

### Software Testing Magazine



HOME KNOWLEDGE TOOLS VIDEOS CONFERENCES RESOURCES

NEWS CONTACT

# How to Choose the Right Name for Unit Tests

⊙ May 22, 2023 Software Testing Magazine Articles,
 Knowledge © 0

As unit testing is often the first quality activity targeting code during the software development lifecycle, this article considers several examples of effective approaches to the naming of unit tests.

Author: Nataliia Syvynska,

TestMatick, https://testmatick.com/

Unit testing is an essential part of software development that helps ensure code quality and catch bugs before they become bigger problems. However, writing effective unit tests requires more than just writing code that checks for the expected output.

One often overlooked aspect of unit testing is the importance of naming tests correctly. A well-named

### **SEARCH**

SEARCH ...

BE NICE: SHARE THIS KNOWLEDGE

### SOFTWARE TESTING WHITE PAPERS

Improve your Software Quality Process with Test Data Management

Come To Our Great SQA Job Portal – Qualitician

Best Features to Look for in API Testing Tools

unit test can communicate its purpose clearly, make it easier to understand the functionality being tested and help organize tests in large codebases.

That is why in this article we will consider several examples of effective approaches to the naming of unit tests. By following these guidelines, you can improve the readability and maintainability of your unit tests, making unit testing an even more valuable activity in your software development process.

First of all, let's take a look at some fundamental guidelines for test naming:

- 1. Be descriptive: it should be clear and easy to understand what the test is verifying.
- 2. Use a naming convention: such as prefixing the test name with "test" or using camelCase. This makes it easy to identify unit tests and also helps automated testing tools locate and run them.
- 3. Keep it short: a long name can make it difficult to read and understand the purpose of the test.
- 4. Use meaningful names that reflect the functionality being tested. Avoid using generic names like "test1" or "test2" which do not convey any information about the test.
- 5. Use underscores or dashes for readability.

  Avoid using spaces or other special characters that can cause issues with certain testing frameworks.
- 6. Follow a consistent structure. For example, you may choose to prefix all tests related to a

### **SEARCH WITH KEYWORDS**

.NET ACCEPTANCE TESTING

AGILE TESTING ANDROID

APPIUM ARQUILLIAN

**BEHAVIOR-DRIVEN DEVELOPMENT** 

**CODE ANALYSIS** 

**CONTINUOUS DEPLOYMENT** 

**CONTINUOUS INTEGRATION** 

**CUCUMBER DATABASE** 

**DEVOPS** 

**EXPLORATORY TESTING** 

**FUNCTIONAL TESTING** 

**INTEGRATION TESTING** 

JAVA JAVASCRIPT

JMETER JUNIT

**LOAD TESTING** 

MOBILE TESTING MOCKING

OPEN SOURCE PEOPLE

PERFORMANCE TESTING

PHP PYTHON

REFACTORING

REGRESSION TESTING RUBY

**SELENIUM** 

**SOFTWARE QUALITY ASSURANCE** 

TDD TEAM

2 of 6

particular module with the module name, followed by an underscore and the test description.

7. Avoid redundant information. For example, if a test is checking the behavior of a specific function, there is no need to include the function name in the test name.

Thus, we will give some popular names which are believed to be in demand among developers and QA specialists alike.

### Designation

### MethodName\_StateUnderTest\_ExpectedBehavior

There are several arguments against this approach because if method names are changed after the refactoring process, it should (additionally) change the name of the test, and this becomes very difficult to understand late in the development process and independent testing.

Here are a few examples of this designation.

```
public void IsAdult_AgeLessThan18_False()
{
}

public void WithdrawMoney_InvalidAccount_ExceptionThrown()
{
}

public void AdmitStudent_MissingMandatoryFields_FailToAdmit()
{
}
```

## Designation MethodName\_ExpectedBehavior\_St ateUnderTest

This variant is slightly different from the first one,

**TEST AUTOMATION** 

TEST CASE TEST COVERAGE

**TEST MANAGEMENT** 

TEST PLAN UNIT TESTING

**USER INTERFACE** 

**VISUAL STUDIO** 

**WEB TESTING** 

**WHITE PAPERS** 

#### **PARTNER CONFERENCES**

Agile Testing Days USA, May 22–24 2023, Chicago/Illinois South East European Software Testing Conference (SEETEST), Belgrade, Serbia Pacific Northwest Software Quality Conference, Portland, USA

but some programmers also recommend using this method of naming. By the way, this designation also has the same disadvantage as the previous one, if the names of the methods change, it becomes extremely difficult to understand the names of the tests at later stages.

Below is an example of how the group of tests from the first example would be interpreted as if those names were used with the names from the second designation.

```
public void IsAdult_False_AgeLessThan18()
{
}
public void WithdrawMoney_ThrowsException_IfAccountIsInvalid()
{
}
public void AdmitStudent_FailToAdmit_IfMandatoryFieldsAreMissing()
{
}
```

### **Designation Feature to be tested**

Some people think that it is more efficient to write a trivial test function because in any case the annotation will be used to identify the method as a software test method. It is also recommended for the reason that it allows you to create a unit test in an alternative form, and does not "spoil" the software code.

An example of the use of such a designation.

# Designation Should\_ExpectedBehavior\_When\_StateUnderTest

It is very popular because with it you can easily explain what this or that tested function should do.

```
public void Should_ThrowException_When_AgeLessThan18()
{
}
public void Should_FailToWithdrawMoney_ForInvalidAccount()
{
}
public void Should_FailToAdmit_IfMandatoryFieldsAreMissing()
{
}
```

The use of data and similar designations will allow you not only to correctly structure unit tests, but also to quickly find parts of program code that need to be improved and tested.

### **Conclusion**

In conclusion, naming unit tests is an essential part of writing high-quality code. A good test name communicates the purpose of the test and helps other developers understand the functionality being tested. Naming unit tests consistently and following established conventions can also make it easier to find and organize tests in large codebases. When naming unit tests, it is important to use descriptive and specific names that communicate the expected behavior being tested. Additionally, it can be helpful to use a consistent format or structure for naming tests, such as starting with "test\_" or including the name of the method being tested. By following these guidelines, QA engineers from software testing companies can create well-named unit tests

that improve code quality and collaboration.

	share	share	tweet	share
•	UNIT TESTING			

#### **LAST SOFTWARE** RECENT CONTENT **RECENTS COMMENTS MORE INFORMATION TESTING NEWS** Top 5 Software to Spike Testing versus About Partnership Between Monitor and Manage Performance and Load Contact LambdaTest and Your Server Activity Testing on Free Load GenRocket Testing Tools & Services Partnership Between **SOFTWARE TESTING** LambdaTest and Irene Ervin on Mastering **KNOWLEDGE** No-Code Mobile Testing GenRocket the Basics of Web Tool Waldo Bought by **Tricentis** Application Testing: a Why Use Test Tools Methods & Tools **O** July 10, 2023 $\bigcirc$ 0 Guide for Checking Your **During Test Design** Software Development Product for Errors and Applitools Buys Preflight Magazine Advanced Software Scam Penetrability Teaming the Mob **Bug Tracking Tools** w3softech on 4 Tips to Zebrunner Launched **Programming Way** Improve the Quality of **Test Case Management Load Testing Tools** and Testing Platform Your Software Testing **Italian Software Testing** Forum, Online, Italy, Joann on Open Source November 21 2023 SapientAl Gets \$5 Million **Database Testing Tools** Financing for Al-powered Web Testing with Test Coder Software Development Nightwatch.js – Part 1 Linkopedia January 2023: Installation Agile, Scrum on Which How To Write The Perfect Tools Are The Best for Bug Report? Tips, Tricks **API Test Automation?**

Copyright © 2009-2023 Software Testing Magazine by Martinig & Associates

& Best Practices