## IMG_256What is System Testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

## What is System Under Test (SUT)?

System under test (SUT) refers to a system that is being validated by the testers. The terminology is also known as application under test.

The System Under Test (SUT) also corresponds to a software that is matured and has gone through unit and integration testing.

## What do you verify in System Testing?

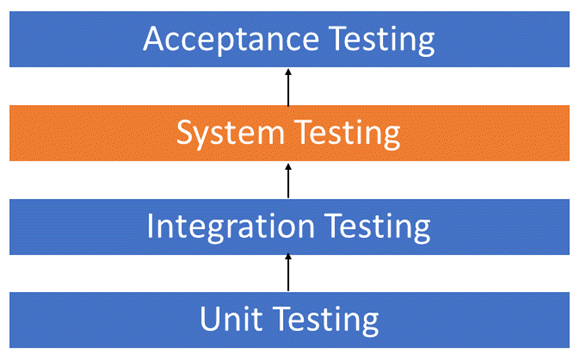
System Testing involves testing the software code for following

## What do you verify in System Testing?

* Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.
* Verify thorough testing of every input in the application to check for desired outputs.
* Testing of the user's experience with the application.

That is a very basic description of what is involved in system testing. You need to build detailed test cases and test suites that test each aspect of the application as seen from the outside without looking at the actual source code.

## Software Testing Hierarchy

[](https://www.guru99.com/images/1/System-Testing.png)

As with almost any software engineering process, software testing has a prescribed order in which things should be done. The following is a list of software testing categories arranged in chronological order. These are the steps taken to fully test new software in preparation for marketing it:

* Unit testing performed on each module or block of code during development. [Unit Testing](https://www.guru99.com/unit-testing-guide.html) is normally done by the programmer who writes the code.
* Integration testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effect on the entire program model.
* System testing done by a professional testing agent on the completed software product before it is introduced to the market.
* Acceptance testing - beta testing of the product done by the actual end users.

## Different Types of System Testing

There are more than 50 types of System Testing. For an exhaustive list of software testing types click [here](https://www.guru99.com/types-of-software-testing.html). Below we have listed types of system testing a large software development company would typically use

1. [Usability Testing](https://www.guru99.com/usability-testing-tutorial.html)- mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives
2. [Load Testing](https://www.guru99.com/load-testing-tutorial.html)- is necessary to know that a software solution will perform under real-life loads.
3. [Regression Testing](https://www.guru99.com/regression-testing.html)- involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.
4. Recovery testing - is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.
5. Migration testing- is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.
6. Functional Testing - Also known as functional completeness testing, [Functional Testing](https://www.guru99.com/functional-testing.html) involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.
7. Hardware/Software Testing - IBM refers to Hardware/Software testing as "HW/SW Testing". This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing.

## What Types of System Testing Should Testers Use?

There are over 50 different types of system testing. The specific types used by a tester depend on several variables. Those variables include:

* Who the tester works for - This is a major factor in determining the types of system testing a tester will use. Methods used by large companies are different than that used by medium and small companies.
* Time available for testing - Ultimately, all 50 testing types could be used. Time is often what limits us to using only the types that are most relevant for the software project.
* Resources available to the tester - Of course some testers will not have the necessary resources to conduct a testing type. For example, if you are a tester working for a large software development firm, you are likely to have expensive automated testing software not available to others.
* Software Tester's Education- There is a certain learning curve for each type of software testing available. To use some of the software involved, a tester has to learn how to use it.
* Testing Budget - Money becomes a factor not just for smaller companies and individual software developers but large companies as well.

## What is UAT?

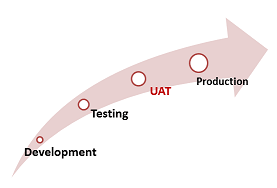
**User Acceptance Testing (UAT)** is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

Purpose of UAT

The main **Purpose of UAT** is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

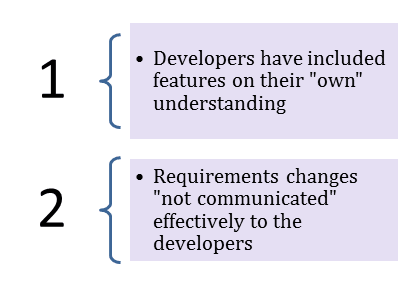
## Who Performs UAT?

* Client
* End users

[](https://www.guru99.com/images/5-2015/050115_0701_WhatisUserA2.png)

Need of User Acceptance Testing

**Need of User Acceptance Testing** arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

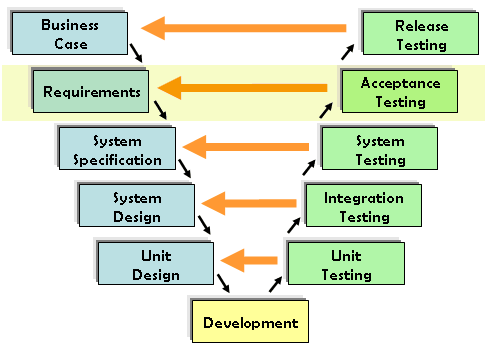
[](https://www.guru99.com/images/5-2015/050115_0701_WhatisUserA3.png)

Developers code software based on requirements document which is their "own" understanding of the requirements and **may not actually be what the client needs from the software**.

* Requirements changes during the course of the project may not be communicated effectively to the developers.

## Acceptance Testing and V-Model

In VModel, User acceptance testing corresponds to the requirement phase of the Software Development life cycle(SDLC).

[](https://www.guru99.com/images/5-2015/050115_0701_WhatisUserA4.png)

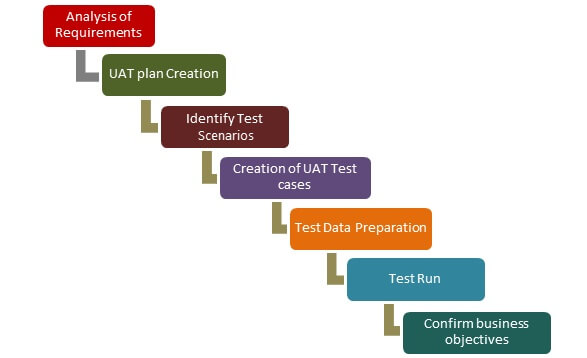
### Prerequisites of User Acceptance Testing:

Following are the entry criteria for User Acceptance Testing:

* Business Requirements must be available.
* Application Code should be fully developed
* Unit Testing, Integration Testing & System Testing should be completed
* No Showstoppers, High, Medium defects in System Integration Test Phase -
* Only Cosmetic error is acceptable before UAT
* Regression Testing should be completed with no major defects
* All the reported defects should be fixed and tested before UAT
* Traceability matrix for all testing should be completed
* UAT Environment must be ready
* Sign off mail or communication from System Testing Team that the system is ready for UAT execution

## How to do UAT Testing

UAT is done by the intended users of the system or software. This type of Software Testing usually happens at the client location which is known as Beta Testing. Once Entry criteria for UAT are satisfied, following are the tasks need to be performed by the testers:

[](https://www.guru99.com/images/5-2015/user_accptance_testing.png)UAT Process

* Analysis of Business Requirements
* Creation of UAT test plan
* Identify Test Scenarios
* Create UAT Test Cases
* Preparation of Test Data(Production like Data)
* Run the Test cases
* Record the Results
* Confirm business objectives

### Step 1) Analysis of Business Requirements

One of the most important activities in the UAT is to identify and develop test scenarios. These test scenarios are derived from the following documents:

* Project Charter
* Business Use Cases
* Process Flow Diagrams
* Business Requirements Document(BRD)
* System Requirements Specification(SRS)

### Step 2) Creation of UAT Plan:

The UAT test plan outlines the strategy that will be used to verify and ensure an application meets its business requirements. It documents entry and **exit criteria for UAT, Test scenarios and test cases approach and timelines of testing**.

### Step 3) Identify Test Scenarios and Test Cases:

Identify the test scenarios with respect to high-level business process and create test cases with clear test steps. Test Cases should sufficiently cover most of the UAT scenarios. Business Use cases are input for creating the test cases.

### Step 4) Preparation of Test Data:

It is best advised to use live data for UAT. Data should be scrambled for privacy and [security](https://www.guru99.com/ethical-hacking-tutorials.html) reasons. Tester should be familiar with the database flow.

### Step 5) Run and record the results:

Execute test cases and report bugs if any. Re-test bugs once fixed. [Test Management](https://www.guru99.com/test-management.html) tools can be used for execution.

### Step 6) Confirm Business Objectives met:

Business Analysts or UAT Testers needs to send a sign off mail after the UAT testing. After sign-off, the product is good to go for production. Deliverables for UAT testing are Test Plan, UAT Scenarios and Test Cases, Test Results and Defect Log

## Exit criteria for UAT:

Before moving into production, following needs to be considered:

* No critical defects open
* Business process works satisfactorily
* UAT Sign off meeting with all stakeholders

## Qualities of UAT Testers:

[](https://www.guru99.com/images/5-2015/050115_0701_WhatisUserA7.jpg)

UAT Tester should possess good knowledge of the business. He should be independent and think as an **unknown user to the system**. Tester should be Analytical and Lateral thinker and combine all sort of data to make the UAT successful.

Tester or Business Analyst or Subject Matter Experts who understand the business requirements or flows can prepare test and data which are realistic to the business.

## Best Practices:

Following points needs to be considered to make UAT Success:

* Prepare UAT plan early in the project life cycle
* Prepare Checklist before the UAT starts
* Conduct Pre-UAT session during System Testing phase itself
* Set the expectation and define the scope of UAT clearly
* Test End to End business flow and avoid system tests
* Test the system or application with real-world scenarios and data
* Think as an Unknown user to the system
* Perform Usability Testing
* Conduct Feedback session and meeting before moving to production

## UAT Tools

There are several tools in the market used for User acceptance testing and some are listed for reference:

Fitness tool: It is [a java](https://www.guru99.com/java-tutorial.html) tool used as a testing engine. It is easy to create tests and record results in a table. Users of the tool enter the formatted input and tests are created automatically. The tests are then executed and the output is returned back to the user.

[Watir](http://watir.com/) : It is toolkit used to automate browser-based tests during User acceptance testing. Ruby is the programming language used for inter-process communication between ruby and Internet Explorer.

## Some Example Guidelines of UAT

* Most of the times in regular software developing scenarios, UAT is carried out in the QA environment. If there is no staging or UAT environment
* UAT is classified into Beta and Alpha testing but it is not so important when software is developed for a service based industry
* UAT makes more sense when the customer is involved to a greater extent

### Conclusion:

* In Software Engineering, Full form of UAT is User Acceptance Testing.
* In Software Engineering, UAT stands for User Acceptance Testing.
* UAT is one of the many flavors of testing that has emerged over last twenty-five years.
* With UAT, the client can be sure "What to expect" from the product rather than assuming.
* The benefit of UAT is that there will be no surprises when the product is released to the market.