

# What are the differences between unit tests, integration tests, smoke tests, and regression tests? [closed]

Asked 15 years, 10 months ago   Modified 2 years, 5 months ago

Viewed 363k times



811



**Closed.** This question needs to be more [focused](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it focuses on one problem only by [editing this post](#).

Closed 3 years ago.

The community reviewed whether to reopen this question 2 years ago and left it closed:

**Opinion-based** Update the question so it can be answered with facts and citations by [editing this post](#).

[Improve this question](#)

What are unit tests, integration tests, smoke tests, and regression tests? What are the differences between them and which tools can I use for each of them?

For example, I use [JUnit](#) and [NUnit](#) for *unit testing* and *integration testing*. Are there any tools for the last two, *smoke testing* or *regression testing*?

unit-testing

testing

definition

Share Follow

edited Jul 10, 2022 at 0:39



Michael Freidgeim

28.3k ● 17 ● 162 ● 178

asked Feb 6, 2009 at 12:08



caltuntas

10.8k ● 7 ● 37 ● 41

---

3 Related: [stackoverflow.com/questions/437897/...](https://stackoverflow.com/questions/437897/...)  
– [Bill the Lizard](#) Feb 6, 2009 at 12:57

---

2 Others have already answered well, but I'd like to add that I personally think that Smoke Test and Regression Test are redundant. They do the same thing: test to make sure that changes to the system didn't break anything. – [Randolpho](#) Mar 9, 2009 at 14:51

---

17 I think they are quite different to regression tests. I think they are deliberately 'light-weight' quick tests that are run at the start to save time because if any of these fail then you know it's not worth bothering with any additional testing. e.g. Can the client connect to the database, is .net installed, is the correct version installed... You might also have pre-deployment (we are upgrading from v1 to v1.1, so check that v1 is installed) and post-deployment smoke tests. – [AndyM](#) Mar 19, 2010 at 13:18

---

- 2 Related: [stackoverflow.com/questions/4904096/...](https://stackoverflow.com/questions/4904096/...)  
– [rogerdpack](#) Aug 4, 2015 at 16:27
- 
- 1 looks like you forgot to ask about canary tests, A/B tests, penetration tests, failover testing, health checks, HA/DR testing and so on ... – [Sławomir Lenart](#) Dec 25, 2019 at 11:55
- 

17 Answers

Sorted by:

Highest score (default)



1189



- **Unit test:** Specify and test one point of the contract of single method of a class. This should have a very narrow and well defined scope. Complex dependencies and interactions to the outside world are [stubbed or mocked](#).
- **Integration test:** Test the correct inter-operation of multiple subsystems. There is whole spectrum there, from testing integration between two classes, to testing integration with the production environment.
- **Smoke test (aka *sanity check*):** A simple integration test where we just check that when the system under test is invoked it returns normally and does not blow up.
  - Smoke testing is both an analogy with electronics, where the first test occurs when powering up a circuit (if it smokes, it's bad!)
  - ... and, [apparently](#), with [plumbing](#), where a system of pipes is literally filled by smoke and

then checked visually. If anything smokes, the system is leaky.

- **Regression test:** A test that was written when a bug was fixed. It ensures that this specific bug will not occur again. The full name is "non-regression test". It can also be a test made prior to changing an application to make sure the application provides the same outcome.

To this, I will add:

- **Acceptance test:** Test that a feature or use case is correctly implemented. It is similar to an integration test, but with a focus on the use case to provide rather than on the components involved.
- **System test:** Tests a system as a black box. Dependencies on other systems are often mocked or stubbed during the test (otherwise it would be more of an integration test).
- **Pre-flight check:** Tests that are repeated in a production-like environment, to alleviate the 'builds on my machine' syndrome. Often this is realized by doing an acceptance or smoke test in a production like environment.

Share Follow

edited Jun 1, 2020 at 12:54



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Feb 6, 2009 at 12:28



ddaa

54.3k ● 8 ● 52 ● 59

---

282 Smoke testing [predates electronics by a century](#) and comes from plumbing, when a system of pipes were filled by an actual smoke and then checked visually. If it smoked, it was leaky. – [SnakeE](#) May 10, 2012 at 12:52

---

4 Regression tests are also used for feature changes, not just bug fixes. Nikita's answer below is a more comprehensive definition. – [BobRodes](#) Dec 13, 2012 at 19:53 ✎

---

26 @AndyM The background of 'Smoke test' is inaccurate. IIRC it comes from plumbing, where smoke is pumped in the system of pipes after it's built (and before it's connected to the water supply). If any smoke comes out, the pipes are not properly sealed. This is less damaging than actually letting the water flow and see if any puddles occur (possibly damaging walls/masonry in the process). It's a gross approximation that the system will not fail catastrophically. A dev process may be: "Build " passed? => "Smoke test" passed ? => "Acceptance test" passed => out to QA team for detailed testing. – [Cristian Diaconescu](#) Jun 5, 2013 at 11:58

---

6 I believe you made an error in stating that a "Regression Test" really shorthand for "Non-Regression Test"? I am skeptical, in part because that's just unintuitive and confusing (though there are plenty of terms that are), but also Wikipedia has two separate articles on Regression and Non-Regression Tests. The article on Regression testing even says: *Contrast with non-regression testing... which aims to verify whether, after introducing or updating a given software application, the change has had the intended effect.* – [Brian C](#) Jun 23, 2016 at 17:56

---

2 @ddaa Sanity testing and smoke testing are not same. Sanity testing is performed after the build has clear the Smoke test and has been accepted by QA team for further

testing, sanity testing checks the major functionality with finer details. – [Bharat Pahalwani](#) Dec 2, 2016 at 6:43

---



116



- **Unit test:** an automatic test to test the internal workings of a class. It should be a stand-alone test which is not related to other resources.
- **Integration test:** an automatic test that is done on an environment, so similar to unit tests but with external resources (db, disk access)
- **Regression test:** after implementing new features or bug fixes, you re-test scenarios which worked in the past. Here you cover the possibility in which your new features break existing features.
- **Smoke testing:** first tests on which testers can conclude if they will continue testing.

Share Follow

answered Feb 6, 2009 at 12:13



[Gerrie Schenck](#)


22.4k ● 21 ● 69 ● 96

---

2 Regression test's definition isn't really exactly how it is. @ddaa defines it correctly. – [Robert Koritnik](#) Oct 11, 2010 at 14:37

---

The definition of Integration Test is definitely fuzzy. For instance on the answer here [stackoverflow.com/a/4904533/32453](https://stackoverflow.com/a/4904533/32453) it's more defined as testing multiple interactions of your code, not necessarily needing a real DB (external resource)...though some people define it the way you've described...ahh terminology. (I do

somewhat prefer the former definition, FWIW, testing multiple interactions.) – [rogerdpack](#) Aug 4, 2015 at 16:31 

---

That answers the title, but not the one about **tools** for the last two types of tests, for *smoke testing* or *regression testing*.

– [Peter Mortensen](#) Jun 1, 2020 at 13:02

---



Everyone will have slightly different definitions, and there are often grey areas. However:

99



- Unit test: does this one little bit (as isolated as possible) work?
- Integration test: do these two (or more) components work together?
- Smoke test: does this whole system (as close to being a production system as possible) hang together reasonably well? (i.e. are we reasonably confident it won't create a black hole?)
- Regression test: have we inadvertently re-introduced any bugs we'd previously fixed?

Share Follow

answered Feb 6, 2009 at 12:15



[Jon Skeet](#)

1.5m ● 889 ● 9.3k ● 9.3k

---

How do you place your integration tests with respect to unit-tests? If `myprj` is the main project directory, and `mypkg` is located under `myprj`, I have the unit tests located under `myprj/tests/test_module1.py`, and my package located under `myprj/mypkg`. This works great for unit tests, but I wonder if there is any convention, I should follow for where

the integration tests should reside? – [alpha\\_989](#) Feb 15, 2018 at 23:38

---

- 1 @alpha\_989: I don't know what the convention would be for Python. In .NET I currently have the production code, unit tests and integration tests in three separate projects, peers of each other - but there are lots of alternatives too. – [Jon Skeet](#) Feb 16, 2018 at 3:25
- 

That answers the title, but not the one about **tools** for the last two types of tests, for *smoke testing* or *regression testing*. – [Peter Mortensen](#) Jun 1, 2020 at 13:02

---

@PeterMortensen: That part of the question was added after I'd already written this answer - and I'd argue that it's off-topic for Stack Overflow, in the normal way of "Seeking recommendations for books, tools, software libraries, and more" – [Jon Skeet](#) Jun 1, 2020 at 13:56

---

- 1 1. end-to-end-testing is not mention and I think it is important too are part of this Testing pyramid in the link below [lawrey.medium.com/...](#) End-to-end tests should be run using both automated testing and manual testing tactics – [leonidaa](#) Apr 19, 2022 at 10:26 ✎
- 



57



A new test category I've just become aware of is the [canary test](#). A canary test is an automated, non-destructive test that is **run on a regular basis** in a **live** environment, such that if it ever fails, something really bad has happened.



Examples might be:



- Has data that should only ever be available in development/testy appeared *live*?



- Has a background process failed to run?
- Can a user logon?

Share Follow

edited Jun 1, 2020 at 13:20



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Sep 11, 2013 at 3:47



AndyM

3,774 ● 6 ● 43 ● 45

---

3 Can the site be pinged at all - suitably enough, a service called Binary Canary exists. – [Dan Dascalescu](#) Feb 23, 2014 at 23:53

---

17 The name comes from coal mining: take canary with you "down t'pit". When it snuffs it, get out quick. Canaries are very sensitive to small concentrations of noxious gas and would die before the concentrations levels became toxic to humans. If a Canary test fails, fix it quickly because it'll only be a matter of time before LIVE fails. – [Robino](#) Sep 3, 2014 at 12:06

---

1 The way we use Canary testing at my job is to first shift a few customers over to a new version, rather than all of them at once. If the first few customers survive, we can add the rest. Those first few are the canaries. – [00prometheus](#) Sep 15, 2016 at 19:23 ✎

---

2 @00prometheus, that's beta testing. – [GregNash](#) Jan 19, 2017 at 22:32

---

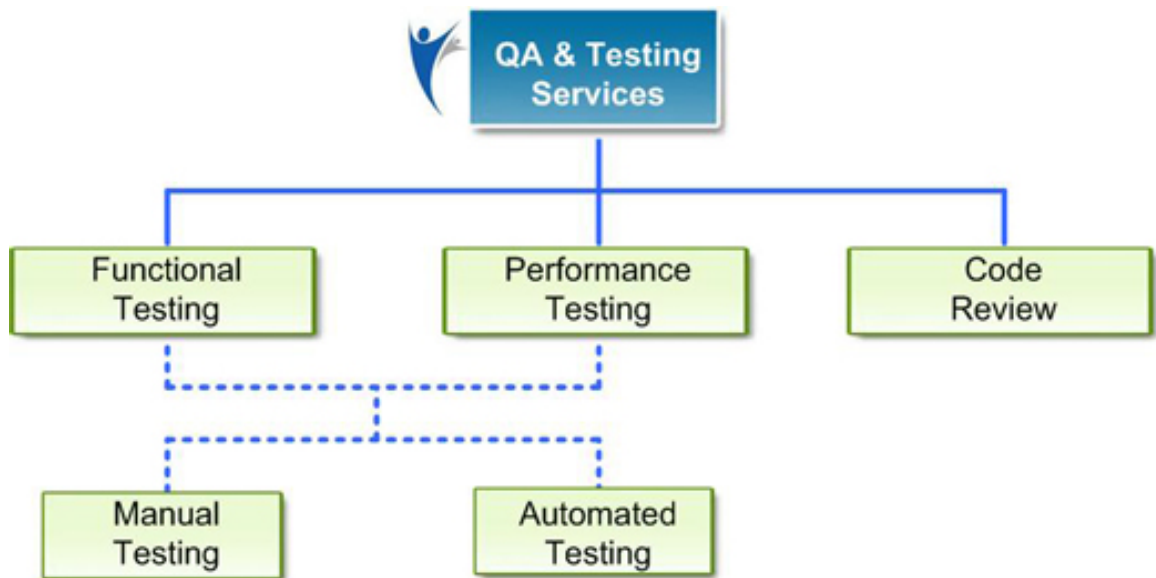
1 @HarveyLin, although a Canary test is necessarily a test that prevents disaster, of course it's not only used this way. But the rule of thumb is "test *this* is working 'cause it IS critical". Of course, every test has *almost* this same goal, but the concept is very specific. In your case, I wouldn't count all



**Answer from one of the best websites for software testing techniques:**

**14**

**Types of software testing – complete list [click here](#)**



It's quite a long description, and I'm not going to paste it here: but it may be helpful for someone who wants to know all the testing techniques.

Share Follow

edited Jun 1, 2020 at 14:02



**Peter Mortensen**

31.6k ● 22 ● 109 ● 133

answered Nov 26, 2015 at 7:28



**Kaleem Ullah**

7,019 ● 3 ● 43 ● 47

---

That article is mixed types of test with methods of test...  
– [Yousha Aleayoub](#) Jun 11, 2020 at 10:56

---



13

**Unit test:** testing of an individual module or independent component in an application is known to be unit testing. The unit testing will be done by the developer.



**Integration test:** combining all the modules and testing the application to verify the communication and the data flow between the modules are working properly or not.



This testing also performed by developers.



**Smoke test** In a smoke test they check the application in a shallow and wide manner. In smoke testing they check the main functionality of the application. If there is any blocker issue in the application they will report to developer team, and the developing team will fix it and rectify the defect, and give it back to the testing team. Now testing team will check all the modules to verify that changes made in one module will impact the other module or not. *In smoke testing* the test cases are scripted.

**Regression testing** executing the same test cases repeatedly to ensure that the unchanged module does not cause any defect. Regression testing comes under functional testing



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Apr 26, 2014 at 12:02



malini

187 ● 1 ● 2

---

That answers the title, but not the one about tools for the last two types of tests, for smoke testing or regression testing. It also repeats previous answers - it could be made unique by answering the question about tools. – [Peter Mortensen](#) Jun 1, 2020 at 13:35

---



10



Unit test: Verifying that particular component (i.e., class) created or modified functions as designed. This test can be manual or automated, but it does not move beyond the boundary of the component.

Integration test: Verifying that the interaction of particular components function as designed. Integration tests can be performed at the unit level or the system level. These tests can be manual or automated.

Regression test: Verifying that new defects are not introduced into existing code. These tests can be manual or automated.

Depending upon your [SDLC](#) ([waterfall](#), [RUP](#), [agile](#), etc.) particular tests may be performed in 'phases' or may all be performed, more or less, at the same time. For example, unit testing may be limited to developers who then turn the code over to testers for integration and

regression testing. However, another approach might have developers doing unit testing and some level of integration and regression testing (using a [TDD](#) approach along with continuous integration and automated unit and regression tests).

The tool set will depend largely on the codebase, but there are many open source tools for unit testing (JUnit). HP's (Mercury) QTP or Borland's [Silk Test](#) are both tools for automated integration and regression testing.

Share Follow

edited Jun 1, 2020 at 13:10

community wiki  
[2 revs](#), [2 users](#) [61%](#)  
[rdrex](#)



## REGRESSION TESTING-

9



"A regression test re-runs previous tests against the changed software to ensure that the changes made in the current software do not affect the functionality of the existing software."



Share Follow

answered Oct 21, 2010 at 5:28



[Nikita G](#)

107 ● 1 ● 1

---

19 Where are you quoting from? – [Daryl](#) Mar 22, 2013 at 21:26

---

- 4 According to [this page](#), that quote came from the [Wikipedia "Software testing" article](#) though it seems that the passage has been changed at some point since 2010.  
– [Zach Lysobey](#) Oct 29, 2013 at 15:58 ✎
- 

Anyways, WP is not a valid source. Sources referenced there may be valid. There are no valid sources referenced in WP, neither in the articles nor on the discussion pages, that would support the claim that the "non-" makes any difference. I compared the text snippets in the result lists from searches in Google Books for both "regression test" and "non-regression test". It's the same.  
– [Rainald62](#) Jul 24, 2018 at 15:14

---

That answers (part of) the title, but not the one about tools for the last two types of tests, for smoke testing or regression testing. It also repeats previous answers - it could be made unique by answering the question about tools.  
– [Peter Mortensen](#) Jun 1, 2020 at 13:49 ✎

---



7

I just wanted to add and give some more context on why we have these levels of test, what they really mean with examples



Mike Cohn in his book “Succeeding with Agile” came up with the “Testing Pyramid” as a way to approach automated tests in projects. There are various interpretations of this model. The model explains what kind of automated tests need to be created, how fast they can give feedback on the application under test and who writes these tests. There are basically 3 levels of automated testing needed for any project and they are as follows.

**Unit Tests-** These test the smallest component of your software application. This could literally be one function in a code which computes a value based on some inputs. This function is part of several other functions of the hardware/software codebase that makes up the application.

For example - Let's take a web based calculator application. The smallest components of this application that needs to be unit tested could be a function that performs addition, another that performs subtraction and so on. All these small functions put together makes up the calculator application.

Historically developer writes these tests as they are usually written in the same programming language as the software application. Unit testing frameworks such as JUnit and NUnit (for java), MSTest (for C# and .NET) and Jasmine/Mocha (for JavaScript) are used for this purpose.

The biggest advantage of unit tests are, they run really fast underneath the UI and we can get quick feedback about the application. This should comprise more than 50% of your automated tests.

**API/Integration Tests-** These test various components of the software system together. The components could include testing databases, API's (Application Programming Interface), 3rd party tools and services along with the application.

For example - In our calculator example above, the web application may use a database to store values, use API's to do some server side validations and it may use a 3rd party tool/service to publish results to the cloud to make it available across different platforms.

Historically a developer or technical QA would write these tests using various tools such as Postman, SoapUI, JMeter and other tools like Testim.

These run much faster than UI tests as they still run underneath the hood but may consume a little more time than unit tests as it has to check the communication between various independent components of the system and ensure they have seamless integration. This should comprise more than 30% of the automated tests.

**UI Tests-** Finally, we have tests that validate the UI of the application. These tests are usually written to test end to end flows through the application.

For example - In the calculator application, an end to end flow could be, opening up the browser-> Entering the calculator application url -> Logging in with username/password -> Opening up the calculator application -> Performing some operations on the calculator -> verifying those results from the UI -> Logging out of the application. This could be one end to end flow that would be a good candidate for UI automation.



Historically, technical QA's or manual testers write UI tests. They use open source frameworks like Selenium or UI testing platforms like Testim to author, execute and maintain the tests. These tests give more visual feedback as you can see how the tests are running, the difference between the expected and actual results through screenshots, logs, test reports.

The biggest limitation of UI tests is, they are relatively slow compared to Unit and API level tests. So, it should comprise only 10-20% of the overall automated tests.

The next two types of tests can vary based on your project but the idea is-

### **Smoke Tests**

This can be a combination of the above 3 levels of testing. The idea is to run it during every code check in and ensure the critical functionalities of the system are still working as expected; after the new code changes are merged. They typically need to run with 5 - 10 mins to get faster feedback on failures

### **Regression Tests**

They usually are run once a day at least and cover various functionalities of the system. They ensure the application is still working as expected. They are more details than the smoke tests and cover more scenarios of the application including the non-critical ones.

Share Follow

edited Aug 26, 2019 at 14:36



halfer

20.4k ● 19 ● 108 ● 200

answered Jul 24, 2019 at 19:13



Raj Subrameyer

432 ● 4 ● 6

---

This answer could be made better by answering the question about tools for *smoke testing* or *regression testing*.

– [Peter Mortensen](#) Jun 1, 2020 at 13:54

---



5



- Integration testing: Integration testing is the integrate another element
- Smoke testing: Smoke testing is also known as build version testing. Smoke testing is the initial testing process exercised to check whether the software under test is ready/stable for further testing.
- Regression testing: Regression testing is *repeated* testing. Whether new software is effected in another module or not.
- Unit testing: It is a white box testing. Only *developers* involve in it

Share Follow

edited Jun 1, 2020 at 13:25



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Jun 30, 2014 at 9:50



Madhivanan

139 ● 2 ● 5

---

That answers the title, but not the one about tools for the last two types of tests, for smoke testing or regression testing. It also repeats previous answers - it could be made unique by answering the question about tools. – [Peter Mortensen](#) Jun 1, 2020 at 13:35

---



5



**Unit testing** is directed at the smallest part of the implementation possible. In Java this means you are testing a single class. If the class depends on other classes these are faked.

When your test calls more than one class, it's an **integration test**.

Full test suites can take a long time to run, so after a change many teams run some quick to complete tests to detect significant breakages. For example, you have broken the URIs to essential resources. These are the **smoke tests**.

**Regression tests** run on every build and allow you to refactor effectively by catching what you break. Any kind of test can be regression test, but I find unit tests are most helpful finding the source of fault.

Share Follow

edited Jun 1, 2020 at 13:32



[Peter Mortensen](#)

31.6k ● 22 ● 109 ● 133

answered Oct 12, 2014 at 19:55



Dave

4,308 ● 9 ● 45 ● 48

---

That answers the title, but not the one about tools for the last two types of tests, for smoke testing or regression testing. It also repeats previous answers - it could be made unique by answering the question about tools. – [Peter Mortensen](#) Jun 1, 2020 at 13:34 ✎

---



3



## Unit Testing

Unit testing is usually done by the developers side, whereas testers are partly evolved in this type of testing where testing is done unit by unit. In Java [JUnit](#) test cases can also be possible to test whether the written code is perfectly designed or not.

## Integration Testing:

This type of testing is possible after the unit testing when all/some components are integrated. This type of testing will make sure that when components are integrated, do they affect each others' working capabilities or functionalities?

## Smoke Testing

This type of testing is done at the last when system is integrated successfully and ready to go on production server.

This type of testing will make sure that every important functionality from start to end is working fine and system is ready to deploy on production server.

## Regression Testing

This type of testing is important to test that unintended/unwanted defects are not present in the system when developer fixed some issues. This testing also make sure that all the bugs are successfully solved and because of that no other issues are occurred.

Share Follow

edited Jun 20, 2020 at 9:12



Community Bot

1 ● 1

answered Sep 16, 2014 at 10:50



mohit sarsar

189 ● 3 ● 8

---

That answers the title, but not the one about tools for the last two types of tests, for smoke testing or regression testing. It also repeats previous answers - it could be made unique by answering the question about tools. – [Peter Mortensen](#) Jun 1, 2020 at 13:34

---



3



Smoke and sanity testing are both performed after a software build to identify whether to start testing. Sanity may or may not be executed after smoke testing. They can be executed separately or at the same time - sanity being immediately after smoke.

**Because sanity testing is more in-depth and takes more time, in most cases it is well worth to be automated.**

Smoke testing usually takes no longer than 5-30 minutes for execution. It is more general: it checks a small number of core functionalities of the whole system, in order to verify that the stability of the software is good enough for further testing and that there are no issues, blocking the run of the planned test cases.

Sanity testing is more detailed than smoke and may take from 15 minutes up to a whole day, depending on the scale of the new build. It is a more specialized type of acceptance testing, performed after progression or re-testing. It checks the core features of certain new functionalities and/or bug fixes together with some closely related to them features, in order to verify that they are functioning as to the required operational logic, before regression testing can be executed at a larger scale.

Share Follow

edited Jun 1, 2020 at 13:59



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Nov 17, 2017 at 11:58



Radostta

41 ● 3

- 
- 1 This elaborates somewhat, but not about **tools** for the last two types of tests, for *smoke testing* or *regression testing*. It could be made unique by answering the question about tools.  
– [Peter Mortensen](#) Jun 1, 2020 at 14:00 ✎
- 



2



Unit Testing: It always performs by developer after their development done to find out issue from their testing side before they make any requirement ready for QA.

Integration Testing: It means tester have to verify module to sub module verification when some data/function output are drive to one module to other module. Or in your system if using third party tool which using your system data for integrate.

Smoke Testing: tester performed to verify system for high-level testing and trying to find out show stopper bug before changes or code goes live.

Regression Testing: Tester performed regression for verification of existing functionality due to changes implemented in system for newly enhancement or changes in system.

Share Follow

answered Jul 18, 2017 at 12:29



Krunal

61 ● 1 ● 10

---

Don't we have to create the test before doing the actual development? – [Vin Shahrddar](#) Nov 29, 2017 at 15:55

---

@VinShahrddar , Are you talking about Unit testing ? – [Krunal](#) Dec 1, 2017 at 10:24

---

yes. I usually create my unit tests before writing production code. That's how you're supposed to do it, correct?

– [Vin Shahrddar](#) Dec 2, 2017 at 22:10

---

- 1 Yeah .. But unit testing also performs before QA faced by the developer. Before deploying code on QA server dev always perform unit testing – [Krunal](#) Dec 6, 2017 at 11:52
- 



2



Regression test - Is a type of software testing where we try to cover or check around the bug *fix*. The functionality around the bug fix should not get changed or altered due to the fix provided. Issues found in such process are called as *regression issues*.



Smoke Testing: Is a kind of testing done to decide whether to accept the build/software for further QA testing.

Share Follow

edited Jun 1, 2020 at 14:06



[Peter Mortensen](#)

31.6k ● 22 ● 109 ● 133

answered Jul 26, 2015 at 4:23



[Sanyal](#)

894 ● 10 ● 22

---





There are some good answers already, but I would like further refine them:

1



Unit testing is the only form of white box testing here. The others are black box testing. White box testing means that you know the input; you know the inner workings of the mechanism and can inspect it and you know the output. With black box testing you only know what the input is and what the output should be.



So clearly unit testing is the only white box testing here.

- Unit testing test specific pieces of code. Usually methods.
- Integration testing test whether your new feature piece of software can integrate with everything else.
- Regression testing. This is testing done to make sure you haven't broken anything. Everything that used to work should still work.
- Smoke testing is done as a quick test to make sure everything looks okay before you get involved in the more vigorous testing.

Share Follow

edited Jun 1, 2020 at 12:58



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Feb 6, 2009 at 12:31



uriDium

13.4k ● 21 ● 80 ● 143

---

6 Unit testing is not necessarily white-box. Some of the best unit tests I've seen are essentially examples drawn from the requirements, specifying expected results independently from any implementation concepts. – [joel.neely](#) Feb 6, 2009 at 12:51

---

1 added to that, your unit tests are included in your regression tests therefore regression tests are neither white or black box tests. I'd go as far to say that even integration and smoke tests can be either white- or blackbox tests.  
– [Lieven Keersmaekers](#) Feb 6, 2009 at 13:29

---

1 I would have to disagree with this. Testing a design pattern implementation is a form of integration testing and is white box testing. – [Hazok](#) Dec 13, 2012 at 5:25

---

That answers the title, but not the one about tools for the last two types of tests, for *smoke testing* or *regression testing*. It also repeats previous answers - it could be made unique by answering the question about tools. – [Peter Mortensen](#) Jun 1, 2020 at 13:51

---



*Smoke tests have been explained here already and is simple. Regression tests come under integration tests.*

1

Automated tests can be divided into just two.



**Unit tests and integration tests** (this is all that matters)



I would call use the phrase "long test" (LT) for all tests like integration tests, functional tests, regression tests, UI tests, etc. And unit tests as "short test".



An LT example could be, automatically loading a web page, logging in to the account and buying a book. If the test passes it is more likely to run on live site the same way(hence the 'better sleep' reference). Long = distance between web page (start) and database (end).

And this is a great article discussing the benefits of [integration testing \(long test\) over unit testing](#).

Share Follow

edited Jun 1, 2020 at 13:57



Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Dec 9, 2017 at 8:45



Blue Clouds

8,123 ● 9 ● 79 ● 126



**Highly active question.** Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.