MINI Edition Of Test

1. Bugs, Black box Testing and White box testing（test_1,test_2）

Bugs ： The term bug is often used to refer to a problem or fault in a computer. There are software bugs and hardware bugs

Black box Testing： Black Box Testing is testing without knowledge of the internal workings of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the "legal" inputs and what the expected outputs should be, but not how the program actually arrives at those outputs. It is because of this that black box testing can be considered testing with respect to the specifications, no other knowledge of the program is necessary.

Synonyms for black box testing: Behavioral, functional, opaque-box and closed-box.

***Advantage for black box testing:**

More effective on larger units of code than glass box testing

Tester needs no knowledge of implementation, including specific programming languages

Tester and programmer are independent of each other

Tests are done from a user's point of view

Will help to expose any ambiguities or inconsistencies in the specifications

Test cases can be designed as soon as the specifications are compile

*** Disadvantage for black box testing:**

. only a small number of possible inputs can actually be tested, to test every possible input stream would take nearly forever

. without clear and concise specifications, test cases are hard to design

. there may be unnecessary repetition of test inputs if the tester is not informed of test cases the programmer has already tried

. may leave many program paths untested

. cannot be directed toward specific segments of code which may be very complex (and therefore more error prone)

. most testing related research has been directed toward glass box testing

White box Testing: White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc.

Synonyms of white box testing: Glass, structural, open box or clear box testing.

Advantage of White box testing:

. As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.

. The other advantage of white box testing is that it helps in optimizing the code

. It helps in removing the extra lines of code, which can bring in hidden defects.

Disadvantage of White box testing:

. As knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, which increases the cost.

. It is nearly impossible to look into every bit of code to find out hidden errors, which may create problems, resulting in failure of the application.

2. Different Test Method

2.1 What's test case?

Answer: A Test Case is a document which consists of some inputs and expected outputs to see whether a particular feature of an application is working correctly or not.

2.2 Incremental integration testing

Answer:

- Continuous testing of an application as new functionality is added
- Requires that various aspects of an application's functionality be independent enough to work separately before all parts of the program are completed, or that test drivers be developed as needed
- Done by programmers or by testers.

2.3 End-to-end testing

Answer:

- Similar to system testing
- The 'macro' end of the test scale
- Involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate

2.4 Sanity testing

Answer:

- Typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort.
- For example, if the new software is crashing systems every 5 minutes, bogging down systems to a crawl, or destroying databases, the software may not be in a 'sane' enough condition to warrant further testing in its current state.

2.5 Regression Testing

Answer:

- Re-testing after fixes or modifications of the software or its environment.
- It can be difficult to determine how much re-testing is needed, especially near the end of the development cycle.
- Automated testing tools can be especially useful for this type of testing.

2.6 Load testing

Answer:

- Testing an application under heavy loads
- Testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.

2.7 Stress testing

Answer:

- Term often used interchangeably with 'load' and 'performance' testing.
- Also used to describe such tests as system functional testing while under unusually heavy loads, heavy repetition of certain actions or inputs, input of large numerical values, large complex queries to a database system, etc.

2.8 Performance testing

Answer:

- Term often used interchangeably with 'stress' and 'load' testing.
- Testing conducted to evaluate the compliance of a system or component with specified performance requirements. [IEEE]
- Ideally 'performance' testing is defined in requirements documentation or QA or Test Plans. s

2.9 Usability testing

Answer:

- Testing for 'user-friendliness'.
- Clearly this is subjective, and will depend on the targeted end-user or customer.
- User interviews, surveys, video recording of user sessions, and other techniques can be used.
- Programmers and testers are usually not appropriate as usability testers.

2.10 Install/uninstall testing

Answer:

- Testing of full, partial, or upgrade install/uninstall processes.

2.11 Recovery testing

Answer:

- Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

2.12 Security testing

Answer:

- Testing how well the system protects against unauthorized internal or external access, willful damage, etc;
- May require sophisticated testing techniques.

2.13 Compatibility testing

Answer:

- Testing how well software performs in a particular hardware/software/operating system/network/etc. environment.

2.14 Exploratory testing

Answer:

- Often taken to mean a creative, informal software test that is not based on formal test plans or test cases
- Testers may be learning the software as they test it.

2.15 Ad-hoc testing

Answer:

- Conducted for very short time.
- Similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it.

2.16 User acceptance testing

Answer:

Determining if software is satisfactory to an end-user or customer.

2.17 Comparison testing

Answer:

- Comparing software weaknesses and strengths to competing products.

2.18 Alpha testing

Answer:

- Testing of an application when development is nearing completion
- Minor design changes may still be made as a result of such testing.
- Typically done by end-users or others,
- At the developers site
- Some dummy data is taken

2.19 Beta Testing

Answer:

- Testing when development and testing are essentially completed and final bugs and problems need to be found before final release.
- Typically done by end-users or others, not by programmers or testers.
- At users site
- Some Live data is taken

2.20 More On Beta Testing

Answer:

- At user's premises, in absence of developers
- Some aspects
    - The number of beta test sites
    - The environment required
    - The support services to be provided
    - Whether beta test software is priced or free
    - Defect reporting mechanism
    - Beta testing period

2.21 Mutation testing

Answer:

- A method for determining if a set of test data or test cases is useful, by deliberately introducing various code changes ('bugs') and retesting with the original test data/cases to determine if the 'bugs' are detected.
- Proper implementation requires large computational resources.

2.22 Gorilla Testing

Answer:

- Imagine that software developed by team is given to a Gorilla for testing.
- The Gorilla will randomly press some keys and it may press some keys that are acceptable inputs.

2.23 Field Trial

Answer:

- Tested on real conditions before releasing in the market.
- Actual working environment testing

2.24 Accessibility Testing

Answer:

Testing carried out to ensure that the software can be used by the physically challenged persons.

2.25 Exhaustive Testing

Answer:

- Testing carried out by giving all possible inputs.
- Exhaustive testing is very time consuming, and is impossible to do for many large projects.

2.26 Smoke Testing

Answer:

- Testing carried out to check major functionality of the software without concern for finer details.
- Also referred to as quick and dirty testing.

2.27 Localization Testing

Answer:

- A product received from a software publisher has (supposedly) already been internationalized.
- Internationalization testing is done in order to determine how well internationalization has been done.
- E.g. will the product be easy to localize? Have all the localizable resources been separated from the source code? Does the software support Unicode

2.28 domain testing

Answer:

Domain testing is the most frequently described test technique. Some authors write only about domain testing when they write about test design. The basic notion is that you take the huge space of possible tests of an individual variable and subdivide it into subsets that are (in some way) equivalent. Then you test a representative from each subset.

2.29 Scenario testing

Answer:

Scenario tests are realistic, credible and motivating to stakeholders, challenging for the program and easy to evaluate for the tester. They provide meaningful combinations of functions and variables rather than the more artificial combinations you get with domain testing or combinatorial test design.

2.30 Old fix regression testing

Answer:

We retest several old bugs that were fixed, to see if they are back. (This is the classical notion of regression: the program has regressed to a bad state.)

2.31 General functional regression testing

Answer:

We retest the product broadly, including areas that worked before, to see whether more recent changes have destabilized working code. (This is the typical scope of automated regression testing.)

2.32 Conversion or port testing

Answer:

The program is ported to a new platform and a subset of the regression test suite is run to determine whether the port was successful. (Here, the main changes of interest might be in the new platform, rather than the modified old code.)

2.33 Configuration testing

Answer:

The program is run with a new device or on a new version of the operating system or in conjunction with a new application. This is like
port testing except that the underlying code hasn't been changed--only the external components that the software under test must interact with.

2.34 Volume testing

Answer:

Volume testing is done against the efficiency of the application. Huge amount of data is processed through the application (which is being tested) in order to check the extreme limitations of the system

3. Other Tests

3.1 System testing

In this the entire software is tested. The main objective of the system Testing is to see whether the software meets the specified requirements or not.

3.2 Acceptance Testing

This is done with the realistic data of the client to see whether it has met with the requirements of the user or not. Testing here focuses on the external behavior of the project rather than the internal logic.

3.3 Unit Testing

Different modules are tested against the specifications produced during
the designing of the modules. Unit Testing is essentially for Verification of the code produced during the coding phase. It is done by the programmer of the module.

3.4 Integration Testing

In this many module tested are combined into subsystems, which are then tested. The goal here is to determine the modules can be integrated properly. This Testing activity can be considered by testing the design.

4. Important concepts in PPT test_5

Static analysis involves going through the code in order to find out any possible defect in the code. **Dynamic analysis** involves executing the code and analyzing the output.

Statement Coverage

In this type of testing the code is executed in such a manner that every statement of the application is executed at least once. It helps in assuring that all the statements execute without any side effect.

5. Important concepts in PPT test_7

Equivalence partitioning

In this method the input domain data is divided into different equivalence data classes. This method is typically used to reduce the total number of test cases to a finite set of testable test cases, still covering maximum requirements.

Boundary value analysis

It's widely recognized that input values at the extreme ends of input domain cause more errors in system. More application errors occur at the boundaries of input domain. 'Boundary value analysis' testing technique is used to identify errors at boundaries rather than finding those exist in center of input domain.

Loop testing:

Loops are the basis of most algorithms implemented using software. However, often we do consider them when conducting testing. Loop testing is a white box testing approach that concentrates on the validity of loop constructs. Four loops can be defined: simple loops, concatenate loops, nested loops, and unstructured loops.

6.Short Notes

6.1 What is a test case?

Answer: A test case is a class which holds a number of test methods. For example if you want to test some methods of a class Book you create a class BookTest which extends the JUnit TestCase class and place your test methods in there.

6.2 What are Test suites?

Answer: If you have two tests and you'll run them together you could run the tests one at a time yourself, but you would quickly grow tired of that. Instead, JUnit provides an object TestSuite which runs any number of test cases together. The suite method is like a main method that is specialized to run tests.