Functional testing

Functional testing is a quality assurance (QA) process[1] and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing).[2] Functional testing is conducted to evaluate the compliance of a system or component with specified functional requirements.[3] Functional testing usually describes *what* the system does.

Since functional testing is a type of black-box testing, the software's functionality can be tested without knowing the internal workings of the software. This means that testers do not need to know programming languages or how the software has been implemented. This, in turn, could lead to reduced developer bias (or confirmation bias) in testing since the tester has not been involved in the software's development.[4]

Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system.

Functional testing differs from system testing in that functional testing "*verifies* a program by checking it against ... design document(s) or specification(s)", while system testing "*validate[s]* a program by checking it against the published user or system requirements."[5]

Contents

Types

Functional testing has many types:[2]

- Smoke testing
- Sanity testing
- Regression testing
- Usability testing

Six Steps

Functional testing typically involves six steps

1. The identification of functions that the software is expected to perform
2. The creation of input data based on the function's specifications
3. The determination of output based on the function's specifications

4. The execution of the test case
5. The comparison of actual and expected outputs
6. To check whether the application works as per the customer need.

See also

- Non-functional testing – Testing of computer software for the way it operates rather than specific behaviours or functions
- Acceptance testing – Test to determine if the requirements of a specification or contract are met
- Regression testing – Checking whether changes to software have broken functionality that used to work
- System testing – Testing conducted on a complete integrated software system
- Software testing – Examination of how efficient and reliable software is at performing its intended function
- Integration testing – Phase in software testing in which individual software modules are combined and tested as a group
- Unit testing – Software testing method by which individual units of source code are validated
- Database testing – The testing of database software systems
- Security testing – The process of finding flaws in the security of information systems
- Load testing – Process of putting demand on a system and measuring its response
- Test automation – Use of special software (separate from what is being tested) to control the execution of tests and compare actual outcomes with predicted outcomes

References

1. Prasad, Dr. K.V.K.K. (2008) *ISTQB Certification Study Guide*, Wiley, ISBN 978-81-7722-711-6, p. vi
2. Kaner, Falk, Nguyen. *Testing Computer Software*. Wiley Computer Publishing, 1999, p. 42. ISBN 0-471-35846-0.
3. *ISO/IEC/IEEE International Standard - Systems and software engineering*. ISO/IEC/IEEE 24765:2010(E). 2010. pp. vol., no., pp.1–418, 15 Dec. 2010.
4. Calikli, Gul; A. Uzundag, Berna; Bener, Ayse (September 19, 2010). "Confirmation Bias in Software Development and Testing: An Analysis of the Effects of Company Size, Experience and Reasoning Skills" (https://www.researchgate.net/publication/235430372_Confirmation_Bias_in_Software_Development_and_Testing_An_Analysis_of_the_Effects_of_Company_Size_Experience_and_Reasoning_Skills) – via ResearchGate.
5. Kaner, Falk, Nguyen 1999, p. 52