# Top 20 Practical Software Testing Tips You Should Read Before Testing Any Application

Last Updated: April 3, 2022

***It would be great if all the testers read the Software Testing practices updated in this tutorial.***

Read each point carefully and try to implement them in your day-to-day testing activities. This is what we expect from the readers through this tutorial. If you don't understand any testing practice, then, **ask for more clarification in the comments section below.**

However, you will learn all these testing practices through experience. But why don't you learn all these things before making any mistakes?

**Come let's take a look at them!**

## About SoftwareTestingHelp

**Helping our community since 2006!** Most popular portal for Software professionals with **100 million+ visits and 300,000+ followers!** You will absolutely love our tutorials on QA Testing, Development, Software Tools and Services Reviews and more!

## Recommended Reading

## Best Software Testing Practices

**Here are some of the Best Testing Practices that can be learned from Experience:**

**#1) Learn to Analyze your test results thoroughly.** Do not ignore any test results. The final test result may be a "pass" or a "fail", but troubleshooting the root cause of a "fail" will give you the solution to the problem. Testers will be respected if they not only log the Bugs but also provide solutions.

**#2) Learn to maximize Test Coverage** each time you test any application. 100% test coverage might not be possible but still, you can always try to reach near it.

**#3)** In order to ensure maximum test coverage, **break your Application Under Test (AUT) into smaller functional modules.** Write test cases on such individual unit modules. Also, if possible

break these modules into smaller parts.

**For example,** let's assume that you have divided your website application into modules and 'accepting user information' is one of the modules. You can break this 'User information' screen into smaller parts for writing test cases: Parts like UI testing, Security Testing, Functional Testing of the 'User information' form etc.

Apply all forms of field types and size tests, negative and validation tests in the input fields and write all such test cases for maximum coverage.

**#4) While Writing Test Cases**, you must write test cases for intended functionality first i.e. for valid conditions according to requirements. Then write test cases for invalid conditions. This will cover the expected as well as unexpected behavior of the application under test.

**#5) Think positive.** Start testing the application with the intent of finding bugs/errors. Don't think beforehand that there will not be any bugs in the application. If you test the application with the intention of finding bugs you will definitely succeed in finding those Subtle Bugs as well.

**#6)** Write your test cases in the requirement analysis and design phase itself. This way you can ensure that all the requirements are testable.

**#7)** Make your **test cases available to the developers prior to coding.** Don't keep your test cases with you thereby waiting to get a final application release for testing, thinking that you can log more bugs. Let the developers analyze your test cases thoroughly to develop a quality application. This will also save the re-work time.

**#8)** If possible identify and **group your test cases for Regression Testing.** This will ensure quick and effective manual Regression Testing.

**#9)** Applications requiring critical response time should be thoroughly tested for performance. **Performance testing is a critical part of many applications.** For Manual Testing, this is the most

ignored part by testers due to a lack of required large data volume in performance testing.

Find ways to test your application for performance. If it is not possible to create test data manually, then write some basic scripts to create test data for performance tests or ask the developers to write one for you.

**#10) Programmers should not test their own code.** As discussed in our previous post, basic Unit Testing of developed applications should be enough for developers to release the application for testers. But you (tester) should not force the developers to release the product for testing.

Let them take their own time. Everyone right from the lead to the manager knows when the module/update will be released for testing and they can estimate the testing time accordingly. This is typical in an Agile project environment.

**#11) Go beyond Requirement Testing.** Test the application for what it is not supposed to do.

**#12)** While doing regression testing, **use the previous Bug graph** (Bug graph – number of bugs found against time for different modules). This module-wise bug graph can be useful to predict the most probable bug part of the application.

**#13)** Note down the new terms and concepts you learn while testing. Keep the text file open while testing any application. **Note down the testing progress and observations** in it. Use these notepad observations while preparing the final test release report. This good habit will help you to provide a complete unambiguous test report and release details.

**#14)** Many times testers or developers make changes in the code base for application under test. This is a required step in development or testing environment to avoid execution of the live transaction processing like in banking projects.

**Note down all such code changes done for testing purposes** and at the time of final release make sure you have removed all these changes from the final client-side deployment file

resources.

**#15) Keep developers away from the test environment.** This is a required step to detect any configuration changes missing in the release or deployment document. Sometimes developers make some system or application configuration changes but forget to mention those in the deployment steps.

If the developers don't have access to the testing environment they will not make any such changes accidentally on the test environment and these missing things that can be captured at the right place.

**#16)** It's a good practice to **involve testers right from the Software Requirement and Design phase itself.**

This way, testers can gain knowledge of application dependability resulting in detailed test coverage. If you are not being asked to be a part of this development cycle then you can make a request to your lead or manager to involve your testing team in all the decision making processes or meetings.

**#17)** Testing teams should **share best testing practices and** experience with the other teams in their organization.

**#18) Increase your conversation with the developers** to know more about the product. Whenever possible, make face-to-face communication to resolve disputes quickly and to avoid any misunderstandings.

But also when you understand the requirements or resolve any dispute – make sure to communicate the same overwritten communication ways like emails. Do not keep anything verbal.

**#19) Don't run <span style="color:red">out of time</span> to do high priority testing tasks.** Prioritize your testing work from high to low priority and plan your work accordingly. Analyze all associated risks to prioritize

your work.

**#20) Write a clear, descriptive, and unambiguous <span style="color:red">Bug Report</span>.** Do not only provide the bug symptoms but also provide the effect of the bug and all the possible solutions.

Don't forget that testing is a creative and challenging task. Finally, it all depends on your skill and experience as to how you handle this challenge.

**Over to you:**

Sharing your own testing experience, tips or testing secrets in the comments below will definitely make this article more interesting and helpful!!

*Let us know your thoughts/suggestions about this article.*

## Recommended Reading

- Best Software Testing Tools 2022 [QA Test Automation Tools]
- Software Testing QA Assistant Job
- Is Software Testing an Emotional Task?
- Software Testing Course: Which Software Testing Institute Should I join?
- Choosing Software Testing as your Career
- Software Testing Technical Content Writer Freelancer Job
- What is Monkey Testing in Software Testing?
- Application Testing – Into the Basics of Software Testing!

# 253 thoughts on "Top 20 Practical Software Testing Tips You Should Read Before Testing Any Application"

**Stacy Lawrence**

thank you for sharing this informative guide on tester core skills required, as I am my self a tester and with surety can share that this is worth a read!!!

Reply

**Quality Professionals**

I read this Blog, "Tips before testing any application" this will help my company q-pros.com, and I must say your post is very informative and helpful to understand the topic easily. Thanks for sharing this Blog. Keep posting more related to mobile app testing and the Best tools for mobile app testing.

Reply

# Leave a Comment

[                                                                    ]
[                                                                    ]
[                                                                    ]
[                                                                    ]
[                                                                    ]

[ Name * ]

[ Email * ]

**POST COMMENT**