ISTQB®

Certified Tester
Foundation Level

CTFL 4.0

Chapter 2

– Summary –

– Questions & Answers –

– Exam Questions Distribution –

Swipe for more

Karim Kalboussi

Examinable Learning Objectives :

Level 1 : Remember (K1)

- The candidate will remember, recognize and recall a term or concept.
- Action verbs : Identify, recall, remember, recognize.
- Exmaple : Identify typical test objectives.

Level 2 : Understand (K2)

- The candidate can select the reasons or explanations for statements related to the topic, and can summerize, compare, classify and give examples for the testing concept.
- Action verbs : Classify, compare, contrast, differentiate, ditinguish...
- Exmaple : Explain the activities of the review process.

Level 3 : Apply (K3)

- The candidate can carry out a procedure when confronted with a familiar task, or select the correct procedure and apply it to a given context.
- Action verbs : Apply, implement, prepare, use.
- Exmaple : Apply test case prioritization.

Chapter 2 Question Distribution in the Exam :

- There is a total of **6 questions** required for Chapter 2 :
  K1 = 2 questions
  K2 = 4 questions
  K3 = 0 questions
- Number of points for this chapter = 6

Question Distribution	K-Level	Number of Questions per LO (group)*	Suggested Points per Question	Probability of Appearance in the exam
Chapter 2				
FL-2.1.2	K1	1	1	1
FL-2.1.3	K1	1	1	1
FL-2.2.1	K2	1	1	0.5
FL-2.2.2				0.5
FL-2.2.3	K2	1	1	0.5
FL-2.3.1				0.5
FL-2.1.1	K2	1	1	0.5
FL-2.1.6				0.5
FL-2.1.4	K2	1	1	0.5
FL-2.1.5				0.5

Summary of Chapter 2

2.1 Testing in the Context of a Software Development Lifecycle

2.1.1 (K2) Explain the impact of the chodden SDLC on tessting

- The choice of the SDLC impats on the : scope and timing of test activities (test levels and test types), level of detail of test documentation, choice of test techniques and test approach, extent of test automation, role & responsibilities of a tester.
- In sequential development models (like V-model), in the initial phases testers do the requirements reviews, test analysis, and test design. Dynamic testing cannot be performed early in the SDLC.
- In some iterative and incremental models, where each iteration delivers a working product, both static and dynamic testing may be performed at all test levels.
- In agile, extensive test automation is needed to make regression testing easier and most of the manual testing tends to be done using experience-based test techniques.

2.1.2 (K1) Recall good testing practices that apply to all SDLC

- For every software development activity, there is a corresponding test activity.
- Each test level has specific and distinct objectives.
- Testers are involved in reviewing work products as soon as drafts of the relevant work become available.
- Test analysis and design for a given test level begins during the corresponding development phase of the SDLC.

2.1.3 (K1) Recall the examples of test-first approaches to development

- TDD (Test-Driven Development), ATDD (Acceptance Test-Driven Development) and BDD (Behavior-Driven Development) all implement the principle of early testing and follows shift-left approach. They support iterative models.

- **Test-Driven Development :** Directs code through test cases, tests are written first, then the code is written to satisfy the tests, and then the tests and code are refactored.
- **Acceptance Test-Driven Development :** Test cases are created based on acceptance criteria to drive the development of the related software. Tests are written before the part of the application is developed to satisfy the tests.
- **Behavior-Driven Development :** The desired behaviour (or acceptance criteria) of the application is written in a simple form of natural language (Given/When/Then). Test cases are then automatically translated into executable tests.

2.1.4 (K2) Summarize how DevOps might have an impact on testing

- DevOps provides fast feedback on code quality and automated regression testing that minimizes regression risk.
- DevOps increase the view on non-functional quality characteristics (performance, reliability).
- CI (Continuous Intergration) promotes shift-left approach by encouraging developers to submit high quality code accompanied by component tests and static analysis.
- DevOps promotes automated processes like CI/CD that faciliate stable test environments.
- Automation through a delivery pipeline reduces the need for repetitive manual testing.
- DevOps delivery pipeline must be defined and established. ⚠
- CI/CD tools must be intotuced and maintained. ⚠

2.1.5 (K2) Explain the Shift-Left approach

Early testing or shift-left approach suggests that testing should be done earlier.
Good practices that illustrate how to achieve a “dhift-left” in testing are :
- Early review of the user requirements and reviewing the specifications from the perspective of testing.
- Writing test cases before the code is written and run code in a test harness.
- Using Ci/CD to accompany source code when it is submitted to the code repository.
- Completing static analysis of code prior to dynamic testing.
- Performing non-functional testing starting at the component test level, where possible.

2.1.6 (K2) Explain how retrospectives can be used as a mechanism for process improvement

- Process weakness identified during the retrospectives can be analyzed and serve as a todo list for the organization’s continious process improvement program.
- Retrospectives give testers an opportunity to identify activities that were successful so that these are retained when potenitial improvements are made in the future.
- The quality of future test objects improves by identifying improvements in development practices.
- Test efficiency improves by speeding up the configuration of test environment through automation.
- Automated test scipts are enhanced through fedback from developers.

2.2 Test Levels and Test Types

2.2.1 (K2) Distinguish the different test levels

- **Component testing** : (Unit testing) Focuses on testing components in isolation, it is normally performed by developers in their development environment.
- **Component integration testing :** (Unit integration testing) Focuses on testing the interfaces and interactions between components. Component intergration testing depends on the integration strategy approach (Bottom-Up, Top-Down or Bin-Bang).
- **System testing :** Focuses on the overall behavior of an entire product or system, also includes functional testing of end-to-end tasks and non-functional testing of quality characteristics.
- **System integration testing :** Focuses on testing the interfaces of the system under test and other systems and external services.
- **Acceptance testing :** Focuses on validation and wether the system fulfills the user's buisness needs. Acceptance testing should be performed by the intended users. Acceptance testing forms are : user acceptance testing (UAT), operational acceptance testing, contractual acceptance testing and regulatory acceptance testing, alpha testing and beta testing.

2.2.2 (K2) Distinguish the different test types

- **Functional testing** : Evaluates "What" the test object should do. It evaluates the functions that a component or system should perform. Functional testing is about checking the functional completeness.
- **Non-Functional testing :** It is the testing of "How well the system behaves". The main objective is checking the non-functional software quality characteristics like performance, compatibility, usability, reliability, security, maintainability, portability...
- **Black-Box testing :** Specification-based, derives tests from the documentation external to the test object. The main objective is checking the system's behavior against its specifications.
- **White-Box testing :** Structure-based, derives tests from the system's implementation or internal structure. The main objective is to cover the underlying structure by the tests to the acceptance level.

2.2.3 (K2) Distinguish confirmation testing from regression testing

- **Confirmation testing** : Confirms that an original defect has been successfully fixed, by either executing all test cases that previously failed due to the defect, or by adding new tests to cover any changes needed to fix the defect.
- **Regression testing :** Confirms that no adverse consequences have been caused by a change, including a fix that has already been confirmation tested.

2.3 Maintenance Testing

2.3.1 (K2) Summarize maintenance testing and its triggers

- Testing the changes to a system in production includes both evaluating the success of the implementation of the change and the checking for possible regressions in unchanged parts of the system. The scope of maintenance testing depends on the degree of risk of the change, the size of the existing system and the size of the change.
- The triggers for maintenance and maintenance testing can be :

- modification, like planned enhancements, corrective changes or hot fixes.
- Upgrades or migrations of the operational environment.
- Retirement (decommissioning), such as when an application reaches the end of its life. When a system is retired, this can require testing of data-archiving (data migration testing). Testing of restore and retrieval procedures after archiving may also be needed. ⚠

Swipe for the questions part

Questions from Chapter 2 in the ISTQB exam

2.1.1 (K2) Explain the impact of the chodden SDLC on tessting

Which of the following statements about the chosen software development lifecycle is CORRECT?

a) If agile software development is used, system test automation replaces the need for regression testing
b) If a sequential development model is used, then the dynamic testing is typically restricted to later in the lifecycle
c) If an iterative development model is used, then component testing is typically performed manually by developers
d) If an incremental development model is used, then static testing is done in early increments and dynamic testing in later increments

Select ONE option.

a) Is not correct. In agile software development, deliverables are produced in each iteration, and the frequent delivery of increments necessitates extensive regression testing. Although some (or all) of this regression testing may be automated, the regression testing (automated or not) cannot be replaced by system test automation b) Is correct. If a sequential development model is used, then early in the lifecycle no code is available for execution, and so during this time static testing (e.g., reviews) is performed. Later in the lifecycle, when code is available for execution, dynamic testing is possible. Note, however, that preparation for dynamic testing will often occur early in any software development lifecycle c) Is not correct. If an iterative development model, like agile software development, is used, then component tests may well be used for regression testing for each iteration. In which case, there is a strong argument for automating these component tests, which will have to be run frequently, and there is unlikely to be a strong argument for developers performing these component tests manually d) Is not correct. In most incremental development models, deliverables are produced in each increment, requiring both static and dynamic testing at all test levels for each increment delivered	FL-2.1.1

In an iterative lifecycle model, which of the following is an accurate statement about testing activities?

a. For every development activity, there should be a corresponding testing activity
b. For every testing activity, appropriate documentation should be produced, versioned and stored
c. For every development activity resulting in code, there should be a testing activity to document test cases
d. For every testing activity, metrics should be recorded and posted to a metrics dashboard for all stakeholders

A is correct. For any lifecycle model, this is a correct statement.

B is not correct because some testing activities may not produce documentation, such as reviews.

C is not correct because test cases are not always written, particularly in an Agile lifecycle (which is an iterative lifecycle) where only exploratory testing might be used.

D is not correct because not all testing activities produce metrics (such as test case creation, reviews, etc.) and, even if they did, not all stakeholders would be interested in those metrics.

2.1.2 (K1) Recall good testing practices that apply to all SDLC

Consider the following rule: “for every SDLC activity there is a corresponding test activity”. In which SDLC models does this rule hold?

a) Only in sequential SDLC models
b) Only in iterative SDLC models
c) Only in iterative and incremental SDLC models
d) In sequential, incremental, and iterative SDLC models

Select ONE option.

a) Is not correct; b) Is not correct; c) Is not correct; d) Is correct; this rule holds for all SDLC models	FL-2.1.2

Which of the following is a good testing practice that applies to all software development lifecycles?

a) Each test level has specific and distinct test objectives
b) Test implementation and execution for a given test level should start during the corresponding development phase
c) Testers should start test design as soon as drafts of the relevant work products become available
d) Every dynamic testing activity has a corresponding static testing activity

Select ONE option.

a) Is correct. Each test level has specific and distinct test objectives as a different form of test object (e.g., single component, complete system) is tested at each test level and overlapping test objectives would lead to unnecessary duplication b) Is not correct. Test analysis and design for a given test level should start during the corresponding development phase to facilitate early testing (e.g., acceptance test analysis and design should begin during requirements analysis). Test implementation will generally start later, and test execution will start during the test level c) Is not correct. Test design for a given test level should start during the corresponding development phase to facilitate early testing, however test design (e.g., test case generation) needs to be based on an agreed test basis, not an early draft, otherwise significant test effort may be wasted on creating test cases for a design that later changes d) Is not correct. Quality control applies to all development activities, meaning that every software development activity has a corresponding test activity. However, the same symmetry does not apply to dynamic and static testing. There are some static testing activities (e.g., static analysis) for which there is no obvious corresponding dynamic testing activity	FL-2.1.2

Which of the following is a good testing practice that applies to all software development lifecycles?

a) For each test level, there is a corresponding development level
b) For each test objective, there is a corresponding development objective
c) For every software test activity, there is a corresponding user activity
d) For every software development activity, there is a corresponding test activity

a) Is not correct. Quality control applies to all development activities, meaning that every software development activity has a corresponding test activity. However, here we are attempting to equate test levels with development levels, and, although we know what is meant by 'test levels', there is no common understanding of the term ' development level' b) Is not correct. Every software development activity has a corresponding test activity; however test objectives are quite different. For instance, there might be a test objective of ensuring that a test object adheres to a contractual requirement that a certain type of testing must be performed before delivery. In this case there is no reason for there to be a corresponding development objective c) Is not correct. Quality control applies to all development activities, meaning that every software development activity has a corresponding test activity. However, the same symmetry does not apply to testing and user activities. For instance, for some systems it is difficult to even identify the end users. Also, some test activities are focused on developers (e.g., testing for ease of maintainability), which has no user aspect to it d) Is correct. Quality control applies to all development activities, meaning that every software development activity has a corresponding test activity	FL-2.1.2

Which of the following is an example of a good testing practice?

a. Different test levels should have specific test objectives
b. Testers should have development experience
c. Developers should determine the order of test execution in the test procedures
d. Test design should begin when the code is complete to avoid changes

A is correct. This is a good testing practice.

B is not a requirement for many testers.

C is not correct because this should be determined by the testers based on priority, risk, availability, etc.

D is not correct because test design should start during code design and implementation.

Which of the following is an example of a good testing practice?

a. Testers should have development experience
b. Developers should determine the order of test execution in the test procedures
c. Test design should begin when the code is complete to avoid changes
d. Testers should review requirements documents as soon as a readable draft is available

D is correct. This is a good testing practice.

A is not a requirement for many testers.

B is not correct because this should be determined by the testers based on priority, risk, availability, etc.

C is not correct because test design should start during code design and implementation.

2.1.3 (K1) Recall the examples of test-first approaches to development

Which of the following statements BEST describes the acceptance test-driven development (ATDD) approach?

a) In ATDD, acceptance criteria are typically created based on the given/when/then format
b) In ATDD, test cases are mainly created at component testing and are code-oriented
c) In ATDD, tests are created, based on acceptance criteria to drive the development of the related software
d) in ATDD, tests are based on the desired behavior of the software, which makes it easier for team members to understand them

a) Is not correct. It is more often used in behavior-driven development (BDD) b) Is not correct. It is the description of test-driven development (TDD) c) Is correct. In acceptance test-driven development (ATDD) tests are written from acceptance criteria as part of the design process d) Is not correct. It is used in BDD	FL-2.1.3

Which of the following is an example of a test-first approach to development?

a) Behavior-Driven Development
b) Test Level Driven Development
c) Function-Driven Development
d) Performance-Driven Development

Select ONE option.

a) Is correct. Behavior-Driven Development (BDD) is a well-known example of a test-first approach to development b) Is not correct. Test Level Driven Development is not a correct example of a test-first approach to development c) Is not correct. Function-Driven Development is not a correct example of a test-first approach to development d) Is not correct. Performance-Driven Development is not a correct example of a test-first approach to development	FL-2.1.3

Which of the following is an example of a test-first approach to development?

a) Component Test-Driven Development
b) Integration Test-Driven Development
c) System Test-Driven Development
d) Acceptance Test-Driven Development

Select ONE option.

a) Is not correct. Component Test-Driven Development is not a correct example of a test-first approach to development b) Is not correct. Integration Test-Driven Development is not a correct example of a test-first approach to development c) Is not correct. System Test-Driven Development is not a correct example of a test-first approach to development d) Is correct. Acceptance Test-Driven Development (ATDD) is a well-known example of a test-first approach to development	FL-2.1.3

Which development approach captures the requirements in a simple test case format?

a. TDD
b. BDD
c. ATDD
d. TBD

B is correct. Behavior-driven development uses the given/when/then format to define the test cases. Those are then used as the requirements to develop the code.

When coding is directed by the test cases, what development approach is being used?

a. TDD
b. BDD
c. ATDD
d. TBD

A is correct. This is an example of test-driven development.

2.1.4 (K2) Summarize how DevOps might have an impact on testing

Which of the following is MOST likely to be a challenge encountered when implementing DevOps?

a) Making sure that non-functional quality characteristics are not overlooked
b) Managing continuously changing test environments
c) The need for more manual testers with suitable experience
d) Setting up the test automation as part of the delivery pipeline

Select ONE option.

a) Is not correct. DevOps generally increases the visibility of non-functional quality characteristics, such as performance and reliability b) Is not correct. Automated processes like continuous integration/continuous delivery (CI/CD) used in DevOps facilitate stable test environments c) Is not correct. Automated processes like CI/CD used in DevOps generally reduce the need for manual testing d) Is correct. DevOps implementation can pose several risks and challenges, including the need to define and set up the delivery pipeline, introduce and maintain CI/CD tools, and establish and maintain test automation	FL-2.1.4

Which of the following statements about DevOps is CORRECT?

a) To speed up releases, continuous integration is used to encourage developers to submit code quickly without the need to complete component testing
b) To be able to update and release systems on a more frequent basis, many automated regression tests are required to reduce the danger of regression
c) To treat both developers and operations equally, the testers will allocate more effort to release testing by operations using a shift-right approach
d) To create increased synergy between testers, developers and operations, the testing must become fully automated with no manual testing

Select ONE option.

a) Is not correct. DevOps enhances testing in several ways, such as by providing fast feedback on code quality, automated regression testing that minimizes regression risk, and promoting a shift-left approach with high-quality code submission and component tests. This is largely provided through continuous integration, where the developers submit component (unit) tests with their new code, which must pass for the code to be admitted to the build. Therefore, developers do need to complete component testin b) Is correct. DevOps enhances testing in several ways, such as by providing fast feedback on code quality, automated regression testing that minimizes regression risk, and promoting a shift-left approach with high-quality code submission and component tests c) Is not correct. DevOps enhances testing in several ways, such as by providing fast feedback on code quality, automated regression testing that minimizes regression risk, and promoting a shift-left approach with high-quality code submission and component tests. Testers do not attempt to treat developers and operations equally by spending more time on release testing, although a shift-right approach to testing (testing in production) may well be used d) Is not correct. Automated processes like continuous integration/continuous delivery (CI/CD) in DevOps facilitate stable test environments and reduce the need for manual testing, however, there is a risk of overlooking the importance of manual testing, especially from a user's perspective	FL-2.1.4

Which of the following are advantages of DevOps?

i. Faster product release and faster time to market
ii. Increases the need for repetitive manual testing
iii. Constant availability of executable software
iv. Reduction in the number of regression tests associated with code refactoring
v. Setting up the test automation framework is inexpensive since everything is automated

a) i, ii, iv are advantages
b) iii, v are advantages
c) i, iii are advantages
d) ii, iv, v are advantages

Select ONE option.

i. is true. Faster product release and faster time to market is an advantage of DevOps ii. is false. Typically, we need less effort for manual tests because of the use of test automation iii. is true. Constant availability of executable software is an advantage iv. is false. More regression tests are needed v. is false. Not everything is automated and setting up a test automation framework is expensive Hence c is correct.	FL-2.1.4

2.1.5 (K2) Explain the Shift-Left approach

. In what way is CI/CD an example of the concept of shift-left?

a. It gets the code to production faster
b. It allows the developers to continuously integrate their code
c. It requires continuous testing throughout the pipeline
d. It elevates the testers as the owners of quality

C is correct. CI/CD requires continuous testing, including test automation, to be implemented for the entire pipeline. This starts testing as early as possible and shifts it to the left in the timeline.

A is not correct as this is not a shift-left concept.

B is true of CI/CD implementations but does not shift-left the testing.

D is not correct because in a good CI/CD implementation, everyone owns quality.

Which of the following is NOT an example of the shift left approach?

a) Reviewing the user requirements before they are formally accepted by the stakeholders
b) Writing a component test before the corresponding code is written
c) Executing a performance efficiency test for a component during component testing
d) Writing a test script before setting up the configuration management process

Select ONE option.

a) Is not correct. Early review is an example of the shift left approach b) Is not correct. TDD is an example of the shift left approach c) Is not correct. Early non-functional testing is an example of the shift left approach d) Is correct. Test scripts should be subject to configuration management, so it makes no sense to create the test scripts before this process is set up	FL-2.1.5

Which of the following provides the BEST description of the shift-left approach?

a) When agreed by the developers, manual activities on the left-hand side of the test process are automated to support the principle of 'early testing saves time and money'
b) Where cost-effective, test activities are moved to be performed earlier in the software development lifecycle (SDLC) to reduce the total cost of quality by reducing the number of defects found later in the SDLC
c) When they have spare time available, testers are required to automate tests for regression testing, starting with component tests and component integration tests
d) When available, testers are trained to perform tasks early in the SDLC to allow more test activities to be automated later in the SDLC

Select ONE option.

a) Is not correct. Practices involved in shift-left testing are aimed at implementing more testing activities in the early phases of the development life cycle, portraying the SDLC as moving from left to right. There is no such thing as the left-hand side of the test process b) Is correct. Shift-left emphasizes the importance of starting testing earlier in the software development lifecycle (SDLC). Implementing shift-left testing necessitates additional training, and increased effort and costs during the early stages of the SDLC, nevertheless, overall savings should be higher c) Is not correct. Although automated component tests and component integration tests for regression testing are generally valuable, the creation of these tests is normally the responsibility of the developers, and if a continuous integration/continuous delivery (CI/CD) approach is followed, then these tests will have been submitted with the code. In some situations the tester may automate tests for regression testing, and sometimes even for component tests and component integration tests, however this is not part of a 'shift-left' approach which moves testing earlier in the SDLC d) Is not correct. Training testers to perform tasks early in the SDLC would support a shift-left approach by emphasizing the importance of starting testing earlier in the SDLC. However, automating more test activities to be performed later in the SDLC is not part of a 'shift-left' approach	FL-2.1.5

2.1.6 (K2) Explain how retrospectives can be used as a mechanism for process improvement

Which of the arguments below would you use to convince your manager to organize retrospectives at the end of each release cycle?

a) Retrospectives are very popular these days and clients would appreciate it if we added them to our processes
b) Organizing retrospectives will save the organization money because without them end user representatives do not provide immediate feedback about the product
c) Process weaknesses identified during the retrospective can be analyzed and serve as a to do list for the organization's continuous process improvement program
d) Retrospectives embrace five values including courage and respect, which are crucial to maintain continuous improvement in the organization

Select ONE option.

a) Is not correct. Retrospectives are more useful for identifying improvement opportunities and have little importance for clients b) Is not correct. Business representatives are not giving feedback about the product itself. Therefore, there is no financial gain to the organization c) Is correct. Regularly conducted retrospectives, when appropriate follow up activities occur, are critical to continual improvement of development and testing d) Is not correct. Courage and respect are values of Extreme Programming and are not closely related to retrospectives	FL-2.1.6

Which of the following BEST describes retrospectives?

a) Retrospectives allow team members to identify other team members who did not fully contribute to achieving quality as required by the whole-team approach
b) Retrospectives give testers an opportunity to identify activities that were successful so that these are retained when potential improvements are made in the future
c) Retrospectives are where agile team members are allowed to voice their concerns about management and customers in a blame-free environment
d) Retrospectives give agile team members a forum where they focus on discussing the plan and technical decisions for the next iteration

Select ONE option.

a) Is not correct. The benefits of retrospectives include team bonding and learning from sharing issues, and better collaboration between developers and testers through reviewing and improving working practices. Calling out individuals who a team member may feel did not fully contribute to achieving quality as required by the whole-team approach will not contribute to this team bonding and collaboration b) Is correct. During the retrospective, the group discusses what aspects of the project were successful and should be retained, as well as areas that could be improved, and how to do so c) Is not correct. The benefits of retrospectives are based on increased effectiveness and efficiency through process improvements; they are not an opportunity to let off steam and criticize management and customers. Also, the results are recorded, usually in the test completion report, so anything said in the meeting could be read by other stakeholders d) Is not correct. Retrospectives are meetings that are typically held at the end of an iteration where team members will focus on discussing quality-related issues that have occurred in the current iteration. They are not used for making plans or technical decisions for the next iteration; this would be done in the iteration planning meeting at the start of the next iteration	FL-2.1.6

Which of the following is LEAST likely to occur as a result of a retrospective?

a) The quality of future test objects improves by identifying improvements in development practices
b) Test efficiency improves by speeding up the configuration of test environments through automation
c) End users' understanding of the development and test processes is improved
d) Automated test scripts are enhanced through feedback from developers

Select ONE option.

a) Is not correct. One of the purposes of retrospectives is to identify potential process improvements, which, if put into practice, should result in the quality of future outputs of the development process (test objects) being higher. So, this is likely to occur as a result of a retrospective b) Is not correct. A benefit of retrospectives for testing includes increased test efficiency through process improvements. So, this is likely to occur as a result of a retrospective c) Is correct. Participants at retrospectives typically include testers, developers, architects, product owners, and business analysts, but end users are rarely invited or attend these meetings – and they are also unlikely to receive any reports from these meetings. So, it is very unlikely that they will learn and understand more about the development and test processes through retrospectives d) Is not correct. A benefit of retrospectives for testing includes improved quality of testware (including automated test scripts) through joint reviews with developers. So, this is likely to occur as a result of a retrospective	FL-2.1.6

2.2.1 (K2) Distinguish the different test levels

Which types of failures (1-4) fit which test levels (A-D) BEST?

1. Failures in system behavior as it deviates from the user's business needs
2. Failures in communication between components
3. Failures in logic in a module
4. Failures in not correctly implemented business rules

A. Component testing
B. Component integration testing
C. System testing
D. Acceptance testing

a) 1D, 2B, 3A, 4C
b) 1D, 2B, 3C, 4A
c) 1B, 2A, 3D, 4C
d) 1C, 2B, 3A, 4D

Select ONE option.

The test basis for acceptance testing is the user's business needs (1D). Communication between components is tested during component integration testing (2B). Failures in logic can be found during component testing (3A). Business rules are the test basis for system testing (4C). Hence a is correct.	FL-2.2.1

Which of the following is MOST likely to be performed as part of system testing?

a) Security testing of a credit management system by an independent test team
b) Testing the interface of a currency exchange system with an external banking system
c) Beta testing of a remote learning system by courseware developers
d) Testing interactions between the user interface and database of a human resources system

Select ONE option.

a) Is correct. System testing examines the behavior and capabilities of the complete system and covers non-functional testing of quality characteristics, which includes security testing. This type of testing is often performed by an independent test team based on system specifications b) Is not correct. System integration testing examines the interfaces with other systems and external services c) Is not correct. Beta testing is a type of acceptance testing performed at an external site by roles outside the development organization d) Is not correct. Component integration testing involves testing the (interfaces and) interactions between components of a system, such as the user interface and database	FL-2.2.1

During which level(s) of testing should non-functional tests be executed?

a. Unit and integration only
b. System testing only
c. Integration, system and acceptance only
d. Unit, integration, system and acceptance only

D is correct. Non-functional tests can and should be executed at all levels of testing.

2.2.2 (K2) Distinguish the different test types

You work as a tester in a project on a mobile application for food ordering for one of your clients. The client sent you a list of requirements. One of them, with high priority, says

"The order must be processed in less than 10 seconds in 95% of the cases".

You created a set of test cases in which a number of random orders were made, the processing time measured, and the test results were checked against the requirements.

What test type did you perform?

a) Functional, because the test cases cover the user's business requirement for the system
b) Non-functional, because the measure the system's performance
c) Functional, because the test cases interact with the user interface
d) Structural, because we need to know the internal structure of the program to measure the order processing time

Select ONE option.

a) Is not correct. The fact that the requirement about the system's performance comes directly from the client and that the performance is important from the business point of view (i.e., high priority) does not make these tests functional, because they do not check "what" the system does, but "how" (i.e., how fast the orders are processed) b) Is correct. This is an example of performance testing, a type of non-functional testing c) Is not correct. From the scenario we do not know if interacting with the user interface is a part of the test conditions. But even if we did, the main test objective of these tests is to check the performance, not the usability d) Is not correct. We do not need to know the internal structure of the code to perform the performance testing. One can execute performance efficiency tests without structural knowledge	FL-2.2.2

Which of the following tests is MOST likely to be performed as part of functional testing?

a) The test checks that the sort function puts the elements of the list or array in ascending order
b) The test checks whether the sort function completes sorting within one second of starting
c) The test checks how easily the sort function can be changed from sorting ascending to sorting descending
d) The test checks that the sort function still functions correctly when moved from a 32-bit to a 64-bit architecture

Select ONE option.

a) Is correct. Checking that the sort function puts the elements of the list or array in ascending order is evaluating the functional correctness of the sort function, which is part of functional testing b) Is not correct. Assessing whether the sort function meets its non-functional requirement to complete within one second is part of testing its performance efficiency, which is part of non-functional testing c) Is not correct. Evaluating the ease with which the sort function can be modified from sorting ascending to sorting descending is testing its modifiability, a form of non-functional maintainability testing, which is part of non-functional testing d) Is not correct. Assessing that the sort function still functions correctly when moved from a 32-bit to a 64-bit architecture is testing its adaptability, a form of portability testing, which is part of non-functional testing	FL-2.2.2

Usability testing is an example of which type of testing?

a. Functional
b. Non-functional
c. Structural
d. Change-related

B is correct. Usability is one of the non-functional test types according to ISO 25010.

2.2.3 (K2) Distinguish confirmation testing from regression testing

You are testing a user story with three acceptance criteria: AC1, AC2 and AC3. AC1 is covered by test case TC1, AC2 by TC2, and AC3 by TC3. The test execution history had three test runs on three consecutive versions of the software as follows:

	Execution 1	Execution 2	Execution 3
TC1	(1) Failed	(4) Passed	(7) Passed
TC2	(2) Passed	(5) Failed	(8) Passed
TC3	(3) Failed	(6) Failed	(9) Passed

Tests are repeated once you are informed that all defects found in the test run are corrected and a new version of the software is available.
Which of the above tests are executed as regression tests?

a) Only 4, 7, 8, 9
b) Only 5, 7
c) Only 4, 6, 8, 9
d) Only 5, 6

Select ONE option.

Because TC1 and TC3 failed in Execution 1 (i.e., test (1) and test (3)), test (4) and test (6) are confirmation tests. Because TC2 and TC3 failed in Execution 2 (i.e., tests (5) and (6)), test (8) and test (9) are also confirmation tests. TC2 passed in Execution 1 (i.e., test (2)), so test (5) is a regression test. TC1 passed in the Execution 2 (i.e., test (4)), so test (7) is also a regression test. Hence b is correct.	FL-2.2.3

The navigation system software has been updated due to it suggesting routes that break traffic laws, such as driving the wrong way down one-way streets. Which of the following BEST describes the testing that will be performed?

a) Only confirmation testing
b) Confirmation testing then regression testing
c) Only regression testing
d) Regression testing then confirmation testing

Select ONE option.

a) Is not correct. Confirmation testing to check that the updates have resulted in a correct implementation is necessary, however, it would then be sensible to perform regression testing to ensure that no defects have been introduced or uncovered in unchanged areas of the system b) Is correct. Confirmation testing will check that the updates have resulted in a correct implementation, and then regression testing will be used to ensure that no defects have been introduced or uncovered in unchanged areas of the system c) Is not correct. Regression testing should be used to ensure that no defects have been introduced or uncovered in unchanged areas of the system when the update was made, however it is also necessary to perform confirmation testing that will check that the updates have resulted in a correct implementation d) Is not correct. Confirmation testing will check that the updates have resulted in a correct implementation, and regression testing will be used to ensure that no defects have been introduced or uncovered in unchanged areas of the system. However, when performed (i.e., when an update needs to be tested), confirmation testing precedes regression testing	FL-2.2.3

Which of the following statements is CORRECT?

a) Regression tests increase in number as the project progresses, whereas the number of confirmation tests decreases as the project progresses
b) Regression tests are created and run when the test object is fixed, whereas confirmation tests are run whenever the test object is enhanced
c) Regression testing is concerned with checking that the operational environment remains unchanged, whereas confirmation testing is concerned with testing changes to the test object
d) Regression testing is concerned with adverse effects in unchanged code, whereas confirmation testing is concerned with testing changed code

Select ONE option.

a) Is not correct. Regression tests increase in number as the project progresses, as new regression tests are typically required as changes are made to the system. Similarly, the number of confirmation tests also typically increases as the project progresses as new confirmation tests are needed for each fix made to a system b) Is not correct. It is the other way round. Confirmation tests are created and run when the test object is fixed, and regression tests are (ideally) run whenever the test object is enhanced (changed) c) Is not correct. Confirmation testing verifies that a defect has been fixed correctly and so is concerned with testing changes to the test object. However, regression testing ensures that changes (including changes to the operational environment) do not have negative effects on unchanged software and so does not check that the operational environment remains unchanged d) Is correct. Regression testing ensures that changes do not have negative effects on unchanged software. Confirmation testing verifies that a defect has been fixed – and so is concerned with changed code	FL-2.2.3

2.3.1 (K2) Summarize maintenance testing and its triggers

Your organization's test strategy suggests that once a system is going to be retired, data migration shall be tested. As part of what test type is this testing MOST likely to be performed?

a) Maintenance testing
b) Regression testing
c) Component testing
d) Integration testing

Select ONE option.

a) Is correct. When a system is retired, this can require testing of data migration, which is a form of maintenance testing b) Is not correct. Regression testing verifies whether a fix accidentally affected the behavior of other parts of the code, but now we are talking about data migration to a new system c) Is not correct. Component testing focuses on individual hardware or software components, not on data migration d) Is not correct. Integration testing focuses on interactions between components and/or systems, not on data migration	FL-2.3.1

Which of the following is MOST likely to be a trigger that leads to maintenance testing of a currency exchange system?

a) The developers reported that changing the currency exchange system was difficult and the testers decided to check if this was true
b) The refund option of the currency exchange system was removed as it did not always repay the correct amount to customers
c) The agile team has started developing a user story that adds a new customer loyalty feature to the currency exchange system
d) The language support option of the currency exchange system was used to enable both English and local language currency transactions

a) Is not correct. Assuming that testers could check the ease of changing the currency exchange system then it would be done by maintainability testing rather than maintenance testing, so this is not a trigger for maintenance testing b) Is correct. A system modification (such as a fix or enhancement) is an example of a trigger for maintenance testing. The removal of the refund option of the currency exchange system was a fix that would lead to maintenance testing c) Is not correct. If the agile team has started developing a user story that adds a new customer loyalty feature to the currency exchange system, then this will result in them testing the new feature, and then they would perform regression testing. No maintenance testing is required in this situation d) Is not correct. Reconfiguration of the currency exchange system to support both the local language and English currency transactions is not a system modification, a change to the operational environment, or a system retirement, which are the three triggers for maintenance testing	FL-2.3.1

. When a system is targeted for decommissioning, what type of maintenance testing may be required?

a. Retirement testing
b. Regression testing
c. Data migration testing
d. Patch testing

C is correct, per syllabus. Data migration to another system or data migration to an archival system may be needed.

A is incorrect, there is no such testing type.

B is incorrect because this is more appropriate for current systems, not the system being retired.

D is incorrect because this is of no use for a system being retired.

Conclusion

In this document :

- **We identified the examinable learning objectives.**
- **We presented the probability of questions for each part of chapter 2.**
- **We summarized chapter 2.**
- **We provided section-wise questions and their answers for chapter 2.**

If you need any assistance or have questions, feel free to reach out! 😊

Karim Kalboussi

kalboussikarim3@gmail.com