11 Important Software Testing Techniques (Tips Included)

t **testbytes.net**/blog/software-testing-techniques

January 24, 2019

10 SOFTWARE TESTING TECHNIQUES TO LOOK FOR IN 2019 - TIPS INCLUDED

Process
Thursday January 24, 2019

2019 has already arrived with a bang; bringing us the most unique technological solutions to rule over the out-dated ones. One sector which is sure to see new techniques is that of software testing!

New approaches for testing are being introduced in the IT industry due to the emergence of development technologies like DevOps and Agile. Therefore, the need to keep up and transform your own testing techniques according to the new ones is very important.

For this reason we have created a list of the top 11 software testing techniques that you must look forward to this year!

1. IoT Testing

The '*Internet of Things'* is a technology that has brought with it a radical change in the way communication between multiple devices took place traditionally. This has facilitated more and more devices to be connected together.

Therefore, IoT testing is a technique which is used to test the devices based on the IoT technology. This testing process is conducted with:

A few examples of how the **test cases** for IoT testing should be created are:

- Check whether the required IoT devices are successful in data connections and able to get registered to the network.
- Check whether IoT devices are capable of transmitting large amounts of user data.
- Check that only authorized devices are able to connect to the IoT network.
- Check whether the device automatically disconnects or not when the SIM is removed from it.

2. Integrating Manual And Automation Testing Techniques

Since automation testing is not capable of conducting tasks from every area of testing. This is why software testers always benefit most by combining the efforts of both- manual as well as automation testing techniques.

By integrating automation testing while conducting manual tests, your productivity and efficiency can increase by 10 folds! This is why it is important to recognize and segregate the test case that can be automated.

Examples of such test cases may include:

- Test cases that are most often needed
- Test cases which are the most time consuming
- Test cases that have a critical need for accuracy but have scope for human errors

For example, these test cases make up around 25% of your test plan. This will mean a 25% reduction of manual efforts and time consumption!

3. Equivalence Class Testing

This type of testing allows testers to segregate test conditions into partitions. This in turn allows test cases to be designed according to the different input domains created.

The logic behind **Equivalence Class Testing** is that test case for any input value from one domain is equivalent to tests for every other value from that domain.

A simple example:

Consider that the input values are valid between- *5* to *15* and *55* to *65*

Therefore the test conditions created will have 3 equivalence classes-

— to 4 *(invalid)*

5 to 15 *(valid)*

16 to 54 *(invalid)*

55 to 65 *(valid)*

66 to — *(invalid)*

Therefore, you can now select test cases according to one value from each class, i.e., 1, 6, 38, 61 and 90

5. Agile Methodology in Digital Transformation

Organizations are going big on digital transformations in the last few years. This is because it helps in building new strategies as technology as their core to improve customer value and business.

Agile helps in defining business objectives, challenges and its use cases. Therefore, adapting this methodology in digital transformation ensures the rapid generation of quality business solutions. The best practices for Agile Methodology include:

Creating Work Environment

It is important to create the right type of work environment for different 'A*gile teams'.* Such environments must facilitate more efficient information sharing between teams. For example, the presence of the '*burn down chart'*, which tracks the amount of work completed by one team, where everyone can see it is very important.

Scale Agile Initiatives with Frameworks

The most common way an organization integrates Agile is by implementing it into a particular part of their business. This, however, creates a challenge if you aim to implement Agile for bringing about end-to-end changes in your organization.

This is why it is a best practice in Agile to use frameworks to make such activities easier. Examples of such frameworks include- Nexus, LeSS, and Large Scale Scrum.

6. Decision Table Based Testing

This testing technique comes into use mostly for functions that respond to different combinations of inputs. This includes identifying the functionalities where output depends on inputs.

For example, this technique can be used to check whether a '*submit'* button becomes available to the user when all required fields are filled.

Guidelines to create the decision table are:

- List all inputs in rows
- Determine all the rules in columns
- Create all feasible input combinations in the table
- Note the outputs against each input combination at the last row of the table

7. Cloud-Based Testing Technique

Cloud based testing includes the use of cloud-based tools for testing web, installed applications and web. These tools are used to match the environments and user traffic with the real-world.

A few tips to effectively implement the cloud-based testing techniques are:

- **Set Objectives-** This testing proves to be advantageous only if you have a clear objective set for your business needs. It requires cooperation between testers and developers for conducting all tests throughout the SDLC.
- **Creating Test Strategy-** Before transporting your project onto the cloud, determine the tests you need to perform, the time they will take and the risks involved in them. This will help you get an estimated idea of the testing budget.
- **Plan the Infrastructure-** Create test strategies that align with the infrastructure requirements needed for building the test environment.
- **Selecting a Provider-** To select the best provider, compare the quality, reliability and security being offered by them.
- **Determine Level of Access-** To conduct cloud-based testing many testers must have access to the cloud. Therefore, determine how many and who all can have access to it, so as to prevent the generation of additional costs from service overuse.

8. DevOps Testing

'*Development and Operations'* is a development methodology that integrates all development functions including development and operations in the same cycle.

> **Also Read: Software Testing Trends 2019: What To Expect?**

This methodology when used in software testing facilitates the testers to combine test cases, test automation and test designs so as to verify changes within the code while avoiding the product from crashing during the development phase.

A few DevOps testing technique tips include:

- Test execution must be lean
- Test cases required for particular builds need to be developed
- Standardize all the environment requirements for testing and automate the deployments

- Set exit criteria for every run to facilitate the go/no-go decision for production
- The testers must have the ability to use different types of automation techniques over different cross-platform environments.

9. Big Data Testing

Big Data testing helps in ensuring that the quality of data is uncompromising. It's going to be widely used testing technique this year because of the decreasing costs of data storage. In this type of testing, testers verify whether the terabytes of data have been successfully processed using supportive components like commodity cluster or not.

A few examples of the test cases in big data testing are:

- Determine if the correct alter mechanisms, such as Mail on alert, are executed.
- Check whether errors and exceptions are properly displayed with appropriate error messages so that error/exception handling becomes easy.
- Implementing integration testing for complete workflow, from data ingestion till its storage or visualization.
- Performance testing for different parameters of processing random data and monitoring parameters like time taken in execution of particular metrics.

10. Error Guessing

This is a technique where the tester bases of his test cases primarily by guessing the error which can occur in the code. The tester use their past experience to identify the problematic areas in the application.

A few guidelines for this type of testing are:

- Test cases should be made using past experience with similar type of applications
- Keep track of previous error areas
- It is necessary to have an understanding of the application under test
- The tester must have knowledge of common implementation errors
- It is necessary to evaluate historical data and the test results obtained.

11. Risk-Based Testing Technique

This type of testing is implemented with the aim of finding out the critical errors as soon as possible with least cost. Here, functionalities are prioritized and tested according to the level to which they are error prone.

The steps to follow for effective risk-based testing include:

- Identify and prioritize risks
- Create test plans accordingly
- Eliminate or add risks according to results obtained

This 2019 will be a bumpy ride for those who do not stay at par with the latest software testing techniques. Therefore, if you wish to stay ahead in the IT industry, make sure you follow the mentioned software trends to help your testers get more effective QA solutions and tools!