

Extensive testing is the process of examining the program's functionality under all potential input scenarios and preconditions.

- Since the program can never test at every test case, exhaustive testing is not possible.
- It is limited in what it can test, and it assumes that the program is accurate and will generate the right result in each test scenario.
- It would be excessively expensive, time-consuming, and unfeasible to have the program test every test case.

Early Testing

Early test activity must be initiated to identify the software defect.

- The cost of a problem found early in the SDLC process will be extremely low.
- Program testing will begin at the beginning, that is, during the requirement analysis process, to improve program performance.

Defect Clustering

The majority of flaws in a project may be present in a small number of modules. Software testing's Pareto Principle argues that 20% of modules account for 80% of software faults.

Pesticide Paradox

It will not be possible to discover new bugs by repeatedly running the same test cases. Therefore, to identify new bugs, it is required to evaluate the test cases and add or update test cases.

Context-Based Testing

The software development setting influences the testing methodology. Different software types require different kinds of testing to be done. For example, testing an Android application is not the same as testing an e-commerce website. Accelerate your career with **API testing training**.

Absence of Errors

Software that satisfies user requirements but has 99 percent of its bugs fixed is not usable. Ensuring that software is 99.9% bug-free and meets all customer criteria is equally important.

Software Testing Course Syllabus PDF

Types of Software Testing

By conducting both functional and non-functional

testing, you may improve software performance and reliability while guaranteeing user happiness. Here are the various software testing techniques, such as automated and manual testing, to enhance quality control:

Three major categories can be used to categorize software testing:

- **Functional Testing:** Software systems are validated against the functional requirements through a sort of testing called functional testing. This is done to see if the application is operating following the functional requirements of the program.

Unit testing, integration testing, system testing, smoke testing, and other forms of functional testing are among them.

- **Non-functional Testing:** Software that tests for non-functional criteria such as performance, scalability, portability, stress, etc. is known as non-functional testing.

Performance testing, stress testing, usability testing, and other forms of non-functional testing are among them.

- **Maintenance Testing:** Updating, altering, and upgrading the software to meet the demands of the user is known as maintenance testing.

Regression testing is used to ensure that recent code modifications haven't negatively impacted any previously functional software components.

Various Software Testing Levels

Four main levels can be distinguished in software-level testing:

- **Unit Testing:** Individual units or components of a software or system are tested at the unit testing stage of the software testing process. Verifying that every software unit operates as

intended is the goal.

- **Integration Testing:** This stage of the software testing procedure involves combining and testing individual modules collectively. This level of testing aims to identify errors in the way integrated components interact with one another.
- **System Testing:** It is a stage of the software testing process that involves testing an integrated, whole system. This test's objective is to determine whether the system complies with the given specifications.
- **Acceptance Testing:** A system is examined for acceptability at the acceptance testing stage of the software testing process. This test's objective is to determine whether the system satisfies the business requirements and whether it can be delivered.

Our **Appium testing training in Chennai** helps you find your dream career in IT.

Software Testing Life Cycle

The traditional procedure known as the Software Testing Life Cycle (STLC) is divided into several stages. A team follows it to complete testing for high-quality software at every level.

The Software Testing Life Cycle (STLC) is a subset of the Software Development Life Cycle (SDLC) and is restricted to software development testing tasks.

Features of STLC

The following is a list of STLC's features:

- The foundational phase of the SDLC is the STLC.
- STLC commences from the time needs are shared from the client's perspective.
- A systematic technique for achieving software quality is called STLC.

Software Tester Salary in Chennai

Different Stages of STLC

The following is a list of the several STLC stages.

Requirement Analysis

- The clients discuss the needs with the team during the STLC's requirement analysis phase.
- After reviewing and analyzing these criteria, the testing team develops a plan and strategy for the testing procedure.
- A well-thought-out test strategy and plan must be in place to guarantee high-quality software.

Test Planning

- Project managers and leads collaborate to create a test plan during the STLC's test planning phase.
- The items in scope, out of scope, budget, strategy, methodologies, team member tasks, estimations, number of testing cycles, team size, tools, extra resources, percentage of automated and manual testing efforts, etc. are all covered in these documents.

Test Strategizing

- Project managers and leads collaborate to create a strategy document during the STLC's test strategizing phase.
- Typically, a test strategy document is created and then left unmodified until the project is finished.
- During the test strategy phase, the protocols and techniques to be used for testing are primarily decided.

Environment Identification

- The proper environment in which software is to be verified is identified during the STLC's environment identification step.
- Before beginning the testing process, a routine sanity check should be performed to make

sure the software satisfies the minimal requirements for testing.

Test Data Identification

The STLC's test data identification step identifies the correct inputs and data that need software verification.

Designing of the Test Cases

- All potential use cases and situations (positive, negative, and edge) found in the requirements are turned into test cases during the test case designing stage of the STLC.
- They undergo a thorough examination and obtain approvals from the relevant parties to guarantee test coverage and software quality.
- Additionally, throughout this phase, RTM and all other conventional test procedures and principles are adhered to.

Test Cases Execution

During the STLC's test case execution phase, every test case is run to see if the software functions as intended and to find any potential faults.

Defect Logging

All of the flaws found during the execution phase of the STLC are logged, prioritized, corrected, and properly closed during the defect logging phase.

Test Closure

All of the comprehensive reports on test execution (number of test cases, passed, failed, percentage of failure, impacted modules, defects, etc.) and metrics are recorded and distributed to all project stakeholders during the test closure phase of the STLC.

Entry and Exit Criteria in STLC

- The requirements that must be met before the testing process can start are known as entry

criteria.

- The tasks that must be finished before the software can advance to the following phases of the SDLC are known as exit criteria.

Every testing phase and level in the STLC has its own set of entry and exit criteria. Enrich your skills with our **JMeter testing training in Chennai**.

Test Environments

The test team uses a test environment to validate the program and find and fix faults. Testing various data combinations before the software is shipped to the customer aids in certifying the product's quality.

Various Types of Test Environments

The testing environments that differ are mentioned below:

Security Environment

To ensure that software is safe and does not jeopardize user confidentiality or integrity, a security test environment is utilized.

Performance Environment

To ensure that the program meets client criteria for stability, usability, responsiveness, performance, etc., a performance test environment is utilized.

It should be equipped with the necessary RAM, data, and other parameters to run the performance tests.

Integration Environment

To connect several software clusters and ensure that the program is functioning properly by putting all the parts together, an integration test environment is utilized. It ought to be near the production setting.

Learn and level up your career with our **load runner testing course in Chennai**.

Configuring Test Environment

Below is a list of steps to follow when setting up a test environment.

- Configure the test server so that it is not necessary to test the software locally alone.
- For the program to function properly when developers, testers, and other team members work simultaneously, network access via CAN, LAN, wireless, etc. must be present.
- The testing team needs to set up the various systems and browsers.
- It is necessary to set up a defect-tracking system in the testing environment.
- It is necessary to provide a copy of the production data in the testing environment. In the test environment, a disorganized and anonymous form of production data is used to protect user privacy.

The Best Ways to Configure a Testing Environment

- Verify that the test environment is configured with the necessary hardware and software that are compatible with the application being tested.
- Check that the debugger, defect tracking, and automation test tool are operational in the testing environment.
- Verify the test environment's availability of the test data.

Test Strategy

Early in the Software Testing Life Cycle (STLC) process, a test strategy document is prepared. It is an essential step in the process of creating test documentation.

It primarily focuses on the different kinds, degrees, and testing strategies that should be used to guarantee the quality of the software.

Gain expertise with software testing through our **manual testing training courses in Chennai**.

Components of a Test Strategy

The following is a list of test strategy components:

Overview and Scope of the Project

- This is the test strategy document's first section. It includes information about the project's scope, intended audience, and test strategy document approvers.
- The project deadlines specified in the test plan also outline the testing procedures, phases, and timetables that will be adhered to.

Test Methodologies

- The test strategy document ends with this section. It outlines the scope of testing, as well as the levels, procedures, approaches, and duties and obligations of each team member.
- It also describes the rationale behind each kind of testing, when to begin, what automation tools to employ, and other things.

Test Environment

- This is an additional section of the exam strategy paper. It explains how to generate, obtain, duplicate, and recover test environments, setups, and test data.
- It also details the gear, software, user credentials, and other requirements for the test environment, such as the number of users.

Testing Tools

- The tools that will be utilized for test management and automation are covered in this additional section of the test strategy document.
- It also describes the testing methodologies and equipment needed for security, load, and performance testing.

- It makes it apparent whether tools are licensed or open-source, as well as how many people are utilizing them for the project.

Release Control

To make release management and test execution a methodical procedure, this section has been included in the test strategy document.

Risk Analysis

- This portion of the test plan document contains a list of all possible dangers and risks that could arise when testing the software.
- It enumerates all dangers, risks, and issues, along with backup and emergency plans for each.

Reviews and Approvals

- The test strategy document ends with this section.
- The development, business, and management teams should review each element of the test strategy document.
- The date, names of the approver and reviewer, comments, revision history, and other details should all be accurate.

Learn our **QTP course in Chennai** to kickstart your software testing career.

Software Testing Project Ideas

Testing Techniques

The following is a list of the various software testing methodologies.

- **Static testing:** This method finds software issues without actually executing the code. It identifies problems right at the start of the SDLC.
- **Dynamic testing** involves running the code to

confirm if the program is dynamic.

Static Testing

The following is a list of the various static testing methods:

- **Walkthrough:** The code author leads this meeting, which is essentially a review session.
- **Technical Review:** An evaluation of every aspect of the software is conducted critically.
- **Inspection:** The purpose of the inspection is to examine the software and find any flaws.
- **Data Flow:** This is a crucial component of white box testing. It obtains the paths where the new variables are defined and then used by using the control flow graph.
- **Control Flow:** The software code is structurally tested in this step.

Dynamic Testing

The following is a list of the various dynamic testing methods.

- **Structure-Based Testing:** Decision coverage testing, statement coverage testing, condition coverage testing, and multiple coverage testing are subcategories of structure-based testing.
- **Experience-Based Testing:** Ad hoc testing and error guessing are two subsets of experience-based testing.
- **Specification-Based Testing:** This type of testing is further subdivided into state transitions, decision tables, equivalency partitioning, boundary value analysis, and use case testing.

Enhance your career with our **Selenium training in Chennai**.

Advantages of Software Testing

Here are the advantages of software testing:

- **Customer Satisfaction:** By guaranteeing a flawless application, software testing contributes to the development of consumer pleasure and trust.
- **Cost Effective:** The maintenance area saves money by reducing too many folds through application testing.
- **Quality Product:** Compatibility testing is required if an application functions well with other related apps for it to meet product standards.
- **Low Failure:** Software testing increases the stability and dependability of an application.
- **Bug-Free Application:** An application that functions properly and without any hiccups, errors, or defects is said to be bug-free.
- **Security:** The development team works to add several levels of protection to the program, while the testing team uses security testing to find flaws.
- **Easy Recovery:** Software testing helps in finding the recovery rate of an application with the overall time it is required to recover.
- **Accelerate the Development Process:** To inform the development team where a flaw exists, software testing work is conducted in tandem with them.
- **Early Defect Detection:** Finding flaws is the main goal of software testing.
- **Reliable Product:** By subjecting an application to performance, security, and other testing methods, software testing improves the application's reliability.

Software Testing Online Training

Conclusion

We hope this software testing tutorial helps you get a fundamental understanding. Through applicable projects and real-world scenarios, gain practical learning experience with the newest testing methodologies, automation technologies, and

industry best practices through our **software testing training in Chennai**.

Share on your Social Media

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Courses

Python

Software Testing

Navigation

About Us

Blog Posts

Careers

Contact

Placement Training

Corporate Training

Hire With Us
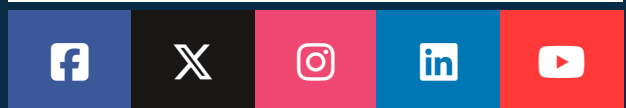
Job Seekers

SLA's Recently Placed Students

Reviews

Sitemap

Important Links

Disclaimer

Privacy Policy

Terms and Conditions

Social Media Links

Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

Review Sources

Google

Trustpilot

Glassdoor

Mouthshut

Sulekha

Justdial

Ambitionbox

Indeed

Software Suggest

Sitejabber