SAP Testing

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

Many Organizations implement SAP ERP to manage their business operations efficiently. SAP Testing is about testing the functionality of various SAP modules and to ensure that they perform as per the configuration. This tutorial explains the basics of SAP testing.

Audience

This tutorial is meant for all those readers who want to learn complete end-to-end testing and validation of all SAP modules in SAP ERP environment.

Prerequisites

You need to have a basic understanding of the concepts of enterprise resource planning in order to make the most of this tutorial.

Disclaimer & Copyright

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

1. SAP Testing – Overview

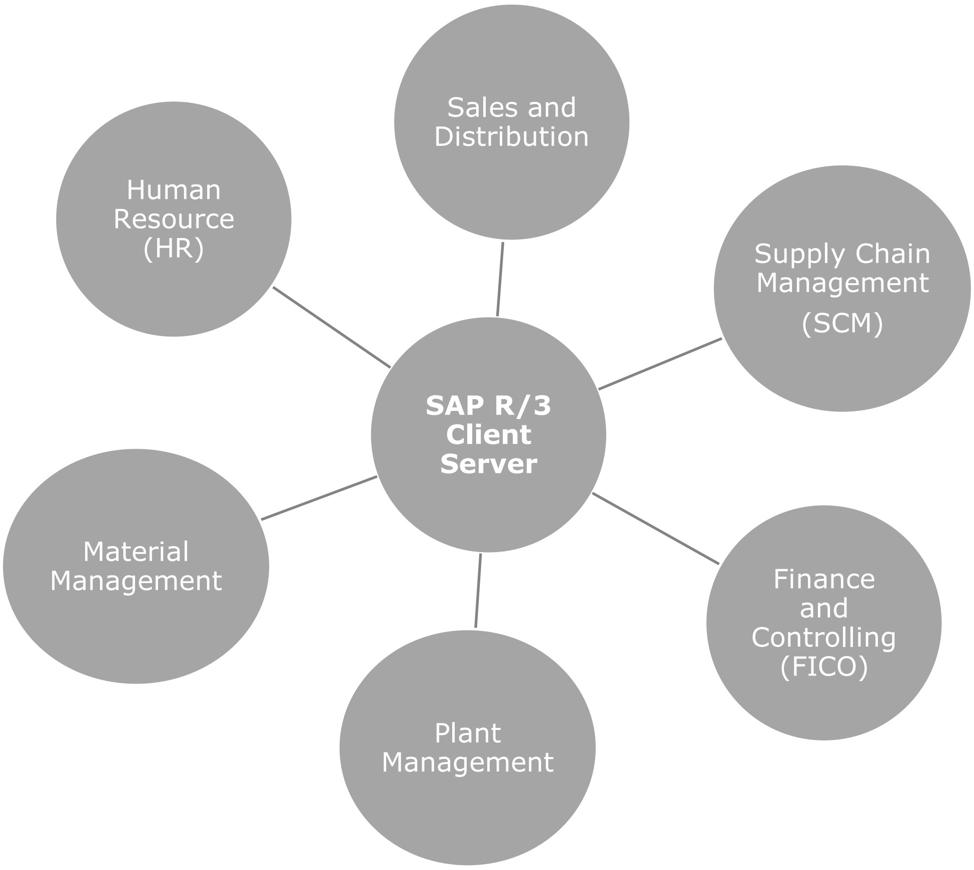
Many organizations implement SAP ERP (Enterprise Resource Planning) to manage their business operations and adapt according to new market challenges. SAP R/3 is an integrated ERP software that allows organizations to manage their business efficiently. Organizations can reduce the cost to run their operations by using SAP R/3 ERP packages.

SAP R/3 also allows customers to interact with different databases to run different applications with the help of a user-friendly GUI. The SAP R/3 system is divided into different modules to cover the functionality of different business operations in an organization.

The most common SAP R/3 modules are:

- SAP Material Management
- SAP Financial Accounting and Controlling
- SAP Sales and Distribution
- SAP Human Resource
- SAP Supply Chain Management
- SAP Plant Management

SAP Testing is about testing the functionality of these modules and to ensure that they perform as per the configuration.



A SAP system undergoes various changes like patch management and fixes, new module implementations, and various other configuration changes. All these modifications raise a need for Regression testing that is to be performed in SAP environments. SAP testing automation tools like SAP Test Acceleration and Optimization tools can be used for this purpose.

SAP TAO is an automation tool to generate test cases for end-to-end scenarios for SAP applications. Apart from this, there are various other Automation testing tools for SAP testing like HP QTP, and ECATT, etc. that can be used.

Importance of Testing

Here is a list of key reasons why SAP testing is performed and why it is an important function in the growth of an organization:

- **System Validation:** SAP Testing involves complete end-to-end testing and validation of all SAP modules in SAP ERP environment.

- **Quality and Revenue:** SAP Testing is an output-based testing and not like conventional testing methods which are input-based. It ensures the quality of SAP system and also focuses on revenue and cost of the organization.

- **Cost and Predictability:** SAP Testing involves reducing the SAP development costs and improve predictability.

- **Compliance Requirement:** SAP Testing ensures that the SAP implementation is meeting the new compliance requirements in a specific organization and all the modules are working as per the expected configuration.

- **New Implementation and Configuration Changes:** There are different types of changes implemented in a SAP system, like patches and fixes, new implementation, configurational changes. Therefore, SAP testing ensures that all the modules are performing as per requirement in this dynamic system environment.

- **Integration:** SAP testing is performed to test various reports, data flows and work flows, GUI forms, etc. It is used to check system integration between different modules. For example, if an order posting is done that requires an action in Sales and Distribution, MM and FICO, then SAP testing checks the integration between these systems.

- **Performance:** It is also used to ensure if the system will be able to meet the Service Level agreements, time taken by system to perform specific actions, performance of the system, etc.

2. SAP Testing – Types of Testing

There are different testing methods that can be used to test the functionality of a software, system, or an application.

The most common testing techniques are:

- **Unit Testing**: It is a type of white-box testing that involves testing a single unit or group of units.

- **Integration Testing**: In this testing, multiple systems are combined together to test the output of the integrated system.

- **Functional Testing:** It checks the functionality of each module as per the desired result.

- **Usability Testing:** It checks the ease of use of an application or a system. It checks how easy it would be for a new user to use an application or to understand the system.

- **Acceptance Testing:** Acceptance testing is performed to test if a system meets the user requirement and whether to accept the application or system.

- **System Testing:** Entire system is tested as per the requirement and specification.

- **Stress Testing:** In this testing, the system is put into stress beyond its specification to check when it fails.

- **Performance Testing:** This testing is performed to check if the system meets the performance requirement.

- **Regression Testing:** It includes testing the full application or system for the modifications.

- **Beta Testing:** The aim of beta testing is to cover unexpected errors. It falls under the class of black-box testing. It is performed by releasing the pre-version of the final product, called Beta.

- **Database Testing:** Database testing is used to test the data in the database. It is performed using SQL statements.

- **ETL Testing:** ETL testing is performed to ensure if data is correctly extracted, transformed, and loaded from a source system to a target system.

3. SAP Testing – Manual and Automation Testing

Manual Testing

Manual testing means you are testing a software manually without using any automated tools or any script. In this type of testing, the tester takes over the role of an end-user and tests the software to identify bugs or any unexpected behavior.

There are different stages of a manual testing. They are: unit testing, integration testing, system testing, and user acceptance testing.

Various test plans, test cases, or test scenarios are used by a manual tester to ensure the completeness of testing. Manual testing can also be called **exploratory testing** because the testers explore the software to identify errors in it manually.

Automation Testing

In Automation testing, the tester writes the scripts and uses software tools to test the product. This process involves the automation of a manual process. Automation testing includes re-running the test-cases multiple times that were performed manually.

Automation testing is also used to test the application from load, performance, and stress purpose. It is used to increase the coverage of test. Automation testing improves the accuracy and saves time and money in comparison to manual testing.

Software Testing Tools

The following tools can be used for Automation testing:

- HP Quick Test Professional (QTP)
- Selenium
- SAP TAO
- ECATT
- IBM Rational Functional Tester
- SilkTest
- TestComplete
- Testing Anywhere
- WinRunner
- LaodRunner
- Visual Studio Test Professional
- WATIR

4. SAP Testing – SDLC

Software Development Life Cycle determines the series of steps to be performed to develop an application or the efficiency of a software. In this chapter, we will discuss the phases defined in SDLC. Each phase has its own process and deliverables that goes into the next phase.

Feasibility Study

The first stage of SDLC is **requirement gathering**. After the requirements are gathered, the team comes up with a rough plan of software process. At this step, the team analyzes if a software can be made to fulfill all the requirements of the user. It is found out if the project is financially, practically, and technologically feasible for the organization to take up. There are many algorithms available, which help the developers to conclude the feasibility of a software project.

System Analysis

At this step, the developers decide a roadmap of their plan and try to bring up the best software model suitable for the project. System analysis includes understanding of software product limitations, learning system-related problems or changes to be done in the existing systems, identifying and addressing the impact of the project on the organization and personnel etc. The project team analyzes the scope of the project and plans the schedule and resources accordingly.

Software Design

The next step is to bring the whole knowledge of requirements and analysis on to the desk and design the software product. The inputs from the users and the information gathered in the requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; **logical design** and **physical design**. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams and in some cases pseudocodes.

Coding

This step is also known as **programming phase**. The implementation of software design starts in terms of writing the program code in a suitable programming language and developing error-free executable programs efficiently.

Testing

An estimate says that 50% of the whole software development process should be tested. Errors may ruin the software from critical level to its own removal. Software testing is done while coding by the developers and thorough testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, in-house testing and testing the product at user's end. Early discovery of errors and their remedy is the key to developing a reliable software.

Integration

Software may need to be integrated with the libraries, databases, and other program(s). This stage of SDLC deals with the integration of the software with outer world entities.

Implementation

Implementation or deployment means installing the software on user machines. At times, the software needs post-installation configurations at the user's end. Software is tested for portability and adaptability and integration related issues are solved during implementation.

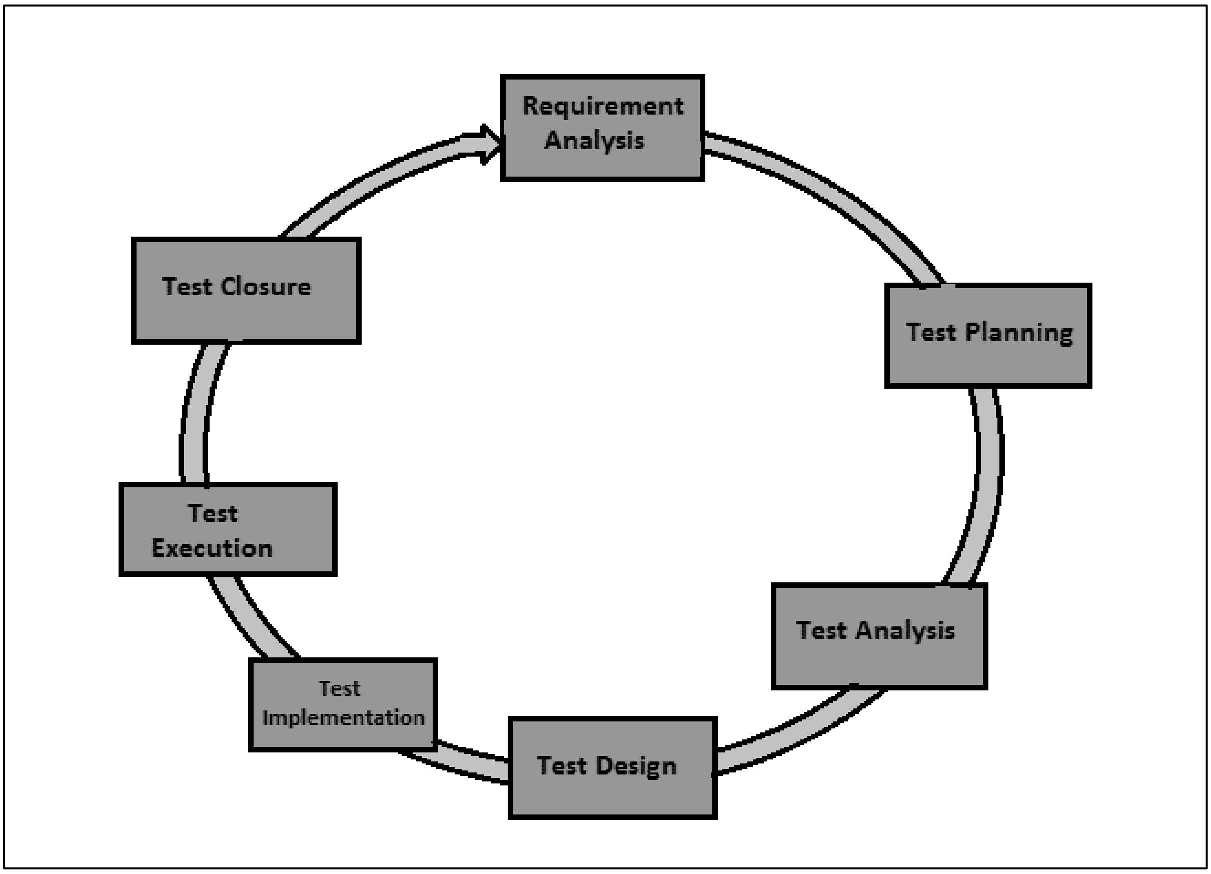
Software Testing Life Cycle

Software Testing Life Cycle (STLC) consists of all the steps that are performed in a specific way to ensure that quality goals are met and each step has specific goals and deliverables.

STLC is used to improve the quality of a software product and to make it capable to meet the business requirements to achieve certain goals.

The different stages that come under Software Testing Life Cycle are as follows:

- Requirements phase
- Test Planning
- Test Analysis
- Test Design Phase
- Test Implementation
- Test Execution Phase
- Test Closure Phase



Requirement Phase

This is the first phase of Software Testing Life Cycle. During this phase, the tester's job is to analyze the requirements. There are various methods for Requirement Analysis like conducting brainstorming sessions with business people, team members, and try to find out whether the requirements are testable or not.

This phase determines the scope of the testing. If a testing team finds any features that can't be tested, then that should be communicated to the client.

Test Planning

In this phase, the tester identifies the activities and resources which would help to meet the testing objectives.

Various metrics are defined and there are methods available to determine and track those metrics. Test planning also includes identifying key performance indicators for testing evaluation.

Test Analysis

This phase determines the guidelines that has to be tested. It includes identifying the test conditions using the requirements document, any risks involved, and other test criteria.

Various factors are used to find out the test conditions:

- Product Complexity
- Depth of Testing
- Risk Involved
- Skills Required
- Knowledge of testing team members
- Test management
- Availability of the stakeholders

Test conditions should be written in a detailed way.

Let us take an **example**. For a website selling products online, a test condition is that a customer should be able to make an online payment. You can add detailed conditions like, payment should be feasible using Credit card, NEFT transfer, debit card or net banking.

The advantage of writing the detailed test condition is that it increases the scope of testing because test-cases are normally written on the basis of the test condition. It allows to write more detailed test cases. It also helps in determining the condition of when to stop the testing of a software product.

Test Design Phase

This phase determines how the tests are performed.

- Break down the test conditions into multiple sub-conditions to increase its coverage.
- Get the test data.
- Set up the test environment.
- Get the requirement traceability metrics.
- Create the test coverage metrics.

Test Implementation Phase

This phase includes the creation of detailed test-cases as per the test conditions and metrics defined.

- Prioritize the test case.
- Test-case to be used for Regression.
- Ensure the correctness of the test-cases.
- Sign off of the test-cases before the actual execution starts.

Test Execution Phase

This phase of Software Testing Life Cycle involves actual execution of test-cases.

- Execute the test-cases.
- Log the defects.
- Check traceability metrics to track progress.

Test Closure

This phase includes checking for the completion of the test.

- Check if all the test-cases are executed and opened defects.
- Note down the lessons learnt.
- Close the Testing phase.