WIKIPEDIA

Black-box testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.[1]

Contents

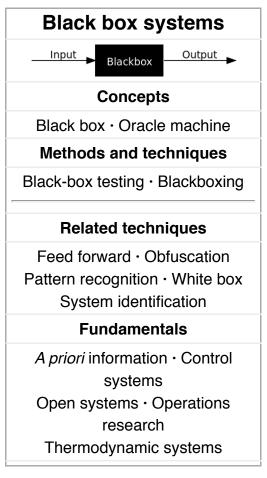| **Black box systems**		
Input → Blackbox → Output		
Concepts		
Black box · Oracle machine		
Methods and techniques		
Black-box testing · Blackboxing		
Related techniques		
Feed forward · Obfuscation Pattern recognition · White box System identification		
Fundamentals		
A priori information · Control systems Open systems · Operations research Thermodynamic systems		

Test procedures

Specific knowledge of the application's code, internal structure and programming knowledge in general is not required.[2] The tester is aware of *what* the software is supposed to do but is not aware of *how* it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of *how* the software produces the output in the first place.[3]

Test cases

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily *functional* in nature, *non-functional* tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of a test oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

Test design techniques

Typical black-box test design techniques include:

- Decision table testing
- All-pairs testing
- Equivalence partitioning
- Boundary value analysis
- Cause–effect graph
- Error guessing
- State transition testing
- Use case testing
- User story testing
- Domain analysis
- Syntax testing
- Combining technique

Hacking

In penetration testing, black-box testing refers to a methodology where an ethical hacker has no knowledge of the system being attacked. The goal of a black-box penetration test is to simulate an external hacking or cyber warfare attack.

See also

- ABX test
- Acceptance testing
- Blind experiment
- Boundary testing
- Fuzz testing
- Gray box testing
- Metasploit Project
- Sanity testing
- Smoke testing
- Software performance testing
- Software testing
- Stress testing
- Test automation
- Unit testing
- Web application security scanner
- White hat hacker
- White-box testing

References

1. Jerry Gao; H.-S. J. Tsao; Ye Wu (2003). *Testing and Quality Assurance for Component-based Software* (https://books .google.com/books?id=VoCX09hOsCoC&pg=PA170). Artech House. pp. 170–. ISBN 978-1-58053-735-3.
2. Milind G. Limaye (2009). *Software Testing* (https://books.google.com/books?id=zUm8My7SiakC&pg=PA216). Tata

McGraw-Hill Education. p. 216. ISBN 978-0-07-013990-9.

3. Patton, Ron (2005). *Software Testing* (2nd ed.). Indianapolis: Sams Publishing. ISBN 978-0672327988.

External links

- BCS SIGIST (British Computer Society Specialist Interest Group in Software Testing): *Standard for Software Component Testing* (http://www.testingstandards.co.uk/Component%20Testing.pdf), Working Draft 3.4, 27. April 2001.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Black-box_testing&oldid=850125401"

This page was last edited on 13 July 2018, at 19:34 (UTC).