



ISTQB®

Certified Tester
Foundation Level

CTFL 4.0

Chapter 1

- Summary -
- Questions & Answers -
- Exam Questions distribution -

Swipe for more



Kalboussi Karim



Examinable Learning Objectives :

Level 1 : Remember (K1)

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

Level 2 : Understand (K2)

- The candidate can select the reasons or explanations for statements related to the topic, and can summarize, compare, classify and give examples for the testing concept.
- Action verbs : Classify, compare, contrast, differentiate, distinguish...
- Example : 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.
- Example : Apply test case prioritization.

Chapter 1 Question Distribution in the Exam :

- There is a total of **8 questions** required for Chapter 1 :

K1 = 2 questions

K2 = 6 questions

K3 = 0 questions

- Number of points for this chapter = 8



Question Distribution	K-Level	Number of Questions per LO (group)*	Suggested Points per Question	Probability of Appearance in the exam
Chapter 1				
FL-1.1.1	K1	1	1	0.5
FL-1.2.2	K1	1	1	0.5
FL-1.5.2	K1	1	1	1
FL-1.1.2	K2	1	1	0.33
FL-1.2.1	K2	1	1	0.33
FL-1.2.3	K2	1	1	0.33
FL-1.3.1	K2	1	1	1
FL-1.4.1	K2	3	1	0.6
FL-1.4.2	K2	1	1	0.6
FL-1.4.3	K2	1	1	0.6
FL-1.4.4	K2	1	1	0.6
FL-1.4.5	K2	1	1	0.6
FL-1.5.1	K2	1	1	0.5
FL-1.5.3	K2	1	1	0.5

Summary of Chapter 1

1.1 What is Testing ?

1.1.1 (K1) identify typical test objectives

- Evaluating work products (requirements, user stories, designs and code).
- Triggering (causing) failures and finding (identifying) defects.
- Building confidence in the quality of the test object.
- Reducing the level of risk of inadequate software quality.
- Verifying that documented/specified requirements are met.
- Ensuring required coverage of a test object.
- Verifying that a test object complies with contractual, legal and regulatory requirements.
- Providing information to stakeholders so they can make informed decisions.
- Validating whether the test object is complete and meets user/stakeholders expectations.

1.1.2 (K2) differentiate testing from debugging

- Debugging is the process of finding, analyzing, and removing defects.
- Debugging is concerned with locating defects (locating defects only when we do dynamic testing) and fixing these defects.
- Testing can directly find defects in the test object when we do static testing.
- Testing can trigger failures that are caused by defects.
- Testing does not find the cause of defects !
- Debugging does not fix the cause of defects !
- Debugging does not fix failures !
- Debugging is a non testing activity !

1.2 Why is testing so necessary ?

1.2.1 (K2) Exemplify why testing is necessary

- Testing indirectly contributes to higher quality test objects.
- Testing provides a means of directly evaluating the quality of a test object at various stages of the SDLC.
- Testing provides users with indirect representation on the development project.
- Reviews increase the quality of requirements specifications and lead to fewer changes being needed in derived work products.
- Static testing is used by developers to identify defects in their program code earlier than can be achieved through dynamic testing.
- Testing may also be required to meet contractual or legal requirements, or to comply with regulatory standards.

1.2.2 (K1) Recall the relation between testing and quality assurance

- Quality Control(QC) is a product-oriented corrective approach and Quality Assurance(QA) is a process-oriented preventive approach.
- Testing is a form of quality control (QC).
- Test results are used by both (QA) and (QC) teams, used by (QA) teams to provide feedback on how well the development and test processes are performing, and used by (QC) teams to fix defects.

1.2.3 (K2) Distinguish between root cause, error, defect, and failure

- Human beings make errors (mistakes), which produce defects (faults/bugs), which in turn may result in failures.
- If a defect in a code is executed, the system may fail to do what it should do, or do something it shouldn't, causing a failure.
- Failures can be caused by errors, defects, or environmental conditions.
- A root cause is a fundamental reason for the occurrence of a problem, a root cause is a situation that leads to an error.
- Root causes are identified through root cause analysis.

1.3 Testing principles

1.3.1 (K2) Explain the seven testing principles

- **Testing shows the presence, not the absence of defects** : If no defects are found, testing cannot prove test object correctness.
- **Exhaustive testing is impossible** : Rather than attempting to test exhaustively, test techniques, test cases prioritization, and risk-based testing should be used to focus test efforts. (Exhaustive testing is not feasible except in trivial cases).
- **Early testing saves time and money** : To find defects early, both static testing and dynamic testing should be started as early as possible.
- **Defects cluster together** : A small number of system components usually contain most of the defects discovered, responsible for most of the operational failures. (Predicted and actual defect clusters are an important input for risk-based testing).
- **Tests wear out** : Repeating same tests becomes ineffective in detecting new defects. Existing tests and test data may need to be modified.
- **Testing is context dependent** : Testing is done differently in different contexts.
- **Absence-of-defects fallacy** : It is a fallacy to expect that software verification will ensure the success of a system, in addition to verification, validation should also be carried out.

1.4 Test activities, Testware and test Roles

1.4.1 (K2) Summarize the different test activities and tasks

Test activities may appear to follow a logical sequence, but they are implemented often iteratively or in parallel.

- **Test planning** : Defining test objectives and selecting an approach that best achieves the objectives.
- **Test monitoring & Control** : Monitoring is the ongoing check of all test activities and comparaison of actual test progress against the plan. Control is about taking actions necessary to meet the objective of testing.
- **Test analysis (What to test?)** : Analyzing the test basis to identify testable features, to define and prioritize associated test condidtion, together with the related risks and risk levels. test basis and test objects are also evaluated to identify defects they may contain. Test analysis is often supported by the use of test techniques.
- **Test Design (How to test?)** : Elaborating test conditions into test cases and other testware, identification of coverage items, which will help specify test cases inputs. Defining the test data requirements, designing the test environment and identifying any other required infrastructure and tools. Test techniques can be used to support this activity.
- **Test implementation** : Includes creating or acquiring the testware necessary for test execution (test data, ...). Test cases can be organized into test procedures and are often assembled into test suites.
- **Test execution** : Running test in accordance with the test execution schedule. Actual test results are compared with the expected results.
- **Test completion** : Change requests or product backlog items created for any unresolved defects. Test environment is shutdown to an agreed state. Any testware that may be useful in the future is identified. A test completion report is created and communicated to the stakeholders. Test activities are analyzed to identify lessons learned and improvements for future iterations, releases, or projects.

1.4.2 (K2) Explain the impact of context on the test process

The way testing is carried out will depend on a number of contextual factors like the SDLC, team members (skills, knowledge, level of experience...), identified product risks, market needs, specific regulations, criticality of the test object, technical factors (type of software, technology used...), project constraints (scope, time, budget...), organizational factors (structure, existing policies, practices...), tools (availability, usability, compliance...) and also stakeholders (needs, expectations, willingness to cooperate...)

1.4.3 (K2) Differentiate the testware that supports the test activities

Testware is created as output work products from the test activities :

- **Test planning** : Test plan that contains test schedule, risk register and entry and exit criteria. Risk register is a list of risks together with the risk likelihood, risk impact and risk mitigation.
- **Test monitoring & Control** : Test progress reports, documentation of control directives and risk information.
- **Test analysis** : Prioritized test conditions and defect reports (regarding defects in the test basis).
- **Test Design** : Prioritized test cases, test charters, coverage items, test data requirements and test environment requirements (elements needed to build it).
- **Test implementation** : Test procedures, automated test scripts, test suites, test data, test execution schedule, test environment items (drivers, simulators...)
- **Test execution** : Test logs and defect reports.
- **Test completion** : Test completion report, action items for improvements, lessons learned and change requests.

1.4.4 (K2) Explain the value of maintaining traceability

It is important to establish traceability throughout the test process between the test basis elements, testware associated with these elements, test results and detected defects. Traceability of test cases to requirements is performed to verify that the requirements are covered by test cases, traceability of test results to risks to evaluate the level of residual risk in a test object, traceability between user requirements and test execution results provides a means of measuring project progress against business goals. Good traceability makes it possible to determine the impact of changes, and also make test progress and completion reports more easily understandable.

1.4.5 (K2) Compare the different roles in testing

- **Test management role** : Takes overall responsibility for the test process, test team and leadership of the test activities. This role is focused on the activities of test planning, test monitoring & control, and test completion.
- **Testing role** : Takes overall responsibility for the engineering (technical) aspect of testing (evaluating test basis and test object, defining test environment requirements, assessing the testability of a test object). The testing role focuses on the activities of test analysis, test design, test implementation, and test execution.

1.5 Essential skills and good practices in Testing

1.5.1 (K2) Give examples of the generic skills required for testing

Testing knowledge, thoroughness, carefulness, curiosity, attention to details, good communication skills, being a team player, analytical thinking, critical thinking, creativity, technical knowledge, and domain knowledge.

1.5.2 (K1) Recall the advantages of the whole team approach

- Any team member with the necessary knowledge and skills can perform any task, and everyone is responsible for quality. The whole team approach (practice coming from Extreme programming) improves team dynamics, enhances communication and collaboration and creates synergy.
- Testers help business representatives to create suitable acceptance tests.
- Testers collaborate with developers to define and agree on the test strategy.

1.5.3 (K2) Distinguish the benefits and drawbacks of independence of testing

- The independence of testing makes the tester more effective at finding defects due to differences between the author's and the tester's cognitive biases.
- Independent testers are likely to recognize different kinds of failures and defects compared to developers because of their different backgrounds, technical perspectives and biases. (developers are not objective about their code).
- Independent testers can verify, challenge, or disprove assumptions made by stakeholders.
- Independent testers may be isolated from the development team, which may lead to a lack of collaboration, communication problems, or an adversarial relationship with the development team.
- The independence of testing level are :
 - **No independence** : Work products are tested by their author.
 - **Some independence** : Work products are evaluated by the author's peer from the same team.
 - **High independence** : Work products are evaluated by testers outside the author's team but within the same organization.
 - **Very high independence** : Work products are evaluated by testers outside the organization.

Swipe for the questions part



Questions from Chapter 1 in the ISTQB exam

1.1.1 (K1) identify typical test objectives

When the tester verifies the test basis while designing tests early in the lifecycle, which test objective is being achieved?

- a. Gaining confidence
- b. Finding defects
- c. Evaluating work products
- d. Providing information for decision making

C is correct per the syllabus.

The other three are achieved primarily by doing dynamic testing. This is a bit tricky because you are very likely to find defects while doing this analysis and this may lead to either gaining or destroying confidence and needing to supply information to the decision makers. What is occurring here is an evaluation of the work product, in this case the requirements.

Which of the following is a typical test objective?

- a) Validating that documented requirements are met
- b) Causing failures and identifying defects
- c) Initiating errors and identifying root causes
- d) Verifying the test object meets user expectations

Select ONE option.

<p>a) Is not correct. Validating that documented requirements are met is incorrect as validation is concerned with meeting user requirements and expectations, while verification is concerned with meeting specified requirements, so this would be correct if we replaced 'validating' with 'verifying'</p> <p>b) Is correct. Causing failures and identifying defects is probably the most common objective of dynamic testing</p> <p>c) Is not correct. Initiating errors and identifying root causes is incorrect because testers do not initiate errors, they try to cause failures. Errors are typically made by developers (and cannot really be initiated) and result in defects, which testers attempt to identify either directly through static testing or indirectly through failures with dynamic testing. Identifying root causes is useful but is part of debugging, which is a separate activity to testing</p> <p>d) Is not correct. Verifying the test object meets user expectations is incorrect as <u>verification</u> is concerned with checking specified (documented) requirements are met, while <u>validation</u> is concerned with meeting user requirements and expectations, so this would be correct if we replaced 'verifying' with 'validating'</p>	FL-1.1.1
--	----------

Which of the following is a typical test objective?

- a) Finding and fixing defects in the test object
- b) Maintaining effective communications with developers
- c) Validating that legal requirements have been met
- d) Building confidence in the quality of the test object

Select ONE option.

<ul style="list-style-type: none">a) Is not correct. Finding and fixing defects in the test object is not a typical test objective as although identifying defects is an objective of testing, fixing defects is not a testing activityb) Is not correct. Maintaining effective communications with developers is not a typical test objective as although it is useful in achieving other objectives of testing, such as providing stakeholders with information that enables them to make informed decisions, it is not a primary reason for performing testingc) Is not correct. Validating that legal requirements have been met is not a typical test objective because validation is concerned with checking whether the system meets users' and other stakeholders' needs in its operational environment. Checking that legal requirements have been met is a form of verificationd) Is correct. Building confidence in the quality of the test object is achieved by executing tests that pass	FL-1.1.1
---	----------

1.1.2 (K2) differentiate testing from debugging

You were given a task to analyze and fix causes of failures in a new system to be released.

Which activity are you performing?

- a) Debugging
- b) Software testing
- c) Requirement elicitation
- d) Defect management

Select ONE option.

<ul style="list-style-type: none">a) Is correct. Debugging is the process of finding, analyzing, and removing the causes of failures in a component or systemb) Is not correct. Testing is the process concerned with planning, preparation and evaluation of a component or system and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects. It is not related to fixing causes of failuresc) Is not correct. Requirement elicitation is the process of gathering, capturing, and consolidating requirements from available sources. It is not related to fixing causes of failuresd) Is not correct. Defect management is the process of recognizing, recording, classifying, investigating, resolving, and disposing of defects. It is not related to fixing causes of failures	FL-1.1.2
--	----------

Which of the following statements BEST describes the difference between testing and debugging?

- a) Testing causes failures while debugging fixes failures
- b) Testing is a negative activity while debugging is a positive activity
- c) Testing determines that defects exist while debugging removes defects
- d) Testing finds the cause of defects while debugging fixes the cause of defects

Select ONE option.

<p>a) Is not correct. Dynamic testing does cause failures (from which defects can then be located and fixed). However, <u>debugging is concerned with locating defects and fixing these defects</u>. Therefore, <u>debugging does not fix failures</u></p> <p>b) Is not correct. Both testing and debugging contribute to improving the quality of the test object, so should really both be considered positively. Debugging is generally considered to be a positive activity as it is fixing something. Dynamic testing does involve intentionally causing the test object to fail, which is why some people consider it a negative activity, but that is a very narrow view (and not one typically held by testers). Both positive and negative test cases are possible. Positive test cases check that the test object correctly performs what it is supposed to do, while negative testing checks that the test object does not do what it is not supposed to do</p> <p>c) Is correct. <u>Testing determines that defects exist either directly through observation of the defect in reviews (or by a tool in static analysis), or indirectly by causing a failure in dynamic testing</u>. Debugging is a <u>separate activity from testing (normally performed by developers)</u> and is <u>concerned with locating defects (only for dynamic testing) and fixing the defects</u></p> <p>d) Is not correct. The causes of defects are typically human errors. Testing finds defects either directly through static testing, or indirectly by causing failures in dynamic testing, and debugging fixes defects. So, testing does not find the cause of defects and debugging does not fix the causes of defects</p>	FL-1.1.2
---	----------

1.2.1 (K2) Exemplify why testing is necessary

Which of the following options shows an example of test activities that contribute to success?

- a) Having testers involved during various software development lifecycle (SDLC) activities will help to detect defects in work products
- b) Testers try not to disturb the developers while coding, so that the developers write better code
- c) Testers collaborating with end users help to improve the quality of defect reports during component integration and system testing
- d) Certified testers will design much better test cases than non-certified testers

<p>a) Is correct. It is important that testers are involved from the beginning of the software development lifecycle (SDLC). It will increase understanding of design decisions and will detect defects early.</p> <p>b) Is not correct. Both developers and testers will have more understanding of each other's work products and how to test the code</p> <p>c) Is not correct. If testers can work closely with system designers, it will give them insight as to how to test</p> <p>d) Is not correct. Testing will not be successful if legal requirements are not tested for compliance</p>	FL-1.2.1
--	----------

Which of the following is an example of why testing is necessary?

- a) Dynamic testing increases quality by causing test objects to fail in ways that could never be achieved by the users
- b) Static testing is used by developers to identify failures in their program code earlier than can be achieved through dynamic testing
- c) Static analysis provides evidence to customers that the elements of the system that provide no outputs are fit for release
- d) Reviews increase the quality of requirements specifications and lead to fewer changes being needed in derived work products

Select ONE option.

<p>a) Is not correct. It is often possible to use dynamic testing to cause a test object to fail in ways that could never be achieved by the users, such as by using fault injection. However, if the failure can never occur with real end users, then identifying it is not especially valuable as testing is ultimately aimed at improving the work product for the end users. Spending time testing for failures that cannot occur with real users is not an efficient use of a tester's time</p> <p>b) Is not correct. Static testing in the form of static analysis is used by developers to identify defects in their program code earlier than can be achieved through dynamic testing. Note, however, that static testing (and static analysis) is used to detect defects, not failures, which are found by dynamic testing. Thus it is the use of the term 'failures' that makes this an incorrect option</p> <p>c) Is not correct. Static analysis directly detects anomalies in code, which may be defects, and this is normally for the developer, not the customer. The provision of evidence for release by the use of static analysis of elements that provide no output is nonsense</p> <p>d) Is correct. Reviews are a form of static testing that can be applied from the very start of the software development lifecycle and are used to find defects that can be removed before subsequent development activities waste effort on faulty requirements. If the defects are not detected and removed early on, then when the defect is found derived work products, such as the design and code, will need to be changed as they were based on faulty requirements</p>	FL-1.2.1
--	----------

1.2.2 (K1) Recall the relation between testing and quality assurance

In many software organizations the test department is called the Quality Assurance (QA) department. Is this sentence correct or not and why?

- a) It is correct. Testing and QA mean exactly the same thing
- b) It is correct. These names can be used interchangeably because both testing and QA focus their activities on the same quality issues
- c) It is not correct. Testing is something more; testing includes all activities with regard to quality. QA focuses on quality-related processes
- d) It is not correct. QA is focused on quality-related processes while testing concentrates on demonstrating that a component or system is fit for purpose and to detect defects

Select ONE option.

<ul style="list-style-type: none">a) It is not correct. See justification db) It is not correct. See justification dc) It is not correct. See justification dd) Is correct. Testing and quality assurance are not the same. Testing is the process consisting of all software development lifecycle (SDLC) activities, both static and dynamic, concerned with planning, preparation and evaluation of a component or system and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects. Quality assurance is focused on establishing, introducing, monitoring, improving, and adhering to the quality-related processes	FL-1.2.2
--	----------

1.2.3 (K2) Distinguish between root cause, error, defect, and failure

A designer documents a design for a user interface that does not suitably address disabled users because the designer is tired. The programmer implements the user interface in line with the design but as they are working under severe time pressure, they do not include suitable exception handling in their program code for bonus calculations. When the operational system is used, complaints are made by some disabled users about the interface and the company is subsequently fined by the relevant regulatory authority. No one notices that bonus calculations are sometimes incorrect.

Which of the following statements is CORRECT?

- a) The miscalculation of bonuses is a defect that occasionally occurs
- b) The fine received for failing to address some disabled users is a failure
- c) The programmer working under severe time pressure is a root cause
- d) The design of the user interface includes a designer error

Select ONE option.

<p>a) Is not correct. The miscalculation of bonuses is a failure by the system, not a defect</p> <p>b) Is not correct. The system not suitably supporting disabled users is a failure which eventually results in a fine, but the fine itself is not a failure (it appears to be the correct functioning of the regulatory system)</p> <p>c) Is correct. The error is made by the programmer and this mistake is caused by them working under severe time pressure, which is the root cause of the subsequent defect</p> <p>d) Is not correct. The poor design of the user interface, which does not suitably address disabled users, is a design defect caused by the designer error. Thus the design of the user interface includes a design defect not a designer error</p>	FL-1.2.3
--	----------

Which of the following is a correct statement?

- a. A developer makes a mistake which causes a defect that may be seen as a failure during dynamic testing
- b. A developer makes an error which results in a failure that may be seen as a fault when the software is executed
- c. A developer has introduced a failure which results in a defect that may be seen as a mistake during dynamic testing
- d. A developer makes a mistake which causes a bug that may be seen as a defect when the software is executed

A is correct. The developer makes a mistake/error which causes a defect/fault/bug which may cause a failure when the code is dynamically tested or executed.

B is incorrect because fault and failure are reversed.

C is incorrect because failure and mistake are reversed.

D is incorrect because it's a failure that's seen during execution, not the defect itself. The failure is a symptom of the defect.

1.3.1 (K2) Explain the seven testing principles

One of the 'principles of testing' states that exhaustive testing is impossible. Which of the following is an example of addressing this principle in practice?

- a) Creating test cases that cover every possible specified output
- b) Documenting all possible test input variations and prioritizing these based on importance
- c) Starting testing as early as possible with reviews and other static testing approaches
- d) Using equivalence partitioning and boundary value analysis to generate test cases

Select ONE option.

The 'exhaustive testing is impossible' principle is concerned with the fact that it is not feasible to test every possible variation of test inputs in all different circumstances, except in trivial cases. Instead, testing utilizes test techniques, test case prioritization, and risk-based testing to sample from the set of possibilities and focus testing efforts.

FL-1.3.1

- a) Is not correct. The principle states that it is not feasible to test everything except in trivial cases. Testing everything would require testing every possible variation of test inputs in all different circumstances, which is generally infeasible as there will be a practically infinite number. Testing every possible specified output will not address this problem as the relationship between inputs and specified outputs can be different for each test object. Sometimes there may be a practically infinite number of possible specified outputs (e.g., when there are several variables representing real numbers), whereas at other times there may be just two specified outputs, such as with a single variable that can be either true or false
- b) Is not correct. The principle states that it is not feasible to test every possible variation of test inputs in all different circumstances. This is because for non-trivial systems there is a practically infinite number. Therefore, in practice, documenting all possible test input variations would be impractical as it would take an infinite length of time
- c) Is not correct. Starting testing as early as possible with reviews and other static testing approaches will not address the problem of there being too many possible test cases. The 'early testing saves time and money' principle is concerned with fixing defects early on to prevent the occurrence of subsequent defects in derived work products, thereby reducing costs and the likelihood of failures
- d) Is correct. The use of equivalence partitioning and boundary value analysis to generate test cases is one way to address the principle as these test techniques provide a systematic way to derive a finite subset of all possible test cases

The 'absence-of-defects fallacy' is one of the principles of testing. Which of the following is an example of addressing this principle in practice?

- a) Explaining that it is not possible for testing to show the absence of defects
- b) Supporting the end users to perform acceptance testing
- c) Ensuring that no implementation defects remain in the delivered system
- d) Modifying tests that cause no failures to ensure few defects remain

Select ONE option.

The ‘absence-of-defects fallacy’ is concerned with the idea that ensuring correctness in accordance with the requirements (i.e., verifying the absence of implementation defects) does not guarantee user satisfaction with the system. To address this it is also necessary to validate that the system meets users' needs and expectations, fulfills business objectives, and outperforms competing systems.

FL-1.3.1

- a) Is not correct. The ‘testing shows the presence, not the absence of defects’ principle explains that while testing can detect the existence of defects in the test object, it is not possible to demonstrate that there are no defects and, therefore, guarantee its correctness. Therefore, explaining that it is not possible for testing to show the absence of defects would partially address this principle, not the ‘absence-of-defects’ fallacy
- b) Is correct. By supporting the end user to perform acceptance testing it should be possible to validate that the system meets users' needs and expectations
- c) Is not correct. It is not possible to ensure that no implementation defects remain in the delivered system as the ‘testing shows the presence, not the absence of defects’ principle explains that while testing can detect the existence of defects in the test object, it is not possible to demonstrate that there are no defects and, therefore, guarantee its correctness
- d) Is not correct. Modifying tests that cause no failures to ensure few defects remain is one way to address the ‘tests wear out’ principle. This principle is concerned with the idea that repeating identical tests on unaltered code is unlikely to uncover novel defects and therefore, modifying tests may be essential. This will not validate that the system meets users' needs and expectations

Why is it important to avoid the principle of tests wearing out?

- a. Dynamic testing is less reliable in finding bugs
- b. Running the same tests repeatedly will consistently find new failures
- c. Tests should not be context dependent
- d. Running the same tests repeatedly will reduce the chance of finding new failures

D is correct. As tests are run repeatedly, the tests become less effective.

A is not correct because dynamic testing should be used and helps to alleviate tests wearing out.

B is not correct. The same tests do wear out.

C is not correct because testing should be context dependent.

1.4.1 (K2) Summarize the different test activities and tasks

When following a standard test process, when should the test control activity take place?

- a. During the planning activities
- b. During the implementation and execution activities
- c. During the monitoring activities
- d. During all the activities

D is correct. Control occurs throughout the project to ensure that it is staying on track based on the plan and to take any corrective steps that may be necessary. The monitoring information is used to determine if control actions are needed.

Which of the following is the activity that compares the planned test progress to the actual test progress?

- a. Test monitoring
- b. Test planning
- c. Test closure
- d. Test control

A is correct. Test monitoring involves the on-going comparison of actual progress against the test plan.

B is incorrect because it defines testing objectives.

C is incorrect because the activities have already completed and the project is closing down.

D is incorrect because test control is when you take actions to correct any issues observed during monitoring.

You work in a team that develops a mobile application for food ordering. In the current iteration the team decided to implement the payment functionality.

Which of the following activities is a part of test analysis?

- a) Estimating that testing the integration with the payment service will take 8 person-days
- b) Deciding that the team should test if it is possible to properly share payment between many users
- c) Using boundary value analysis (BVA) to derive the test data for the test cases that check the correct payment processing for the minimum allowed amount to be paid
- d) Analyzing the discrepancy between the actual result and expected result after executing a test case that checks the process of payment with a credit card, and reporting a defect

- | | |
|--|----------|
| <p>a) Is not correct. Estimating the test effort is part of test planning</p> <p>b) Is correct. This is an example of defining test conditions which is a part of test analysis</p> <p>c) Is not correct. Using test techniques to derive coverage items is a part of test design</p> <p>d) Is not correct. Reporting defects found during dynamic testing is a part of test execution</p> | FL-1.4.1 |
|--|----------|
-

Given the following test tasks:

1. Derive test cases from test conditions
2. Identify reusable testware
3. Organize test cases into test procedures
4. Evaluate test basis and test object

And the following test activities:

- A. Test analysis
- B. Test design
- C. Test implementation
- D. Test completion

Which of the following BEST matches the tasks with the activities?

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

<p>Considering each of the listed test activities and their tasks:</p>	FL-1.4.1
--	----------

- A. Test analysis - To identify the features that require testing, the test basis is analyzed and defined as test conditions, which are then prioritized along with related risks. During this test analysis, defects in the test basis are typically uncovered, and the test object's testability may also be assessed. (Task 4)
- B. Test design - Involves using test conditions to create test cases and other necessary testware, such as test data requirements and test charters for exploratory testing. (Task 1)
- C. Test implementation - Test procedures, such as manual and automated test scripts, are created from test cases and may be assembled into test suites. Test procedures are prioritized and arranged in a test execution schedule. (Task 3)
- D. Test completion - Occurs at project milestones, such as release, end of iteration or end of test level. Testware is identified and archived or handed to the appropriate teams for reuse, the test environment is shut down, and the test activities are analyzed for lessons learned and future improvements. (Task 2)

Thus:

- a) Is not correct
- b) Is correct. The CORRECT match is: 1B, 2D, 3C, 4A
- c) Is not correct
- d) Is not correct

Which of the following test activities are MOST likely to involve the application of boundary value analysis and equivalence partitioning?

- a) Test implementation
- b) Test design
- c) Test execution
- d) Test monitoring
- e) Test analysis

Select TWO options.

Given the following description of test analysis:

To identify the features that require testing, the test basis is analyzed and defined as test conditions, which are then prioritized along with related risks. The systematic identification of test conditions as coverage items often involves using test techniques both during test analysis and as part of the test design activity.

From the above description, it can be seen that test techniques are often used in the test analysis and test design activities. Boundary value analysis and equivalence partitioning are test techniques.

- a) Is not correct. Test implementation is not likely to involve the use of test techniques as it is mostly concerned with assembling test cases into test procedures, while test techniques create test cases
- b) Is correct. Test design is likely to involve the use of test techniques to create test cases from test conditions and coverage items
- c) Is not correct. Test execution is not likely to involve the use of test techniques as it is mostly concerned with executing test procedures (and so test cases), while test techniques create test cases
- d) Is not correct. Test monitoring is not likely to involve the use of test techniques. Test monitoring is mostly concerned with ongoing checks to ensure the plan is being followed, while test techniques create test cases
- e) Is correct. Test analysis is likely to involve the use of test techniques to identify test conditions

FL-1.4.1

Which test activity involves working with test data requirements, test conditions, test environment requirements and test cases?

- a) Test design
- b) Test execution
- c) Test analysis
- d) Test implementation

Select ONE option.

<p>a) Is correct. Test design involves using test conditions to create test cases and other necessary testware, such as test data requirements and test charters for exploratory testing. Test environment requirements are also specified, including the necessary infrastructure and tools</p> <p>b) Is not correct. Test execution involves executing test cases (as part of test procedures), however it does not directly cover the other testware mentioned in the question, such as test data requirements, test environment requirements and test conditions</p> <p>c) Is not correct. Test analysis is used to identify the features that require testing. The test basis is analyzed and defined as test conditions, which are then prioritized along with related risks. While this activity involves working with test conditions, it does not cover the other testware mentioned in the question, such as test data requirements, test environment requirements and test cases</p> <p>d) Is not correct. Test implementation includes the generation of test procedures, such as manual and automated test scripts, which are created from test cases and may be assembled into test suites. Test procedures are prioritized and arranged in a test execution schedule. Test data is created, and the test environment built, and its set up verified. While this activity involves explicitly working with test cases, and may use test data requirements and test environment requirements to create test data and the test environment, it does not cover test conditions</p>	FL-1.4.1
--	----------

1.4.2 (K2) Explain the impact of context on the test process

Which of the following factors have a SIGNIFICANT influence on the test approach?

- i. The SDLC
- ii. The number of defects detected in previous projects
- iii. The identified product risks
- iv. New regulatory requirements forcing formal white-box testing
- v. The test environment setup

- a) i, ii have significant influence
- b) i, iii, iv have significant influence
- c) ii, iv, v have significant influence
- d) iii, v have significant influence

<p>i. Is true. The SDLC has an influence on the test process</p> <p>ii. Is false. The number of defects detected in previous projects may have some influence, but this is not as significant as i, iii and iv</p> <p>iii. Is true. The identified product risks are one of the most important factors influencing the test process</p> <p>iv. Is true. Regulatory requirements are important factors influencing the test process</p> <p>v. Is false. The test environment should be a copy of the production environment but has no significant influence on the test process</p> <p>Hence b is correct.</p>	FL-1.4.2
--	----------

If you are working on a project that is constrained by time and budget, which is pressuring testing to be done quickly. How should the test approach be adjusted?

- a. Develop detailed test cases to reduce the test automation effort
- b. Use techniques such as exploratory and checklist testing to spend less time on test case development
- c. Ensure that your testing starts only after the developers have completed integration tests
- d. Develop end-to-end test automation before performing manual testing so the automation can be used sooner

B is correct. These techniques will help adjust to the project context where there is pressure to test quickly. These are lightweight approaches that require less time preparing documentation and allow test execution to start sooner.

A is not correct because developing detailed test cases will take significant time.

C is not correct because starting testing earlier would be helpful, not later.

D is not correct because this will be problematic automation and the automators will be slowed down by the failures that would have been found in manual testing.

1.4.3 (K2) Differentiate the testware that supports the test activities

Consider the following testware.

Test Charter #04.018	Session time: 1h
Explore:	Registration page
With:	Different sets of incorrect input data
To discover:	Defects related to accepting the registration process with the incorrect input

Which test activity produces this testware as an output?

- a) Test planning
- b) Test monitoring and control
- c) Test analysis
- d) Test design

Select ONE option.

The testware under consideration is a test charter

Test charters are the output from test design

Hence d is correct.

FL-1.4.3

Given the following testware:

- i. Test completion report
- ii. Data held in a database used for test inputs and expected results
- iii. The list of elements needed to build the test environment
- iv. Documented sequences of test cases in execution order
- v. Test cases

Which of the following BEST shows the testware produced as a result of performing test implementation?

- a) ii, iv
- b) iii, v
- c) i, ii, v
- d) i, iii, iv

Select ONE option.

Considering each of the listed testware, and the test activity that produces it:

FL-1.4.3

- i. The test completion report is an output of the test completion activity
- ii. Data held in a database used for test inputs and expected results is the test data - output of test implementation activity
- iii. The list of elements needed to build the test environment is the test environment requirements - output of test design activity.
- iv. Documented sequences of test cases in execution order are the test procedures - output of test implementation activity
- v. Test cases - output of test design activity

Test implementation produces the following outputs: test procedures (iv), automated test scripts, test suites, test data (ii), test execution schedule, and test environment elements such as stubs, drivers, simulators, and service virtualizations.

Thus:

- a) Is correct. Items ii and iv in the list are produced as a result of test implementation
- b) Is not correct
- c) Is not correct
- d) Is not correct

Given the following testware:

1. Coverage items
2. Change requests
3. Test execution schedule
4. Prioritized test conditions

And the following test activities

- A. Test analysis
- B. Test design
- C. Test implementation
- D. Test completion

Which of the following BEST shows the testware produced by the activities?

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

Select ONE option.

Considering each of the listed test activities and their output testware:

- A. **Test analysis - prioritized test conditions (4)** (e.g., acceptance criteria), and defect reports for defects identified in the test basis
- B. **Test design - prioritized test cases, test charters, coverage items (1)**, test data requirements, and test environment requirements
- C. **Test implementation - test procedures, automated test scripts, test suites, test data, test execution schedule (3)**, and test environment elements such as stubs, drivers, simulators, and service virtualizations
- D. **Test completion - test completion report, documented lessons learned, action items for improvement, and change requests (2)** (as product backlog items)

FL-1.4.3

Thus:

- a) Is correct. The correct match is: 1B, 2D, 3C, 4A
- b) Is not correct
- c) Is not correct
- d) Is not correct

Your team has conducted a quality risk analysis and has determined the likelihood, impact and mitigation plan for each identified risk. This information should be captured in what work product?

- a. Test strategy
- b. Test plan
- c. Risk register
- d. Risk plan

C is correct, per the syllabus. This information should be recorded in the risk register. The risk register should be referenced in the test plan for the project.

1.4.4 (K2) Explain the value of maintaining traceability

Which of the following is the BEST example of how traceability supports testing?

- a) Performing the impact analysis of a change will give information about the completion of the tests
- b) Analyzing the traceability between test cases and test results will give information about the estimated level of residual risk
- c) Performing the impact analysis of a change will help selecting the right test cases for regression testing
- d) Analyzing the traceability between the test basis, the test objects and the test cases will help in selecting test data to achieve the assumed coverage of the test object

Select ONE option.

a) Is not correct. <u>Performing the impact analysis will not give information about completeness of tests. Analyzing the impact analysis of changes will help to select the right test cases for execution</u>	FL-1.4.4
b) Is not correct. <u>Traceability does not give information about the estimated level of residual risk if the test cases are not traced back to risks</u>	
c) Is correct. <u>Performing the impact analysis of the changes helps in selecting the test cases for the regression test</u>	
d) Is not correct. <u>Analyzing the traceability between the test basis, test objects and test cases does not help in selecting test data to achieve the assumed coverage of the test object</u> . Selecting test data is more related to test analysis and test implementation, not traceability	

Which of the following statements is a CORRECT example of the value of traceability?

- a) Traceability between the mitigated risks and passing test cases provides a means of determining the level of residual risk
- b) Traceability between user requirements and test execution results provides a means of measuring project progress against business goals
- c) Traceability between testers and failing test cases provides a means of determining the skill level of the testers
- d) Traceability between the identified risks and written test conditions provides a means of determining which risks are worth testing

<ul style="list-style-type: none">a) Is not correct. Traceability between the mitigated risks and passing test cases provides little information, because to be mitigated (by testing) the risks would need to have a corresponding passing test case. To be able to assess residual risk, traceability between all risks and test results needs to be available, so that the risks that do not have a corresponding passing test can be identified as the residual risksb) Is correct. Traceability between user requirements and test execution results provides an indication of which user requirements have been tested and so provides a means of measuring project progress (in the context of testing) against business goalsc) Is not correct. It is not clear that failing test cases provide an indication of tester skills any more than passing test cases. It would partly depend on the test objective (e.g., building confidence or causing failures). Also, such measurement of testers based on passing and failing test cases can be counter-productive as it could cause the testers to optimize their testing based on that metric rather than the test objectived) Is not correct. Traceability between the identified risks and written test conditions provides a means of determining which further test conditions need to be written. Determining which risks are worth testing is part of risk management, and risk mitigation in particular	FL-1.4.4
---	----------

If you need to provide a report showing test case execution coverage of the requirements, what do you need to track?

- a. Traceability between the test cases and the requirements
- b. Coverage of the risk items by test case
- c. Traceability between the requirements and the risk items
- d. Coverage of the requirements by the test cases that have been designed

A is correct. In order to show the test execution coverage of the requirements you will need traceability between the requirements and the test cases. As the test cases are executed this traceability can be used to record tests executed against the requirements.

B is not correct because it's looking for requirements coverage, not risk coverage.

C is not correct because it's looking for test execution, not risk items.

D is not correct because it's looking for test cases that have been executed, not just designed.

1.4.5 (K2) Compare the different roles in testing

Which TWO of the following tasks belong MAINLY to a testing role?

- a) Configure test environments
- b) Maintain the product backlog
- c) Design solutions to new requirements
- d) Create the test plan
- e) Analyze the test basis

Select TWO options.

- | | |
|--|----------|
| <ul style="list-style-type: none">a) Is correct. This is done by the testersb) Is not correct. The <u>product backlog</u> is built and maintained by the product ownerc) Is not correct. This is done by the development teamd) Is not correct. This is a managerial rolee) Is correct. This is done by <u>the testers</u> | FL-1.4.5 |
|--|----------|

Which of the following is MOST likely to describe a task performed by someone in a test management role?

- a) Evaluate test basis and test object
- b) Define test environment requirements
- c) Assess testability of test object
- d) Create test completion report

Select ONE option.

- | | |
|---|----------|
| <ul style="list-style-type: none">a) Is not correct. The testing role is primarily responsible for the technical and engineering aspects of testing, such as test analysis, test design, test implementation, and test execution. Evaluating the test basis for defects and the test object for testability are tasks performed as part of test analysis, so it is likely they are tasks performed by the testing roleb) Is not correct. The testing role is primarily responsible for the technical and engineering aspects of testing, such as test analysis, test design, test implementation, and test execution. Defining the test environment requirements is a task performed as part of test design, so it is likely to be a task performed by the testing rolec) Is not correct. The testing role is primarily responsible for the technical and engineering aspects of testing, such as test analysis, test design, test implementation, and test execution. Assessing the testability of a test object is a task performed as part of test analysis, so it is likely to be a task performed by the testing roled) Is correct. The test management role primarily involves activities related to test planning, test monitoring and control, and test completion. Thus, creating the test completion report, which is the prime output from the test completion activity, is likely to be a task performed by the test management role | FL-1.4.5 |
|---|----------|

Which of the following statements about the different testing roles is MOST likely to be CORRECT?

- a) In Agile software development, the test management role is the primary responsibility of the team, while the testing role is primarily the responsibility of a single individual from outside the team
- b) The testing role is primarily responsible for test monitoring and control, while the test management role is primarily responsible for test planning and test completion
- c) In Agile software development, test management activities that span multiple teams are handled by a test manager outside the team, while some test management tasks are handled by the team itself
- d) The test management role is primarily responsible for test analysis and test design, while the testing role is primarily responsible for test implementation and execution

<ul style="list-style-type: none">a) Is not correct. Although it is correct to say that in Agile software development, some of the test management tasks may be handled by the Agile team itself, the testing role is not primarily the responsibility of a single individual from outside the team. Instead the testing is more likely to be performed by various team members following the whole-team approachb) Is not correct. The test management role primarily involves activities related to test planning, test monitoring and control, and test completion. So, although this statement is partially correct, it is wrong to say that the testing role is primarily responsible for test monitoring and controlc) Is correct. In Agile software development, some of the test management tasks may be handled by the Agile team itself. However, for test activities that span multiple teams within an organization, test managers outside of the development team may perform these tasksd) Is not correct. The test management role primarily involves activities related to test planning, test monitoring and control, and test completion, while the testing role is primarily responsible for the technical and engineering aspects of testing, such as test analysis, test design, test implementation, and test execution. Thus the test management role is not normally responsible for test analysis and test design, although it is correct to say that the testing role is primarily responsible for test implementation and execution	FL-1.4.5
---	----------

Who normally writes the test plan for a project?

- a. The project manager
- b. The product owner
- c. The test manager
- d. The tester

C is correct. Writing and updating the test plan is normally the responsibility of the test manager.

A, B and D may contribute to the test plan, but the overall responsibility belongs to the test manager.

1.5.1 (K2) Give examples of the generic skills required for testing

Which of the following skills (i-v) are the MOST important skills of a tester?

- i. Having domain knowledge
 - ii. Creating a product vision
 - iii. Being a good team player
 - iv. Planning and organizing the work of the team
 - v. Critical thinking
- a) ii and iv are important
 - b) i, iii and v are important
 - c) i, ii and v are important
 - d) iii and iv are important

Select ONE option.

<p>i. Is true. Having domain knowledge is an important tester skill</p> <p>ii. Is false. This is a task of the business analyst together with the business representative</p> <p>iii. Is true. Being a good team player is an important skill</p> <p>iv. Is false. Planning and organizing the work of the team is a task of the test manager or, mostly in an Agile software development project, the whole team and not just the tester</p> <p>v. Is true. Critical thinking is one of the most important skills of testers Hence b is correct.</p>	FL-1.5.1
---	----------

Which of the following is MOST likely to be an example of a tester using a generic skill when testing?

- a) The tester's deep knowledge of a variety of computer games meant that they got on well with one of the developers who was also into gaming
- b) The tester was a former pilot and was better able to understand the acceptance criteria for the helicopter control system
- c) The tester previously worked as a programmer and used their skills in this area to better communicate with the business analysts
- d) The tester was very careful not to make mistakes when they methodically generated test cases prior to starting their exploratory testing session

Select ONE option.

<p>a) Is not correct. Strong communication skills, active listening, and teamwork abilities enable a tester to interact effectively with all stakeholders, however a deep knowledge of a variety of computer games that allowed them to get on well with one developer is not an example of a generic skill useful to testers</p> <p>b) Is correct. Domain knowledge that can be used to understand and communicate with end-users and business representatives is one of the generic skills required by testers. A tester with experience as a pilot would make them better able to appreciate the acceptance criteria for the helicopter control system</p> <p>c) Is not correct. Although programming skills could be considered as technical knowledge which can increase efficiency when utilizing some test tools, it is unlikely that these skills would improve their communication with business analysts</p> <p>d) Is not correct. Although thoroughness, attention to detail, curiosity, and a methodical approach to identifying hard-to-find defects are all useful generic skills for testers, it is doubtful they would be generating test cases prior to starting exploratory testing. This is because one of the main tenets of exploratory testing is that the test cases are generated during the testing, not scripted in advance</p>	FL-1.5.1
---	----------

1.5.2 (K1) Recall the advantages of the whole team approach

How is the whole team approach present in the interactions between testers and business representatives?

- a) Business representatives decide on test automation approaches
- b) Testers help business representatives to define test strategy
- c) Business representatives are not part of the whole team approach
- d) Testers help business representatives to create suitable acceptance tests

Select ONE option.

<p>a) Is not correct. The test automation approach is defined by testers with the help of developers and business representatives</p> <p>b) Is not correct. The test strategy is decided in collaboration with the developers</p> <p>c) Is not correct. Testers, developers, and business representatives are part of the whole team approach</p> <p>d) Is correct. Testers will work closely with business representatives to ensure that the desired quality levels are achieved. This includes supporting and collaborating with them to help them create suitable acceptance tests</p>	FL-1.5.2
--	----------

Which of the following is an advantage of the whole team approach?

- a) Improved communication between team members
- b) Decreased individual accountability for quality
- c) Faster deployment of deliverables to the end users
- d) Reduced collaboration with external business users

Select ONE option.

- | | |
|---|----------|
| <ul style="list-style-type: none">a) Is correct. The whole team approach promotes robust communication and collaboration between the team membersb) Is not correct. While the whole team approach prioritizes collective accountability for quality, each individual team member is still equally accountable for qualityc) Is not correct. The whole team approach is about how the team works together, with the aim of higher quality deliverables, but it does not necessarily result in faster deployment to end usersd) Is not correct. When using the whole team approach, testers work with business representatives to create acceptance tests. There is no suggestion that the approach will reduce collaboration with external business users | FL-1.5.2 |
|---|----------|

Which of the following is an advantage of the whole-team approach?

- a) Teams with no testers
- b) Improved team dynamics
- c) Specialist team members
- d) Larger team sizes

Select ONE option.

- | | |
|--|----------|
| <ul style="list-style-type: none">a) Is not correct. In the whole-team approach, testers play a vital role by sharing their testing expertise with the team and guiding product development. They collaborate with other team members to achieve the desired quality levels and work with business representatives to create acceptance tests. Testers also partner with developers to determine the optimal test strategy and automation approachesb) Is correct. By leveraging the diverse skill sets of each team member most effectively, the whole-team approach fosters superior team dynamics, promotes robust communication and collaboration, and generates a synergistic effect that benefits the entire projectc) Is not correct. The whole-team approach allows any team member with the requisite skills and knowledge to undertake any task, thus specialist team members are not an advantage of this approachd) Is not correct. There is no specific guidance on the optimum size of teams using the whole-team approach, and there is no suggestion that larger teams are better | FL-1.5.2 |
|--|----------|

Which of the following is an advantage of the whole-team approach?

- a) It allows team members to take on any role at any time
- b) It only needs a single team to support the complete development project
- c) It embeds business representatives alongside developers in the same team
- d) It generates a team synergy that benefits the entire project

Select ONE option.

<ul style="list-style-type: none">a) Is not correct. The whole-team approach allows any team member with the requisite skills and knowledge to undertake any task, however that does not mean that team members can take on any role at any time. Typically, they only take on roles in which they are competent, and there is no suggestion that every team member can do every roleb) Is not correct. The whole-team approach applies to how a single team (typically in Agile software development) works; it does not cover how multiple teams are supposed to work on larger projects, and it does not suggest that only one 'whole' team is needed for a complete projectc) Is not correct. The whole-team approach does not expect every team member to be involved in every important decision. For instance, there is no need for the business representative (i.e. the Product Owner) to be involved in every technical decision that does not affect the business outcome and implementing such an approach would unnecessarily slow down the team's progressd) Is correct. By leveraging the diverse skill sets of each team member most effectively, the whole-team approach fosters superior team dynamics, promotes robust communication and collaboration, and generates a team synergy that benefits the entire project	FL-1.5.2
---	----------

#2. In some Agile teams, people are encouraged to use their skills to help the team, regardless of their role. This could mean that testers help the developers write code and developers help testers test. What is this approach called?

- a. Team Aid
- b. Whole Team
- c. Skills First
- d. First Aid

B is correct. This is an example of the Whole Team approach where the team members leverage their skills, regardless of roles, to accomplish the goals of the team.

1.5.3 (K2) Distinguish the benefits and drawbacks of independence of testing

Which of the following BEST explains a benefit of independence of testing?

- a) The use of an independent test team allows project management to assign responsibility for the quality of the final deliverable to the test team
- b) If a test team external to the organization can be afforded, then there are distinct benefits in terms of this external team not being so easily swayed by the delivery concerns of project management and the need to meet strict delivery deadlines
- c) An independent test team can work separately from the developers, need not be distracted with project requirement changes, and can restrict communication with the developers to defect reporting through the defect management system
- d) When specifications contain ambiguities and inconsistencies, assumptions are made on their interpretation, and an independent tester can be useful in questioning those assumptions and the interpretation made by the developer

Select ONE option.

a) Is not correct. Quality should be the responsibility of everyone working on the project and not the sole responsibility of the test team	FL-1.5.3
b) Is not correct. First, it is not a benefit if an external test team does not meet delivery deadlines, and second, there is no reason to believe that external test teams will feel they do not have to meet strict delivery deadlines	
c) Is not correct. It is bad practice for the test team to work in complete isolation, and we would expect an external test team to be concerned with changing project requirements and communicating well with developers	
d) Is correct. Specifications are never perfect, meaning that assumptions will have to be made by the developer. An independent tester is useful in that they can challenge and verify the assumptions and subsequent interpretation made by the developer	

Given the following benefits and drawbacks of the independence of testing:

- i. The testers work in a different location from the developers
- ii. Testers question the assumptions programmers make while writing code
- iii. A confrontational dynamic has been established between testers and developers
- iv. Developers have convinced themselves that testers are mostly accountable for quality
- v. Testers have different biases than those held by the developers

Which are MOST likely to be considered benefits?

- a) i, iv
- b) ii, v
- c) i, iii, iv
- d) ii, iii, v

Select ONE option.

Considering each of the listed benefits and drawbacks of the independence of testing:

FL-1.5.3

- i. Ideally, we want close collaboration between testers and developers, which is not increased by isolation. Thus, this is a disadvantage
- ii. Testers and developers have varied backgrounds, technical viewpoints, and potential biases, allowing testers to usefully challenge assumptions made by stakeholders during system specification and implementation. Thus, this is an advantage
- iii. The main disadvantage of independence in testing is that testers may become isolated from the development team, leading to communication problems, a lack of collaboration, and potentially an adversarial relationship, with testers being blamed for delays and bottlenecks in the release process. Thus, this is a disadvantage
- iv. One of the disadvantages of independence in testing is that testers may become isolated from the development team, leading to developers feeling less accountable for quality. Thus, this is a disadvantage
- v. The primary benefit of independence in testing is that testers are more likely to identify different types of failures and defects compared to developers, due to their varied backgrounds, technical viewpoints, and potential biases, including cognitive bias

Thus:

- a) Is not correct
- b) Is correct. The list entries showing benefits are ii and v
- c) Is not correct
- d) Is not correct

Which of the following statements about the independence of testing is CORRECT?

- a) Independent testers will find defects due to their different technical perspective from developers, but their independence may lead to an adversarial relationship with the developers
- b) Developers' familiarity with their own code means they only find a few defects in it, however their shared software background with testers means these defects would also be found by the testers
- c) Independent testing requires testers who are outside the developer's team and ideally from outside the organization, however these testers find it difficult to understand the application domain
- d) Testers from outside the developer's team are more independent than testers from within the team, but the testers from within the team are more likely to be blamed for delays in product release

Select ONE option.

- a) Is correct. The primary benefit of independence in testing is that testers are more likely to identify different types of failures and defects compared to developers, due to their varied backgrounds, technical viewpoints, and potential biases, including cognitive bias. However, the main disadvantage of independence in testing is that testers may become isolated from the development team, leading to communication problems, a lack of collaboration, and potentially an adversarial relationship, with testers being blamed for delays and bottlenecks in the release process
- b) Is not correct. A developer's familiarity with the code does not mean that they rarely find defects in it, instead this familiarity means they can efficiently find many defects in their own code. And, rather than developers and testers having a shared background, developers having a different background to testers is normally cited as the reason that testers and developers find different kinds of defects
- c) Is not correct. Testing can be conducted at different levels of independence, ranging from no independence for the author to very high independence for testers from outside the organization. In most projects, multiple levels of independence are utilized, with developers performing component and component integration testing, the test team performing system and system integration testing, and business representatives performing acceptance testing. So, testers can be in the developer's team and do not need to come from outside the organization. Knowledge of the application domain will change from case-to-case and is not dependent on the level of independence
- d) Is not correct. Testing can be conducted at different levels of independence, ranging from no independence for the author to very high independence for testers from outside the organization, with testers from outside the developer's team generally more independent than testers from within the team. However, there is more reason to believe that testers from outside the team are likely to be more isolated from the developers and so are more likely to be blamed for delays in product release

FL-1.5.3

What is the biggest problem with a developer testing his own code?

- a. Developers are not good testers
- b. Developers are not quality focused
- c. Developers are not objective about their own code
- d. Developers do not have time to test their own code

C is correct. This is the biggest problem because developers have biases toward the accuracy and implementation of their own code. Testers and developers think differently and testers can be more objective as they are not invested in the code.

A and B are not necessarily true – some developers are good testers and have a good quality focus.

D is not correct because unit testing is part of their job and time should be made in the schedule for at least unit testing.

Conclusion

In this document :

- **We identified the examinable learning objectives.**
- **We presented the probability of questions for each part of chapter 1.**
- **We summarized chapter 1.**
- **We provided section-wise questions and their answers for chapter 1.**

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



Karim Kalboussi
kalboussikarim3@gmail.com