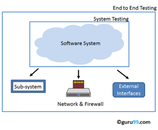
END-To-END Testing Tutorial: What is E2E Testing with Example

End To End Testing

End To End Testing is a software testing method that validates entire software from starting to the end along with its integration with external interfaces. The purpose of end-to-end testing is testing whole software for dependencies, data integrity and communication with other systems, interfaces and databases to exercise complete production like scenario.

Along with the software system, it also validates batch/data processing from other upstream/downstream systems. Hence, the name **"End-to-End"**. End to End Testing is usually executed after functional and System Testing. It uses actual production like data and test environment to simulate real-time settings. End-to-End testing is also called **Chain Testing**.

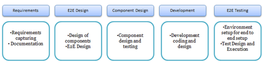


Why End to End Testing?

End To End Testing verifies complete system flow and increases confidence by detecting issues and increasing Test Coverage of subsystems. Modern software systems are complex and interconnected with multiple subsystems that may differ from current systems. The whole system can collapse by failure of any subsystem that is major risk which can be avoided by End-to-End testing.

End to End Testing Process:

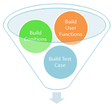
The following diagram gives an overview of the End to End testing process.



The chief activities involved in End to End Testing are -

- Study of an end to end testing requirements
- Test Environment setup and hardware/software requirements
- Describe all the systems and its subsystems processes.
- Description of roles and responsibilities for all the systems
- Testing methodology and standards
- End to end requirements tracking and designing of test cases
- Input and output data for each system

How to create End-to-End Test Cases?



End to End Testing

End to End Testing Design framework consists of three parts

1. Build user functions
2. Build Conditions
3. Build Test Cases

Let's look at them in detail: -

Build User Functions

Following activities should be done as a part of build user functions:

- List down the features of the system and their interconnected components
- List the input data, action and the output data for each feature or function
- Identify the relationships between the functions
- Determine whether the function can be reusable or independent

For example -Consider a scenario where you login into your bank account and transfer some money to another account from some other bank (3rd party sub-system)

1. Login into the banking system
2. Check for the balance amount in the account
3. Transfer some amount from your account to some other bank account (3rd party sub-system)
4. Check your latest account balance
5. Logout of the application

Build Conditions based on User Function

Following activities are performed as a part of build conditions:

- Building a set of conditions for each user function defined
- Conditions include sequence, timing and data conditions

For example -Checking of more conditions like

Login Page

- Invalid User Name and Password
- Checking with valid username and password
- Password strength checking
- Checking of error messages

Balance Amount

- Check the current balance after 24 hours. (If the transfer is sent to a different bank)
- Check for the error message if the transfer amount is greater than the current balance amount

Build a Test Scenario

Building the Test Scenario for the user function defined

In this case,

- Login into the system
- Check of bank balance amount
- Transfer the bank balance amount

Build Multiple Test cases

Build one or more test cases for each scenario defined. Test cases may include each condition as a single test case.

Metrics for End to End testing:

Following are few of many metrics used for End to End Testing.

- **Test Case preparation status:** It gives Test Case preparation progress against planned
- **Weekly Test Progress**- Provides week-wise details of percentage test completion- Failed, not executed & executed against planned for execution tests.
- **Defects Status & Details**- It gives Percentage of open & closed defects by the week. Also, week-wise defects distribution based on severity and priority
- **Environment Availability** -Total number of hours "up" / Total number of hours scheduled per day for testing

End to End Testing Vs System Testing

End to End Testing	System Testing
Validates the software system as well as interconnected sub-systems	Validates just the software system as per the requirements specifications.
It checks the complete end-to-end process flow.	It checks system functionalities and features.
All interfaces, backend systems will be considered for testing	Functional and Non-Functional Testing will be considered for testing
It's executed once System Testing is completed.	It's executed after Integration Testing.
End to End testing involves checking external interfaces which can be complex to automate. Hence Manual Testing is preferred.	Both Manual and Automation can be performed for System Testing

Conclusion

In Software Engineering, End to End Testing is the process verifying a software system along with its sub-systems. The biggest challenge in this testing is to have enough knowledge of the whole system as well as an interconnected sub-system.

Prev Report a Bug Next

Testing Tutorials