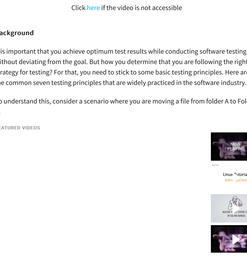
7 Principles of Software Testing: Learn with Examples

This tutorial introduces the seven basic Software Testing Principles that every Software tester and QA professional should know.

7 Principles of Software Testing

- Testing shows presence of defects
- Exhaustive testing is not possible
- Early testing
- Defect clustering
- Pesticide paradox
- Testing is context dependent
- Absence of errors fallacy

Let's learn the testing principles with the following video example-



Click here if the video is not accessible

Background

It is important that you achieve optimum test results while conducting software testing without deviating from the goal. But how you determine that you are following the right strategy for testing? For that, you need to stick to some basic testing principles. Here are the common seven testing principles that are widely practiced in the software industry.

To understand this, consider a scenario where you are moving a file from folder A to Folder B.

FEATURED VIDEOS

Think of all the possible ways you can test this.

Apart from the usual scenarios, you can also test the following conditions

- Trying to move the file when it is Open
- You do not have the security rights to paste the file in Folder B
- Folder B is on a shared drive and storage capacity is full.
- Folder B already has a file with the same name, in fact, the list is endless
- Or suppose you have 15 input fields to test, each having 5 possible values, the number of combinations to be tested would be 5^15

If you were to test the entire possible combinations project EXECUTION TIME & COSTS would rise exponentially. We need certain principles and strategies to optimize the testing effort

Here are the 7 Principles:

1) Exhaustive testing is not possible

Yes! Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application.

And the million dollar question is, how do you determine this risk?

To answer this let's do an exercise

In your opinion, Which operation is most likely to cause your Operating system to fail?

I am sure most of you would have guessed, Opening 10 different application all at the same time.

So if you were testing this Operating system, you would realize that defects are likely to be found in multi-tasking activity and need to be tested thoroughly which brings us to our next principle Defect Clustering

2) Defect Clustering

Defect Clustering which states that a small number of modules contain most of the defects detected. This is the application of the Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules.

By experience, you can identify such risky modules. But this approach has its own problems

If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

3) Pesticide Paradox

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.

To overcome this, the test cases need to be regularly reviewed & revised, adding new & different test cases to help find more defects.

Testers cannot simply depend on existing test techniques. He must look out continually to improve the existing methods to make testing more effective. But even after all this sweat & hard work in testing, you can never claim your product is bug-free. To drive home this point, let's see this video of the public launch of Windows 98

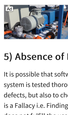
You think a company like MICROSOFT would not have tested their OS thoroughly & would risk their reputation just to see their OS crashing during its public launch!

4) Testing shows a presence of defects

Hence, testing principle states that - Testing talks about the presence of defects and don't talk about the absence of defects. i.e. Software Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

But what if, you work extra hard, taking all precautions & make your software product 99% bug-free. And the software does not meet the needs & requirements of the clients.

This leads us to our next principle, which states that- Absence of Error



5) Absence of Error - fallacy

It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement. Software testing is not mere finding defects, but also to check that software addresses the business needs. The absence of Error is a Fallacy i.e. Finding and fixing defects does not help if the system build is unusable and does not fulfill the user's needs & requirements.

To solve this problem, the next principle of testing states that Early Testing

6) Early Testing

Early Testing - Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing. But "early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined. More on this principle in a later training tutorial.

7) Testing is context dependent

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application. All the developed software's are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type. For instance testing, any POS system at a retail store will be different than testing an ATM machine.

Myth: "Principles are just for reference. I will not use them in practice ."

This is so very untrue. Test Principles will help you create an effective Test Strategy and draft error catching test cases.

But learning testing principles is just like learning to drive for the first time.

Initially, while you learn to drive, you pay attention to each and everything like gear shifts, speed, clutch handling, etc. But with experience, you just focus on driving the rest comes naturally. Such that you even hold conversations with other passengers in the car.

Same is true for testing principles. Experienced testers have internalized these principles to a level that they apply them even without thinking. Hence the myth that the principles are not used in practice is simply not true.

Prev | Report a Bug | Next

YOU MIGHT LIKE:

SOFTWARE TESTING

Test Condition vs Test Scenario: What's the Difference?
What is a Test Scenario? A Test Scenario is a probable way or method to test an Application. Test...
Read more »

SDLC

Incremental Model in SDLC: Use, Advantage & Disadvantage
What is Incremental Model? Incremental Model is a process of software development where...
Read more »

SOFTWARE TESTING

What is STRESS Testing in Software Testing? Tools, Types, Examples
Stress Testing Stress Testing is a type of software testing that verifies stability & reliability of...
Read more »

SOFTWARE TESTING

What is Operational Acceptance Testing(OAT)? Example Test Cases
Operational Acceptance Testing Operational Acceptance Testing (OAT) is a software testing...
Read more »

LOADRUNNER

Correlation in LoadRunner with Web_Reg_Save_Param Example
What is Correlation? Correlation, as the name suggests, is a mechanism of defining a relationship between...
Read more »

SOFTWARE TESTING

What is Benchmark Testing? Test Plan, Tools, Example
Before we learn Benchmark Testing, let's understand-Benchmark in Performance Testing A Benchmark in...
Read more »

Testing Tutorials
Testing Career Path
Introduction
Seven Principles
V-Model
Scaled Agile Framework(SAFe)

Top Tutorials
Selenium
Testing
Hacking
SAP
Java
Python
Jmeter
Informatica

About
About Us
Advertise with Us
Write For Us
Contact Us

Career Suggestion
SAP Career Suggestion Tool
Software Testing as a Career

Interesting
eBook
Blog
Quiz
SAP eBook

Execute online
Execute Java Online
Execute Javascript
Execute HTML
Execute Python

© Copyright - Guru99 2021 Privacy Policy | Affiliate Disclaimer | ToS