Xinyang wang

Master Report

The basic concepts and methods in software testing

2014SE3

**The basic concepts and methods in software testing**

1. The scope of the software quality   
(1) Accountability- the user can be based on the description and definition of a product or service for use. (For example: the market requirements specification, functional design specification.)  
(2) The Availability (effectiveness) - a product or service to 99.999% customer always effective (for example, performance test and recovery test)  
(3) the Accessibility (ease) - for users, the product or service is very easy to use and must be a very useful function. (For example: validation tests and user usability testing)  
The high quality software, should be relatively no defects (Bug Free) or only a tiny amount of defects, it can be delivered to the user and the cost is within budget and meet customer demand, can be maintained. However, the final evaluation by the quality of relies on the user feedback.  
   
2. Software quality standards  
Functionality  
Availability & Usability (simple installation, easy to use, friendly interface)  
Reliability (The fundamental for user)  
Performance  
Capacity   
Scalability  
Maintainability (Service manageability)  
Compatibility  
Extensibility  
   
3. The concept of software testing  
Software testing is an important means of quality control to ensure that the customer or user use high quality software products.  
Software testing is to find errors and execution process.  
A good test can detect errors existing in the program in the first time.  
A good test is found up to detect error that never be found in previous test.

4. The principle of software testing

(1) All testing standards are based on user requirements.  
(2) The software testing must be based on the idea of "quality first" to carry out the work, when time conflicts quality, time is subject to quality.  
(3) The standard of product quality is defined in advance, if only had the quality standards, according to the results of the tests, quality of product can be analyzed and evaluated.  
(4)When the software project launches, software testing also start, rather than waiting for program finished to test.  
(5) The exhaustive testing is proper and impossible. Even a modest program size, the number of its path arrangement is also very big, therefore, may not run in the test path of every kind of combination.  
(6) The third party test will be more objective, more effective.  
(7) Software test plan is premise to do a good job of software testing.  
(8) Test cases are designed, not written, which is according to the aim of the test, using the appropriate method to design test cases, thus to improve the efficiency of the test, to find more errors and improve the reliability of the program.  
(9) For those procedure sections with mass error, it should be further testing. In general, the more the number of errors has been found in a program, the more error probability.  
(10) Great importance to the document, and keep all test process documents (test plan, test cases and test reports, etc.)  
Note:  
A)"as soon as possible and constantly test" should be as the motto of the tester  
B) the correlation of regression testing must be sufficient attention at, modify a mistake and cause more errors appear phenomenon is not uncommon test, should be started "small" gradually turned to "massive".  
C) Pay attention to test case, eliminate the arbitrary.  
D) It must be thoroughly checked every test results.  
E) It is important to notice that the concentration of occurrence of error in the test, the programmer skills and habits may lead this.

F) The result of the test errors should have a confirmation process.

5. The test methods in common use   
**(1) black box testing**   
Any engineering products can be tested by one of the following two methods   
Black box testing: known product functional design specifications, test can prove that function is in accordance with the requirements of each implementation.  
White box testing: known internal working process of the product, can pass the test to prove each internal operation is in accordance with the design specifications, all internal components whether to pass inspection.  
   
Software black box testing means testing needs to be done in the software interface. This approach is to put the test object as a black box, testers do not consider internal logical structure and the internal characteristics of the program, only according to application requirements specifications, check whether the function of the program in accordance with its requirement. So the black box testing is called functional testing or data-driven testing. Black box testing is aimed to find the following errors:   
1, have any incorrect or missing of function  
2, on the interface, can it access and accept properly or not? Can it output the correct results?  
3, is there a data structure errors or external information (such as data files) access errors?  
4. Can performance meet the requirements?  
5, any initialization or termination error?

**Advantages:**

Well suited and efficient for large code segments.

Code Access not required.

Clearly separates user’s perspective from the developer’s perspective through visibly defined roles.

Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language or operating systems.

**Disadvantages:**

Limited Coverage since only a selected number of test scenarios are actually performed.

Inefficient testing, due to the fact that the tester only has limited knowledge about an application.

Blind Coverage, since the tester cannot target specific code segments or error prone areas.

The test cases are difficult to design.

**2. White box testing**   
Software white-box testing is a software to do careful inspection on the procedural details. This method regards the test object as an open box, it allows testers to use program internal logical structure and relevant information, design or select test cases, testing procedures all logical path. Through the check program status differences, to determine whether actual state in accordance with the desired state. So the white box test is also called structure test or logical drive test. White box testing is primarily to test the following program module for:  
   
1, all of the program module independent execution path at least test it again.  
2, for all the logical judgment, take "true" and take "fake" two cases can be measured at least once.  
3, in the boundary of the cycle and run within the boundaries of the loop body.  
4, testing the effectiveness of the internal data structures, and so on.

**Advantages:**

As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.

It helps in optimizing the code.

Extra lines of code can be removed which can bring in hidden defects.

Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.

**Disadvantages:**

Due to the fact that a skilled tester is needed to perform white box testing, the costs are increased.

Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems as many paths will go untested.

It is difficult to maintain white box testing as the use of specialized tools like code analyzers and debugging tools are required.

**3. Grey box testing**

Grey Box testing is a technique to test the application with limited knowledge of the internal workings of an application. In software testing, the term “the more you know the better” carries a lot of weight when testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black box testing, where the tester only tests the application’s user interface, in grey box testing, the tester has access to design documents and the database. Having this knowledge, the tester is able to better prepare test data and test scenarios when making the test plan.

**Advantages:**

Offers combined benefits of black box and white box testing wherever possible.

Grey box testers don’t rely on the source code; instead they rely on interface definition and functional specifications.

Based on the limited information available, a grey box tester can design excellent test scenarios especially around communication protocols and data type handling.

The test is done from the point of view of the user and not the designer.

**Disadvantages:**

Since the access to source code is not available, the ability to go over the code and test coverage is limited.

The tests can be redundant if the software designer has already run a test case.

Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.

**4** **static test & dynamic testing**

A static method is to point to not run the program itself to be tested, only through the analysis of the source or check the grammar of the program, structure, process and interface to check the correctness of the program. Static method is to detect error through checking specification, software design specifications, the structure of the source program to do analysis, chart analysis, symbolic execution. Static method through the application of static characteristic analysis, find out the deficiency and suspicious, such as unmatched parameters, the improper nested loop and branch nested, non-allowed recursion, unused variables, null pointer references and suspicious calculation, etc. Static test results can be used to further check the wrong, and provide guidance for test case selection.  
Dynamic test method is to point to by running the program is being tested to inspect the difference between examination results and expected results, and analyze the efficiency and robustness, such as performance, this method consists of three parts: the structure test cases, execute a program, analysis of the output of the program.

6. The process of the test

(1) Alpha test is usually beginning stage after completion of the development, continues to enter the in front of the Beta test stage. In this phase, potential customers/clients in the software or an independent test team, are using off-the-shelf software, to simulate or actual operational black box testing and the grey box testing to implement tests.

(2) After the completion of the development process the Alpha stage, test turn into the Beta phase. In Beta phase, part of products used for Beta testing is released to controlled external personnel, through this part of controlled external personnel test and feedback, existing in the Beta phase can try to find the product defects and mistakes. In some cases, the Beta version may be issued to a wider range of external personnel hands (for example, issued to the public by site or by other means).Beta testing mainly use black box testing technique. In Beta phase, of course, the tester can still use white box testing technology for products continue to test, it is usually not a part of Beta test. In simple terms, we think the Beta test is part controlled by customer for black box testing.

(3) Gamma test was a rarely mentioned informal testing, the testing phase is corresponding to "flawed" product test. Considering any products can be referred to as the "defect" products (test can only find product problem, cannot prove the product there is no problem), so this concept has certain uncertainty.

For one of Alpha and Beta testing common misunderstanding is "Beta = black box testing”. In fact, Alpha and Beta testing can be regarded as Alpha and Beta phase before the software product released, while the white box and black box and gray box testing technology is description from the technology and method level of the test, should not mix up two concepts.

**References:**

Fundamentals of software testing, 2012, Homè̀s, B, Homè̀s, Hoboken, NJ: ISTE/​Wiley, 2012.

Software testing fundamentals methods and metrics, 2003,​Marnie L. Hutcheson,

Indianapolis, Ind.; Effective software testing: 50 specific ways to improve your Testing, 2002, Elfriede D,Boston : Addison-Wesley, 2002

Software testing: principles and practice,2006, Srinivasan D, Gopalaswany R, Bangalore, India Dorling Kindersley (India) 2006.

Introduction to Software Testing, 2008, Paul A Jeff O,Cambridge University Press