We already learned about ***test levels*** and how these test levels contribute to the overall quality of testing. But how do we ensure that what we test in a test level will ensure quality? Each test level (*E.g., System Test*) has a defined set of test cases, so the quality of testing will largely depend on the quality of test cases. So how do we ensure our ***test cases*** are of the highest quality?

It is where ***Testing Techniques*** come into the picture. They are the best practices and scientific techniques that established over a period, to ensure that we write test cases such that they provide maximum coverage. Even then, we keep the test case number minimal. (*Why? Of course to save time and money !*)

How to Choose Right Testing Techniques?

There are several testing techniques, and most of the time, you can apply more than one - So how do you chose the right test technique?

It depends on several project factors. We will discuss the actual techniques in subsequent articles, but let's find out what are the factors that influence this decision.

> ***Type of Component or System:*** *Assume that you are testing the amazon website. Two components need testing - Home page Categories and User Registration. Do you see the difference? One is the navigation of categories to ensure they open up the right pages. Whereas, the other will require you to enter data in the user registration form. Therefore, you will have more variations when you enter data.*
> ***Component or System Complexity:*** *Complexity of the system is another crucial factor in determining the test technique. E.g., a user registration form is a simple component. All we have to do is to figure out what values to enter. Now compare it with another component - The order status, which gives you information about the order. If you are familiar with Amazon or Flipkart, you will realize that order status can be order Created, Shipped, Cancelled, Delivered, or order Returned. However, if you further dig down, there are several variations to this. For instance, the order created; but, the payment failed, Full order shipped, Partial order Shipped, Full order Cancelled, Partial order Cancelled, User Cancellation Vs. System cancellation, etc. The list is big! So we see that complexity will play a significant role in determining test techniques.*
> ***Regulatory Standards:*** *The software needs to adhere to the regulatory standards of each country that will be using it. E.g., A retail site in the UK will need to give a 30-day return option to customers. Whereas, the same website in India might do away with seven days! Knowing these regulatory standards helps to choose the right testing techniques.*
> ***Customer or Contractual Requirements:*** *At times, there could be specific contractual requirements like the site should be responsive, and it should have validation on 20 devices/browser combination. The testing techniques that you will apply in this case will differ from the case where you are testing only two devices.*
> ***Risk Levels:*** *Test techniques depends on the risk level of component or system under test. Assume the amazon website again. Say you are testing a footer component that will have static links like Careers, Contact Us, etc. and compare it with a Login Component. The site may survive if there is an issue in the footer component, but what happens if login throws an error? The risk level would impact the test technique, and as you rightly figured out - the number of test cases will be more for a high-risk component vs. the low-risk component.*

Risk Types: *We have seen risk levels, but what is risk type? Let's retake the login component example. We determined that it's high risk, but what is the impact - probably a financial impact as users will not be able to place orders. Now consider emergency management software. So, if you are in the US, you will realize that they have got the number 911, which takes care of all emergencies (Fire, Medical, theft, etc.). When you call, the software also detects where you are calling from, and alerts the nearest emergency center. Imagine if this software errors out? Can you compare it with Login failure at amazon? As you rightly realized, the risk type is higher in this case as compared to log in on amazon, and hence the testing techniques will differ as well.*

Test Objectives: *Consider a test objective of executing a component-level test vs. a test objective of running a System Test. The amount of testing (and hence the test technique) will significantly differ between a component test and the System test.*

Available Documentation: *How much the requirements have been detailed out also determines the test technique. Consider an example of a user registration form, where you have to enter the First and Last Name. If the rules have been clearly defined (like Name can contain only letters, minimum length is two, and maximum length is 20), then the test cases are focused on this validation. However, if the rules are not defined, then we have to validate these fields with a lot more combinations.*

Testers' knowledge and Skills - *Re-considering our example of order status, the test technique will depend on testers' knowledge of order management. Which means, how much he knows about the various statuses that the order can have, and his expertise to execute the scenarios to get those order statuses.*

Available Tools - *Consider that you have been testing the Sign-in component of the Amazon website(Yes, Amazon seems to be my favorite retailer !). There could be several input variations for the sign-in field. Therefore, manual testing will lead to the application of test techniques to a minimum to get maximum coverage. On the other hand, the availability of automation tools will lead to more test cases. Thereby further increasing the coverage.*

Time and Budget - *End of the day, everything boils down to time and money! Everyone understands that you cannot have 0 bugs in software, and you need to stop testing at some point. So a project with two resources can execute a maximum of say 400 test cases, given the testing needs to complete in 10 days. Now the question arises on how you will write 1000 test cases? We cannot execute them. So you need to apply the right techniques so we can minimize test cases but still keep the coverage high.*

Software Development Life cycle model - *We all remember the old waterfall days, and how the testing team had enough time to write test cases. So we could always write many more test cases to increase coverage. However, in agile, it's all on the go! So we need to be smart in identifying test cases that can execute within the sprints. Additionally, we have to ensure that we don't lose out on coverage.*

Expected Use of the software - *We can take the same example of amazon vs. the emergency management software. The expected usage of software determines risk levels and types. Hence, it will influence the test techniques used.*

Previous experience with using the test techniques on the component or system to be tested - *Well, nothing beats experience! No matter what test techniques you apply, it might still result in failures. So it's crucial to retrospect, and see what worked and what didn't, and accordingly modify the test techniques.*

The types of defects expected in the component of the system - *If we are doing a UI redesign project where only look and feel will change. But existing core functionalities will remain the same Versus a project where we are doing say only DB changes. Both will have a different type of defect. Therefore, the test technique in UI change will focus more on error messages, alignment, etc. In contrast, for DB, it will focus more on the data that passes to DB.*

That was a long list of factors that can influence the test technique that needs to choose. As we see, there is not just one factor. You can have multiple factors within the same project. For instance, a project has a high-risk level, yet it has very little time! So the success of the test highly depends on the effective application of the right testing techniques.