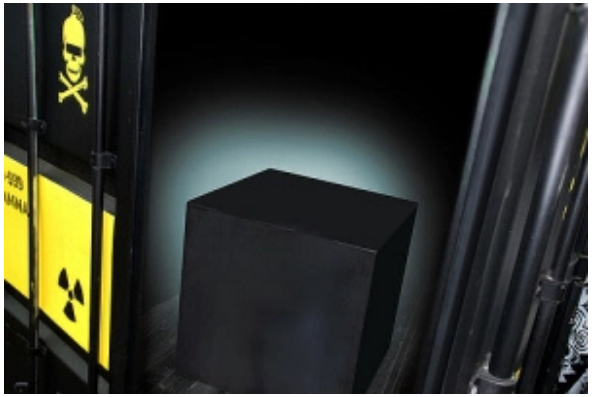
Technology Conversations

Black-box vs White-box Testing

> *Testing shows the presence, not the absence of bugs. Edsger W. Dijkstra*

Two common types of testing are black-box and white-box testing. Both can drive or be driven by development.

Black-box testing



Black-box testing (also known as functional testing) treats software under test as a black-box without knowing its internals. Tests are using software interfaces and trying to ensure that they work as expected. As long as functionality of interfaces remains unchanged, tests should pass even if internals are changed. Tester is aware of what the program should do but does not have the knowledge of how it does it. Black-box testing is most commonly used type of testing in traditional organizations that have testers as a separate department, especially when they are not proficient in coding and have difficulties to understand the code. It provides **external perspective** of the software under test.

Some of the **advantages of black-box testing** are:

1. Efficient for large segments of code
2. Code access is not required
3. Separation between user's and developer's perspectives

1. Limited coverage since only a fraction of test scenarios is performed
2. Inefficient testing due to tester's luck of knowledge about software internals
3. Blind coverage since tester has limited knowledge about the application

If tests are driving the development, they are often done in the form of acceptance criteria that is later used as definition of what should be developed. In that case black-box testing relies on some form of automation like **Behavior Driven Development**.

White-box testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) looks inside the software that is being tested and uses that knowledge as part of the testing process. If, for example, exception is thrown under certain conditions, test might want to reproduce those conditions. White-box testing requires internal knowledge of the system and programming skills. It provides **internal perspective** of the software under test.

Some of the **advantages of white-box testing** are:

1. Efficient in finding errors and problems
2. Required knowledge of internals of the software under test is beneficial for thorough testing
3. Allows finding hidden errors
4. Programmers introspection
5. Helps optimizing the code
6. Due to required internal knowledge of the software, maximum coverage is obtained

Some of the **disadvantages of white-box testing** are:

1. Might not find unimplemented or missing features
2. Requires high level knowledge of internals of the software under test
3. Requires code access

White-box testing is almost always automated and in most cases has the form of unit tests. If done before the development, it takes the form of Test Driven Development (TDD).

Quality Checking (QC) vs Quality Assurance (QA)

While Quality Checking is focused on defects identification, Quality Assurance tries to prevent them. QC is product oriented and intends to make sure that results are as expected. On the other hand, QA is more focused on processes that assure that quality is built in. It tries to make sure that correct things are done in the correct way. While QC had the more important role in the past, with the emergence of Test Driven Development (TDD), Acceptance Test Driven Development (ATDD) and, later on, Behavior Driven Development (BDD) focus has been shifting towards QA.

Conclusion

Both white and black-box testing are necessary for the successful software delivery. In many cases black-box testing is done by dedicated testers while white-box testing is performed by developers. Emergence of TDD, ATDD and BDD processes and supporting tools allows early defects detection and shifts the focus from QC towards QA.

Related articles

- Quality Assurance vs Quality Checking
- Behavior Driven Development (BDD): Value Through Collaboration (Part 1: Introduction)
- Behavior Driven Development (BDD): Value Through Collaboration (Part 2: Narrative)
- Behavior Driven Development (BDD): Value Through Collaboration (Part 3: Scenarios)
- Behavior Driven Development (BDD): Value Through Collaboration (Part 4: Automation)

photo credit: Abode of Chaos via photopin cc

This entry was posted in Testing and tagged BDD, Behavior-driven development, Black-box testing, Quality Assurance, Test-driven development, testing, White-box testing on December 11, 2013 [https://technologyconversations.com/2013/12/11/black-box-vs-white-box-testing/] by Viktor Farcic.

6 thoughts on “Black-box vs White-box Testing”

dofree
December 12, 2013 at 9:20 am

The best way to reduce bugs is to not focus on QA or QC but to remove safety nets and hold developers accountable to produce working code. I like to let my team know they will be taking support tickets directly from customers. When there are no safety nets, developers migrate towards test driven development and writing the simplest code possible to get the job done. QA and QC is a crutch not a solution.

ledasl
December 12, 2013 at 12:18 pm

Or they are distracted by customers, because having too many direct request (80% of client requests to fix bug, which is not bug in first place) and leaves company.

Viktor Farcic Post author
December 14, 2013 at 1:27 pm

The best way to reduce bugs depends on the type of organization and people working in it. In case of companies where majority are skilled with the code, holding developers accountable to produce working software is probably the best way to go. However, in many other companies there are separate testing departments. It is a common case that those departments have a lot of people without coding background. Assuming that the objective is to best utilize everyone and avoid letting them go, we must find a way for them to become more productive. They are often very skilled with black-box testing and are domain experts. They are valuable assets that might have been used for too long to do manual testing as the only way to approach the subject. They are QCs but they can be converted to QAs given enough time and dedication. BDD can be one of the answers to do the transition of those assets. In any case, the best approach is always to look for the way to utilize what you have in the best way possible and, at the same time, try to steer the direction towards future improvements.

Pingback: How To Do Black Box Testing Java | KH News

Pingback: White Hat vs Black Hat Hacker – Justin's Cyber Security Weekly

Pingback: Challenging Security Limitations: White vs. Black Box Testing & Real Risk – Briq Haus Ltd. Security & Intelligence