**Đề 2\_đáp án**

1. Which statement about testing is true? …[K1]

1. Testing is started after the code is written so that we have a system with which to work.
2. **Testing is started as early as possible in the life cycle.**
3. Testing is most economically done at the end of the life cycle.
4. Testing can only be done by an independent test team.

2. The risk of an undetected defect … [K2]

1. … does not influence testing in any way.
2. **… has influence on test comprehensiveness and intensity.**
3. … has no influence on the specification of the test cases.
4. … has no influence on the prioritization of the tests.

3. During testing, … [K2]

1. … all parts of the system should be tested with the same intensity, because defects can be everywhere.
2. … the user interface should primarily be tested, because failures in it are most annoying for the customer.
3. **… system parts where failures may cause great risks should be tested most intensively.**
4. … data base access should be tested intensively, in order to prevent wrong data and inconsistencies an the database.

4. According to ISTQB syllabus, A software error can be described as…[K1]

1. **A mismatch between the program and its specification.**
2. A description of the relationship between two or more variables or set members in which the values of one does not influence the values of others.
3. The process in which developers determinate the root cause of a bug and identify possible fixes.
4. Any ill-advised, substandard, or temporary fix applied to an urgent problem in the (often misguided) belief that doing so will keep a project moving forward.

5. Deciding how much testing is enough should take account of followings except? [K2]

1. Project constraints
2. Level of Risk
3. **Decision Table**
4. Safety

6. Which of the following Is NOT “ Test Analysis and Design” activity in “Fundamental Test Process” [K2]

1. Identifying necessary test data to support the test conditions and test cases
2. Identifying and prioritizing test conditions based on analyses of test items, the specification, behavior and structure of the software.
3. Creating bi-directional traceability between test basic and test cases.
4. **Creating test suites from the test procedures for efficient test execution.**

7. Which is not a goal of writing effective Problem/Bug report?[K2]

1. Explain how to reproduce the problem.
2. Analyze the error so you can describe it in a minimum number of steps.
3. Write a report that is complete, easy to understand, and non-antagonistic
4. **Illustrate how to fix the problem**

Question about “Testing in the Software Lifecycle”

8. Which of the following is true of the V-model …. [K2]

1. … it only models the test phase
2. … it specifies the test techniques to be used
3. **… it includes the verification of design**
4. … it only states that testing against user requirement

9. Dynamic and--------------approaches, such as exploratory testing were testing is more reactive to events than pre-planned, and where execution and evaluation are concurrent tasks…[K2]

1. **Heuristic**
2. Exploratory
3. Regression-averse
4. Analytical

10. Testing according to the V-model has the following fundamental principles [K2]

1. For every development level, there must be a corresponding test level.
2. **For every development level, there must be a corresponding test level. For every test level early test design should be implemented.**
3. There must a least be the following test levels: Component testing, integration testing, system testing and acceptance testing.
4. The model is only useful for large critical software projects where there is one delivery only.

11. Integration testing has this fundamental purpose: [K2]

1. Testing the new system together with the systems existing before.
2. Testing components together in a hierarchical, incremental way like top-down or bottom-up.
3. Testing the client part together with different other layers of the software (like for example, application server code, database server code etc.).
4. **Testing interfaces between components, interactions between different parts of the system, as well as interfaces between systems.**

12. Negative test is … [K1]

1. … Test aimed at showing that a component or system does work.
2. **… Tests aimed at showing that a component or system does not work.**
3. … Not related to the testers’ attitude.
4. … Related to a specific test approach or test design technique.

13. Which of these is a functional test? … [K1]

1. Measuring response time on an on-line booking system.
2. Checking the effect of high-volumes of traffic in a call-center system.
3. **Checking the online bookings screen information and the database contents against the information on the letter to the customers.**
4. Checking how easy the system is to use.

14. Acceptance testing may occur at more than just a single test level. With the exception of…[K3]

1. **Acceptance testing after a change has been released to the user community.**
2. Acceptance testing of the usability of a component may be done during component testing.
3. Acceptance testing of a new functional enhancement may come before system testing.
4. A COST software product may be acceptance tested when it installed or integrated.

15. Non-functional testing includes … [K1]

1. Testing to see where the system does not function correctly.
2. **Testing the quality attributes of the system including reliability and usability.**
3. Gaining user approval for the system.
4. Testing a system feature using only the software required for that function.

Questions about “Static Testing”

16. What is the fundamental difference between “static analysis” and test? [K2]

1. Static analysis can be applied at any time during the software life cycle. Test can only be applied after implementation.
2. **Static analysis is done without executing the test object. A test is an examination during “run time”.**
3. Static analysis is applied only in the early phases, test only in the late phases of the software life cycle.
4. Static analysis checks only documents without a formal structure. Test checks software systems.

17. Which of the following characteristics and types of review processes belong together? … [K2]

a. Inspection

b. Informal review

c. Peer review

d. Walkthrough

1. Led by the author

2. Undocumented

3. Uses entry and exit criteria

4. Review by peers

1. a=3, b=2, c=1, d=4
2. a=2, b=3, c=4, d=1
3. **a=3, b=2, c=4, d=1**
4. a=1, b=2, c=3, d=4

18. Which of the following options are roles in a formal review?

a) Developer, Moderator, Review leader, Reviewer, Tester.

b) Author, Moderator, Manager, Reviewer, Developer.

c) Author, Manager, Review leader, Reviewer, Designer.

**d) Author, Moderator, Review leader, Reviewer, Scribe.**

19. Which is the most formal kind of review in the following list? [K2]

1. Peer review
2. Walkthrough
3. Informal review
4. **Inspection**

20. Static code analysis typically identifies all but one the following problems. Which is it? [K1]

1. Unreachable code.
2. Undeclared variables.
3. **Faults in the requirements.**
4. Too few comments.

21. Which of the following sequences BEST shows the main activities of the work product review process?

a) Initiate review – Reviewer selection – Individual review – Issue communication and analysis – Rework

b) Planning & preparation – Overview meeting – Individual review – Fix– Report

c) Preparation – Issue Detection – Issue communication and analysis – Rework – Report

**d) Plan – Initiate review – Individual review – Issue communication and analysis – Fix defects & report**

22. Static analysis … [K2]

1. … serves for creating test cases
2. … is the precondition for executing the dynamic analysis
3. … is the check of the program code for compliance with the documentation rules
4. **… is an analysis without execution of the test object (under analysis)**

23. Which alternative contains ONLY defects which can be discovered by static analysis … [K2]

1. Referencing a variable with an undefined value, unreachable (dead) code, security vulnerabilities, race conditions.
2. **Violations of coding standards, referencing a variable with an undefined value, unreachable (dead) code, security vulnerabilities.**
3. Referencing a variable with an undefined value, deadlocks, unreachable (dead) code, security vulnerabilities.
4. Referencing a variable with an undefined value, unreachable (dead) code, inconsistent interface between modules and components, race conditions.

24. How is “expected result” defined? [K1]

1. **The behavior predicted by the specification, or another sources, of the test object when executed under specified conditions.**
2. Intended output and/or state value when executing a test object with test data.
3. The intended behavior and output of the test object when executed under specified conditions.
4. The output result computed by the test object.

25. In which document described in IEEE 29119 (IEEE 829) would you find instructions for the steps to be taken for a test including set-up, logging, environment and measurement? … [K2]

1. Test plan.
2. Test design specification.
3. Test case specification.
4. **Test procedure specification.**

26. White Box techniques … [K1]

1. … are useful in system testing
2. … check the interfaces of component
3. **… take into consideration the structure of the test object**
4. … find if not initials are used

27. Which test technique is a black box technique? [K1]

1. Condition coverage
2. **Equivalence class partitioning**
3. Statement coverage
4. Data flow testing

28. If a test of a test object achieved 100% decision coverage then it is guaranteed that … [K3]

1. … 50% path coverage has been achieved
2. **… 100% statement coverage has been achieved**
3. … 100% minimal condition coverage has been achieved
4. … no other coverage has been achieved

29. Which of the following is usually not be a coverage measure for state transition testing? [K2]

1. All states have been reached.
2. **The response time for each transaction is adequate.**
3. Every transition has been exercised.
4. Specific sequences of transitions have been exercised.

30. The following equivalence class for integer numbers is given: 0 ≤ x < 100. After boundary value analysis, the following test data shall be used: [K3]

1. 0; 100
2. 0; 1; 100; 101
3. **-1; 0; 99; 100**
4. -1; 0; 1; 99; 100

31. How do experience-based techniques differ from specification-based techniques? …[K2]

1. They depend on the tester’s understanding of the way the system is structured rather than on a documented record of what the system should do.
2. They depend on having older testers rather than younger testers.
3. They depend on a documented record of what the system should do rather than on an individual’s personal view.
4. **They depend on an individual’s personal view rather than on a documented record of what the system should do.**

32. Use case testing is useful for which of the following?

A. Designing acceptance tests with users or customers.

B. Making sure that the mainstream business processes are tested.

C. Finding defects in the interaction between components.

D. Identifying the maximum and minimum values for every input field.

E. Identifying the percentage of statements exercised by sets of tests.

1. **A, B and C**
2. B, D and E
3. A, B and D
4. C, D and E

33. How many test cases are necessary in order to achieve 100% branch or decision coverage of the following program part? We assume that the two conditions are independent of each other. [K3]

….

If (condition 1)

then statement 1

else statement 2

end if

If (condition 2)

then statement 3

end if

…

1. **2 test cases**
2. 3 test cases
3. 4 test cases
4. Not achievable

34. Which of following is the repeated testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the change? [K3]

1. **Regression Testing**
2. Structural Testing
3. Functional Testing
4. Non-Functional Testing

35. Non-functional requirements … [K1]

1. … are always documented in detail in the specification by the customers.
2. … need only to be considered in the test if they are documented in the requirements specification.
3. … should only be validated by the customer, because only the customer can decide if they fulfill his requirements.
4. **… are often overlooked when writing specification documents, but should be checked anyway.**

Questions about “Test Management”

36. Incident management deals with…[K2]

A. …. Management of deviations from the project plan.

B. … Registration of problems, faults and failures.

C. … Showing product quality through use of metrics.

D. … Definition and implementation of new software requirements.

E. … Development of proposals to correct faults.

1. A, B and E are true, C and D are false
2. A, C and D are true, B and E are false
3. A, B and C are true, D and E are false
4. **B and C are true, A, D and E are false**

Question about “Test Tools”

37. What is a potential risk in using tools to support testing? … [K2]

1. **Unrealistic expectations, expecting the tool to do too much.**
2. Insufficient reliance on the tool, i.e. still doing manual testing when a test execution tool has been purchased.
3. The tool many find defects that aren’t there.
4. The tool will repeat exactly the same thing it did the previous time.

38. Comparators belong to this category of tools … [K2]

1. … Tools for planning and test management
2. … Tools for test specification and test design
3. **… Tools for test execution**
4. … Tools for test results analysis and test object analysis

39. The selection of a test tool should be done in six steps: [K1]

1. Requirement specification, Evaluation, Market research, Tool demonstrations, Review, Final selection of tool.
2. **Requirement specification, Market research, Evaluation, Tool demonstrations, Review, Final selection of tool.**
3. Requirement specification, Market research, Review, Tool demonstrations, Evaluation, Final selection of tool.
4. Requirement specification, Market research, Tool demonstrations, Evaluation, Review, Final selection of tool.

40. Which of the following would NOT be done as part of selecting a tool for an organization? …[K2]

1. Assess organization maturity, strengths and weaknesses.
2. **Roll out the tool to as many users as possible within the organization.**
3. Evaluate the tool features against clear requirements and objective criteria.
4. Identify internal requirements for coaching and mentoring in the use of the tool.