# Software Testing Revision Questions 2018
# Part I -Theory Questions

In the context of software testing define the following terms: [6 marks]

i. Errors
ii. Faults
iii. Failures

"The objective of software testing is all about finding software faults". Can you list and explain 5 categories of software faults?

Explain any ***three*** of the following types of software faults:

a. Documentation faults
b. Stress or overload faults
c. Throughput or performance faults
d. Recovery faults
e. Syntax Faults

| **Software Fault** | **Description** |
|---|---|
| *Syntax Fault* | These are due to the nonconformity 不一致 of programming language, |
| *Documentation Faults* | Incomplete or incorrect documentation will lead to Documentation faults |
| *Stress of Overload faults* 压力过载故障 | These happen when data structures are filled past their specific capacity whereas the system characteristics are designed to handle no more than a maximum load planned under the requirements |
| *Throughput or performance faults*<br>吞吐量和性能故障 | This is when the developed system does not perform at the speed specified under the stipulated 规定的 requirements |
| *Recovery faults* | This happens when the system does not recover to the expected performance even after a fault is detected and corrected<br>当系统无法恢复到预期的性能时 |

In the context of examining the failure curve and the design of testing methods give ***three*** important differences between hardware and software.

'The classification of failures is more difficult than faults. One approach is to classify a failure by its level on a severity scale' Give an outline description of a scale that has *four* different levels of Failure Severity.

Explain the approach to software development that is known as ‘Big Bang’. Give two drawbacks to using it.
1

What advantages does incremental software testing over the Big Bang approach (i.e. testing all at once)?
5

Can software be tested exhaustively?

Why is Exhaustive Testing not recommended as an approach for software testing?

Give one sentence to explain each of these.

Deciding when the testing of a piece of software should finish can be difficult to judge. Can you suggest two different criteria or benchmarks that could be useful to a Software Testing manager when making that decision?
1)A budgetary criteria 预算标准: when the time or budget allocated has expired
2)An activity criteria: when the software has passed all of the planned tests
当软件已经通过了所有的计划测试。
3)A risk management criteria 风险管理标准: when the predicted failure rate meets some quality criteria 当预测的失败率达到一定的质量标准。

Finishing testing can take three points of view: [3 marks]

i. Budgetary
ii. Activity
iii. Risk management

Describe the differences between static and dynamic testing.
（1）静态测试：静态测试是指不运行被测程序本身，通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。其被测对象是各种与软件相关的有必要进行测试的产物，是对需求规格说明书、软件设计说明书、源程序做结构分析、流程图分析、符号执行来找错。静态测试可以手工进行，充分发挥人的思维的优势，并且不需要特别的条件，容易展开，但是静态测试对测试人员的要求较高，至少测试人员需要具有编程经验。
静态测试包含的内容：
静态测试主要包括各阶段的评审、代码检查、程序分析、软件质量度量等，用于对被测程序进行特性分析。其中评审通常有人来执行；代码检查程序分析、软件质量度量等即可人工完成，也可用工具来完成，但工具的作用和效果相对更大更好一些。
（2）动态测试：通过运行被测程序来检查运行结果与预期结果的差异；

Give a short (*3 lines only*) definition of the following Software Testing activities

i. Unit Testing
ii. Integration Testing
iii. System Testing
iv. Acceptance Testing

**Unit Testing** An individual unit of software is tested to make sure it works correctly. This unit may be a single component, or a compound component formed from multiple individual components. Unit testing almost invariably 总是 makes use of the programming interface of the unit.

**System Testing** The entire software system is tested as a whole to make sure it works correctly. The testing uses the system interface: this may be a Graphical User Interface for an application (GUI), web interface for a web-server, network interface for a network server, etc.

Integration 1
System 1
Acceptance 9
Using diagrams to illustrate your answer, explain the procedures for

(i) Top-down integration testing
and
(ii) Bottom-up integration testing.
2, 3
Drivers and Stubs are important components in the implementation of the Integration testing methods of (i) and (ii). Give definitions for what a Driver is and what a Stub is.
1, 2

Give a practical example that describes how a Driver and Stub might be used in the testing of a piece of software. Use a diagram to illustrate your answer.
1

In the context of the architecture of the XUnit test framework give *one-line* definitions of the following

- Test runner
- Test fixtures
- Test suites
- Test Result

Explain three differences between Black-box testing and White-box testing.

| Black box | White box |
|---|---|
| - Test against the specification<br>- Test cases derived from specification<br>- Tests can be reused if the code is updated or additional | - Test against the implementation<br>- Test cases derived from the source code<br>- Tests are generally invalidated by any changes to the code |

| | |
|---|---|
| functionality is added | |

Using an example, show what the difference is between an Error of Commission and Error of Omission as they can occur in Black and White box testing respectively.

Define the following two terms and say which one is associated with Black box testing and which with White box testing.

i. Errors of Commission
   and
ii. Errors of Omission

**Error** of omission -- a transaction that is not recorded.
**Error of commission** -- a transaction that is calculated incorrectly.

- Black-box tests
  - can be designed earlier
  - are better at catching errors of omission and errors in design
- White-box tests
  - must await the development of the code
  - are better at catching errors of commission and errors after design

What is the relationship between Equivalence Partitioning and Boundary value analysis?

Under which circumstances is Combinational testing using Truth Tables useful?

One Black Box testing method is known as Error Guessing. Explain ***three*** typical errors that would be checked for in this test technique. [3 marks]
21
Give *two* strengths and *two* weaknesses of the Black Box test technique of Random Testing.
20
Give *one* of the strengths and one of the weaknesses of *one* of the following White-box and Black-box software testing methods:

i. Error Guessing
   or
ii. Random Testing

Error 22
Random 20

What is a control flow graph (CFGs)? Draw sample CFGs illustrating their appearance for a Switch statement and a While loop.

A ‘data-flow’ approach to testing is a way of looking at a program as a flow of data from one statement to another. The motivation is to find data flow anomalies. Give explanations of three types of anomalies with short examples.

Give one of the strengths and one of the weaknesses of *five* of the following White-box and Black-box software testing methods:

iii. Statement Testing
iv. Branch Testing
v. Path Testing
vi. Equivalence Partitioning
vii. Boundary Value analysis
viii. Combinational Testing/Truth tables/Cause-Effect Graphing

| **Test Type** | **Strength** | **Weakness** |
|---|---|---|
| Statement | 1)Provides a minimum level of coverage by executing all statements at least once | 1)May be hard to test code that can only be executed in dangerous circumstances 危险环境<br>2)Will not test unreachable code<br>3)does not provide coverage for ‘NULL else’ conditions 不提供“空 else”条件的覆盖率<br>4)Not demanding of compound logical conditions 不要求复合逻辑条件 |
| Branch | 1)Branch coverage ensures that each “decision” is tested at least once.<br>2)100% branch coverage guarantees 100% statement coverage – but the test data is harder to generate. | 1)Branch coverage still doesn’t exercise either all the reasons for taking each branch, or combinations of different branches taken. 仍然没有充分利用每个分支的原因，或者不同分支的组合。 |
| Path Testing | 1)It generates test data in the pattern that data is manipulated in the program rather than following abstract branches<br>2)A strong form of testing | 1)Number of test cases can be very large<br>2)Difficult to apply it to pointer variables<br>3)Difficult to apply it to arrays |
| Equivalence Partitioning | 1)Provides a good basic level of testing.<br>2)Well suited to data processing applications where input variables may | 1)Correct processing at the edges of partitions is not tested.<br>2)Combinations of inputs are not tested.<br>3)The technique does not provide an |

| | | |
|---|---|---|
| | be easily identified and take on distinct values.<br>3)Provides a structured means for identifying basic Test Cases. | algorithm for finding the partitions or selecting the test data. |
| Boundary Value Analysis | Gives more guidance in test data creation<br>Test data focused on areas where faults are more likely to be found | Combinations of input partitions have not been tested<br>输入分区的组合没有测试。 |
| Combinational Testing | Exercises combinations of test data<br>Expected outputs are created as part of the process | The truth tables can sometime be very large. The solution is to identify sub-problems, and develop separate tables for each.<br>Very dependent on the quality of the specification - more detail means more causes and effects, which takes more time to test; less detail means less causes and effects, but less effective testing |

State the difference between Software verification and Software validation in the context of software testing.
Verification is confirming that we have built the software correctly while Validation confirms that we have built the correct software.

Give *three* documents that would be considered as part of a typical Test Plan according to the IEEE model.
Test Design specification test case specification procedure specification 2

Explain two advantages and two problems associated with software testing within the traditional Waterfall software process model.
2, 3
“The V-model of software development emphasizes the verification and validation of software throughout the development process”. With the aid of an appropriate illustration, can you show and describe how this process is carried out at the various test stages from Unit testing through to Acceptance testing?
Diagram? Describe?

Discuss the advantages and disadvantages of using the V-model for software development?

4

The Incremental development model of Software Development is an 'Agile' method. Draw a diagram that illustrates how the Incremental development model works. Give *one* advantage and *one* disadvantage of it.

4, 5

Draw a diagram of the XP, Extreme Programming, software development process. Ensure to highlight in your diagram the points at which software testing is carried out.

Extreme Programming, also known as XP, is a Software Development model that relies on the use of Story Cards. Explain what they are and create an example of one to illustrate how they might appear.

For the SCRUM approach to managing software development explain the following two terms. Also, sketch a diagram to show their relationship to each other.

i. Product Backlog

ii. Sprint and Sprint Backlog

6, 7

Give one advantage and one disadvantage of the DevOps method of Software development. Briefly explain how DevOps differs from other techniques.

7\8