Question 5.8
(FL-5.2.1, K1)

Correct answer: A

Answer A is correct: the risk likelihood and risk impact are independent factors.
Answer B is incorrect: see the rationale for answer A.
Answer C is incorrect: see the rationale for answer A.
Answer D is incorrect: the risk impact should be assessed before the risk occurs.

Question 5.9
(FL-5.2.2, K2)

Correct answer: B

Answer A is incorrect. This is an extreme example of the product risk materialization.

Answer B is correct. The occurrence of project risks often results in problems related to delays in project tasks.

Answer C is incorrect. High software maintenance costs are due to maintainability defects in the product, that is, they are the consequence of product risks, not project risks.

Answer D is incorrect. Customer dissatisfaction is due to a product defect, so it is a consequence of product risks, not project risks.

Question 5.10
(FL-5.2.3, K2)

Correct answer: A

Answer A is correct. A technical review to look at possible usability problems that are associated with the interface seems to be a reasonable idea, as is checking the control flow in the transfer mechanism code by applying branch coverage.

Answer B is incorrect. Component testing may not show problems related to transfer interface—we don't know what the application architecture looks like. Component testing is too low test level for this.

Answer C is incorrect. Usability is not related to the implementation of the business logic of the application.

Answer D is incorrect. White-box testing does not address the risks related to interface usability.

Question 5.11
(FL-5.2.4, K2)

Correct answer: C

Buying insurance transfers the burden of risk to a third party, in this case the insurer. This is an example of risk transfer, so the correct answer is answer C.

Question 5.12
(FL-5.3.1, K1)

Correct answer: B

Answer A is incorrect. The residual risk level is a typical metric used in testing, as it expresses the current level of residual risk in a product after the testing cycle.

Answer B is correct. Coverage of requirements by source code has nothing to do with testing. This metric can represent the progress of development work, not testing.

Answer C is incorrect. The number of critical defects found is directly related to testing.

Answer D is incorrect. The progress of the test environment implementation refers to an important activity performed as part of the testing process, so it is a metric used in testing.

Question 5.13
(FL-5.3.2, K2)

Correct answer: D

Answer A is incorrect. Information about unmitigated risks is a typical information contained in a test completion report.

Answer B is incorrect. Deviations from test plan are a typical information contained in a test completion report.

Answer C is incorrect. Information about defects is a typical information contained in a test completion report.

Answer D is correct. The typical information included in a test progress report is testing scheduled for the next reporting period. This is not a typical information included in a test completion report, as this type of report describes a closed, completed scope of work for which there will be no further reporting periods.

Question 5.14
(FL-5.3.3, K2)

Correct answer: B

There is no single best method of communication. For example, formal reports or e-mails will not be useful when the team needs to communicate quickly, frequently, and in real time. On the other hand, "face-to-face," verbal communication will be impossible when the team is dispersed and works in many different time zones. The form of communication should always be chosen individually, according to the circumstances, taking into account various contextual factors. Therefore, the correct answer is B.

Question 5.15
(FL-5.4.1, K2)

Correct answer: A

Answer A is correct. The purpose of configuration management is to ensure and maintain the integrity of the component/system, the related testware, and the interrelationships between them throughout the project and software life cycle so that the activities described in the scenario can be performed.

Answer B is incorrect. Impact analysis can determine the magnitude or risk of a change, but it does not identify the source work products based on the software version.

Answer C is incorrect. Continuous delivery helps automate the software release process, but it does not ensure the integrity and versioning of work products.

Answer D is incorrect. Retrospectives are for process improvement; they do not provide integrity or versioning of work products.

Question 5.16
(FL-5.5.1, K3)

Correct answer: A

Answer A is correct. The report lacks information on the steps to reproduce the test. For example, the developer may not know why the report says about accepting the passwords of length 6, 7, and 9, but not of length 8. Steps to reproduce might allow the developer to verify this quickly.

Answer B is incorrect: The version of the product (in the form of the last compilation date) is given in the defect report.

Answer C is incorrect. When we create a defect report using a defect management tool, it is most likely that an "open" status is automatically assigned. In addition, this is not as crucial information as the one given in answer A.

Answer D is incorrect. This information is useful to the tester, but does not have to be included in the defect report and may be of little value for the developer who is responsible for fixing the defect.

Answers to Questions from Chap. 6

Question 6.1
(FL-6.1.1, K2)

Correct answer: B

According to the syllabus (Sect. 6.1), tools to support the test management include requirements management tools. Thus, the correct answer is B.

Question 6.2
(Fl-6.1.2, K1)

Correct answer: C, D

According to the syllabus, the benefits are, in particular, increased consistency and repeatability of testing (C) and objective evaluation through the use of well-defined operational definitions of measurement (D). Overdependence on the tool (A), dependence on the vendor (B), and errors in estimating the cost of maintaining the tool (E) are risks. Hence, the correct answers are C and D.

Solutions to Exercises

Solutions to Exercises from Chap. 4

Exercise 4.1
(FL-4.2.1, K3)

A) We have two domains: band = {Iron Maiden, Judas Priest, Black Sabbath} and ticket type = {in front of stage, away from stage}. Each element from each domain will be a separate one-element equivalence partition. Thus, the division of the "band" domain is as follows: {Iron Maiden}, {Judas Priest}, {Black Sabbath}; the division of the "ticket type" domain is as follows: {in front of stage}, {away from stage}.

B) There are no invalid partitions in the problem, because the way the band and ticket type are selected makes this impossible: the user selects these values from predefined drop-down lists. Of course, it is possible to consider a situation in which, after submitting the form, this query is intercepted and modified, e.g., the team name is changed to a non-existent one. Here, however, we focus only on purely functional testing, restricting the user-system interaction to GUI only, and we do not consider advanced application security testing issues.

C) The test suite must cover each of the three partitions of the "band" domain and each of the two partitions of the "ticket type" domain. Since each test case covers one partition from each of these domains, three test cases will be enough, e.g.:

TC1 band = Iron Maiden, ticket type = in front of stage.
TC2 band = Judas Priest, ticket type = away from stage.
TC3 band = Black Sabbath, ticket type = in front of stage.

It should be also pointed out that in each test case, you also need to specify the expected output. In our case, it will be the assignment of a ticket of a specific type for a concert by a specific band.

L. Stapp et al., *ISTQB® Certified Tester Foundation Level*,
https://doi.org/10.1007/978-3-031-42767-1_11

Exercise 4.2
(FL-4.2.1, K3)

A) The input (valid) domain is the natural numbers greater than 1. We can divide this domain into two partitions: prime numbers and composite numbers, namely, {2, 3, 5, 7, 11, 13, ...} and {4, 6, 8, 9, 10, 12, ...}.
B) The interface does not allow the user to enter a value that is not a natural number. Therefore, we can consider that there are no invalid partitions in the problem. However, if we were able, for example, to intercept the message that passes the input value to the system and modify it accordingly, we could then force an invalid value. Whether this is possible and whether testers choose to take such a "hacking" approach depends, of course, on a number of factors, in particular the required level of application security. If we consider the problem only from the user's perspective, we can safely assume that the program will operate only on correct, expected values.
C) Since there are no invalid partitions in the problem, two tests are enough to cover all equivalence partitions: one in which we consider a prime number and one in which we consider a composite number, e.g.:

TC1: input is a prime number (e.g., 7),
TC2: input is a composite number (e.g., 12).

Exercise 4.3
(FL-4.2.2, K3)

The domain under analysis is the total amount of purchases. It is a positive number measurable to two decimal places (that is, to 1 cent). We need to find those of these values which, after rounding, will be the boundary values for the rounded amount (see Table 1). The smallest possible input value that meets the conditions of the task is $0.01.

The value 300 is *the largest* amount, which after rounding (=300) will give the maximum boundary value for 0% discount. The value 300.01 is the *smallest* amount that, *after rounding* (=301), will give the minimum boundary value for a 5% discount, etc.

So we have the following test cases:

- TC1: amount = 0.01, expected result: 0% discount.
- TC2: amount = 300, expected result: 0% discount.
- TC3: amount = 300.01, expected result: 5% discount.

Table 1 Boundary values before and after rounding

	Boundary values for the amount rounded up		Boundary values for the amount *before rounding*	
Discount	Minimum	Maximum	Minimum	Maximum
0%	1	300	0.01	300
5%	301	800	300.01	800
10%	801	–	800.01	–

- TC4: amount = 800, expected result: 5% discount.
- TC5: amount = 800.01, expected result: 10% discount.

Exercise 4.4

(FL-4.2.2, K3)

A)

Partitions for the "width" parameter:

- Valid partition: {30, 31, ..., 99, 100}

Partitions for the "height" parameter:

- Valid partition: {30, 31, ..., 59, 60}

Partitions for the "area" (the price of the service will depend on its value):

- Valid partition for the price of $450: {900, 901, ..., 1600}
- Valid partition for the price of $500: {1601, 1602, ..., 6000}

The values for "area" are derived from the fact that the minimum dimensions of the image are 30 cm wide and 30 cm high, so the minimum surface area is 30 cm*30 cm = 900 cm^2. Similarly, the maximum dimensions are 100 cm width and 60 cm height, so the maximum area is 60 cm*100 cm = 6000 cm^2.

Boundary values:

- For "width": (W1) 30, (W2) 100.
- For "height": (H1) 30, (H2) 60.
- For "area": (A1) 900, (A2) 1600, (A3) 1601, (A4) 6000.

B)

We will represent the test case as a pair (w, h), where w and h are the width and height (input), respectively. We need to cover eight boundary values with tests: W1, W2, H1, H2, A1, A2, A3, and A4. Note that some boundary values for the area can be obtained from multiplying the values that are the height and width boundary values. For example, 900 = 30*30, and 6000 = 100*60. We can use this to minimize the number of test cases.

The test cases and covered boundary values are shown in Table 2.

Note that it is impossible to cover the boundary value of A3. This is because 1601 is a prime number, so it cannot be expressed as the product of two numbers greater than or equal to 30.

Table 2 Covered boundary values

	Input			Covered boundary values for:		
TC	Width	Height	Area	Width	Height	Area
1	30	30	900	W1	H1	A1
2	100	60	6000	W2	H2	A4
3	40	40	1600			A2

We designed three test cases TC1, TC2, and TC3, covering seven of the eight identified boundary values. The smallest number belonging to the partition {1601, 1602, . . ., 6000} that can be represented as a multiplication of two numbers fulfilling the given constraints, is 1610 = 35 * 46. We could add the fourth test case (35, 46) that exercises this 'feasible' boundary value for the partition {1601, 1602, . . ., 6000}.

Exercise 4.5
(FL-4.2.3, K3)

A) The conditions occurring in our problem are "points ≥ 85" (possible values: YES, NO) and "number of errors ≤ 2" (possible values: YES, NO).
B) The system can take the following actions:

- Grant a driver's license (YES, NO)
- Repeat the theory exam (YES, NO)
- Repeat the practical exam (YES, NO)
- Additional driving lessons (YES, NO)

C) All combinations of conditions are shown at the top of Table 3 and can be generated using the "tree" method described earlier in this chapter. Since we have two conditions and each of them takes one of the two possible values, we have 2*2 = 4 combinations of their values. All combinations are feasible.
D) The full decision table is shown in Table 3.

It is easy to see that the various actions follow directly from the provisions of the specification. For example, if the number of points for the theoretical exam is 85 or more, and the candidate has made at most two errors (column 1), this means that a driver's license should be granted (action "grant a driver's license" = YES), and the candidate should not repeat any exams or take additional lessons (other actions = NO).

E) Sample test cases generated from the decision table might look like the following:

Test case 1 (corresponding to column 1)
Name: grant the driver's license.
Pre-conditions: the candidate took the exams for the first time.
Input: theoretical exam score = 85 points, number of errors made = 2.

Table 3 Decision table for the driving test support system

	1	2	3	4
Conditions				
Points ≥ 85?	YES	YES	NO	NO
Errors ≤ 2?	YES	NO	YES	NO
Actions				
Grant a driver's license?	YES	NO	NO	NO
Repeat the theory exam?	NO	NO	YES	YES
Repeat the practical exam?	NO	YES	NO	YES
Additional driving lessons?	NO	NO	NO	YES

Expected output: granting of a driver's license, no need to repeat exams, no need to take additional driving lessons.
Post-conditions: candidate marked as a candidate who has already taken the exams.

Test case 2 (corresponding to column 2)
Name: pass theoretical exam, fail practical exam.
Pre-conditions: the candidate took the exams for the first time.
Input: theoretical exam score = 93 points, number of errors made = 3.
Expected output: driver's license not granted, the candidate has to repeat the practical exam, does not have to take additional driving lessons.
Post-conditions: candidate marked as a candidate who has already taken the exams.

Test case 3 (corresponding to column 3)
Name: fail theoretical exam, pass practical exam.
Pre-conditions: the candidate took the exams for the first time.
Input: theoretical exam score = 84 points, number of errors made = 0.
Expected output: driver's license not granted, the candidate has to repeat the theoretical exam, does not have to take additional driving lessons.
Post-conditions: Candidate marked as a candidate who has already taken the exams.

Test case 4 (corresponding to column 4)
Name: fail both exams.
Pre-conditions: the candidate took the exams for the first time.
Input: theoretical exam score = 42 points, number of errors made = 3.
Expected exit: driver's license not granted, the candidate has to repeat both theoretical and practical tests and also has to take additional driving lessons.
Post-conditions: candidate marked as a candidate who has already taken the exams.

Note that the post-conditions are not out of the question here—perhaps the system has a completely different set of behaviors toward candidates who retake the exam. For example, the system could then check whether the candidate has actually taken additional driving lessons, and this would be part of the input to the system.

Exercise 4.6
(FL-4.2.3, K3)

In Fig. 4.19, describing the process, the rhombuses represent conditions, and rectangles represent actions. We have three conditions: "Does a passenger have a gold card?" "Is economy class full?" and "Is business class full?" The possible actions are as follows: "Issue a boarding pass?" "Type of seat?" [economy (E) or business (B)] and "Is the passenger removed from the passenger list?"

The corresponding decision table is shown in Table 4.

The actions were assigned based on the diagram in Fig. 4.19. Note that the action "type of seat" is not a Boolean variable—its possible values are E, B, and N/A ("not applicable," meaning that no seat can be allocated, because the passenger is removed from the passenger list).

According to the business rules, when a passenger has a gold card and business class is full, they should be assigned a seat in economy class. Columns 1 and

Table 4 Decision table for Exercise 4.6

Conditions	1	2	3	4	5	6	7	8
Has a gold card?	YES	YES	YES	YES	NO	NO	NO	NO
Economy class full?	YES	YES	NO	NO	YES	YES	NO	NO
Business class full?	YES	NO	YES	NO	YES	NO	YES	NO
Actions								
Issue a boarding pass?	YES	YES	YES	YES	NO	YES	YES	YES
Type of seat	E	B	E	B	N/A	B	E	E
Remove from the passenger list?	NO	NO	NO	NO	YES	NO	NO	NO

3 correspond to this situation (gold card = YES, business class full = YES). Let us look carefully at column 1. It describes the situation when economy class is also full. Nevertheless, according to the specification, the system tells the passenger to allocate a seat in this class (see cells with gray background)!

We have discovered a serious error in the specification. We do not know how this problem should be solved. Here are some possible solutions:

- Remove another passenger without a gold card from economy class, and assign their seat to the customer under consideration (the question, however, is what if every passenger in economy class has a gold card? Arguably, this is a very unlikely situation, but possible—the specification should take this into account).
- Add an additional condition to the table—if the customer has a gold card and the business class is full, consider whether the economy class is full. If not, we assign the passenger a seat in this class, as described in column 3. However, if the economy class is full, then we need to take another action, such as removing the passenger from the list.

Exercise 4.7

(FL-4.2.4, K3)

A) In the first step, let us analyze what states the system may be in. We can identify the following eight states (they basically follow directly from the scenario analysis):

- Welcome screen—initial system status, waiting for card insertion.
- Card validation—the state in which the system verifies the inserted card.
- End—the state to which the system will go after validation if the card is wrong.
- Ask for PIN—the state in which the system asks to enter the PIN for the first time.
- Ask for PIN second time—the state in which the system asks to enter the PIN for the second time (after entering PIN incorrectly the first time).

Table 5 Possible events for each state of the state machine

State	Possible events and actions
Welcome screen	InsertCard
Card validation	CardOK (transition to Ask for PIN state) InvalidCard (actions: return card, display "card error" message; transition to End state)
End	—
Ask for PIN	PinOK (transition to Logged state) InvalidPIN (action: "PIN error" message; transition to "Ask for PIN 2nd time" state)
Ask for PIN second time	PinOK (transition to Logged state) InvalidPIN (action: "PIN error" message; transition to "Ask for PIN 3rd time" state)
Ask for PIN third time	PinOK (transition to Logged state) InvalidPIN (actions: "PIN error" message, "card locked" message; transition to "Card blocked" state)
Logged	—
Card blocked	—

- Ask for PIN third time—a state in which the system asks to enter the PIN for the third time (after entering PIN incorrectly the second time).
- Logged—the state to which the system goes after correctly entering the PIN the first, second, or third time.
- Card blocked—the state to which the system goes after the PIN is entered incorrectly three times.

Note that in this state transition model, we need to define as many as three states related to waiting for PIN entry, because our state model has no memory—the representation of the history of past events is only the state we are currently in. So, in order to distinguish the number of incorrectly entered PIN codes, we need three states.

Now let us consider the possible events that can occur in our system and the actions the system can take to handle these events. One of the most convenient ways to do this is to analyze individual states and think about what can happen (based on the specification) while we are in a given state. The results of our analysis are presented in Table 5.

Based on Table 5, we can design a state transition diagram. It is shown in Fig. 1.

B) The transition diagram using guard conditions is shown in Fig. 2. Note that by introducing guard conditions, we are able to reduce the number of states. Now we have only one state related to asking for a PIN, and the number of incorrectly entered PINs is remembered in a variable named "attempts." Whether we go from the state "Ask for PIN" to the state "Card blocked" or "Logged" depends on how many times the wrong PIN was entered. Note that the loop

Ask for PIN (InvalidPIN) Ask for PIN

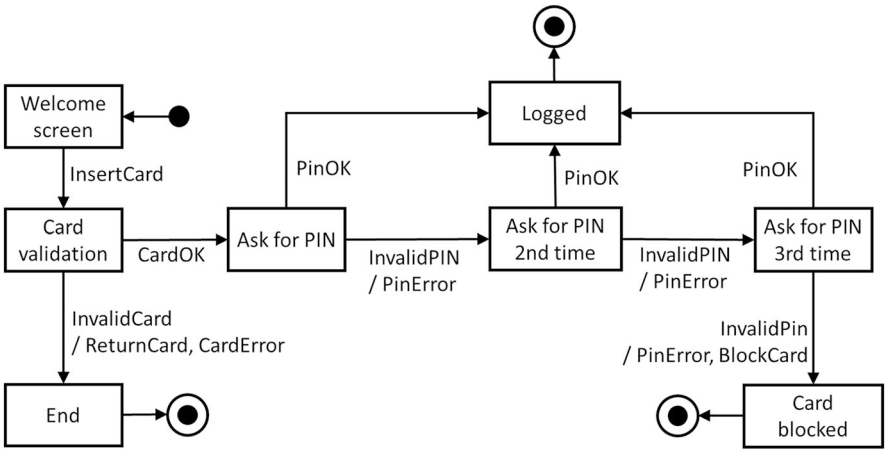


Fig. 1 Transition diagram for PIN verification without using guard conditions

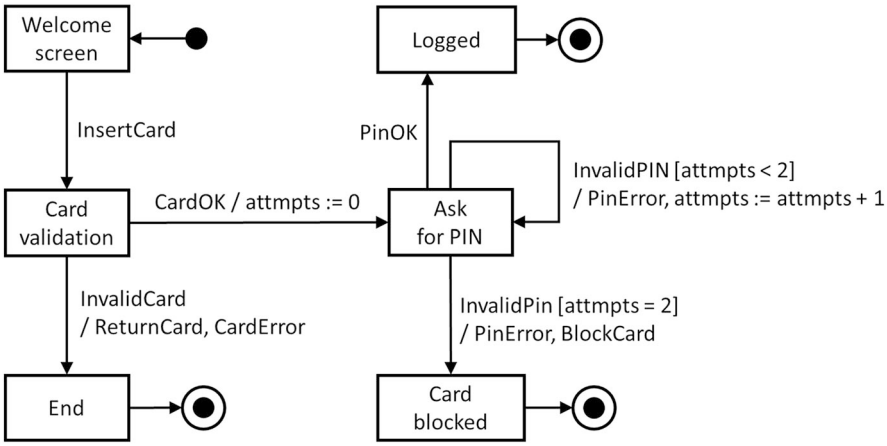


Fig. 2 Transition diagram for PIN verification using guard conditions

can be executed a maximum of two times. This is because each time we execute this loop transition, the value of the "attempt" variable increases by one, and the guard condition allows us to exercise this loop only if this variable has a value less than 2. After the PIN has been entered incorrectly twice, this variable has a value of 2, and at the time of the third failed attempt to enter the PIN, the guard condition is false. Instead, the guard condition on the transition to "Card blocked" becomes true, so after the third failed attempt to enter the PIN, the system does not stay in "Ask for PIN" again but goes to "Card blocked" state.

C) Before we move on to test design, let us consider what test conditions we have for both types of coverage. In the case of all states coverage, we have the following items (states) to cover: Welcome screen, Card validation, End, Ask for PIN, Ask for PIN second time, Ask for PIN third time, Logged, and Card blocked. So we need to cover eight coverage items with tests. For valid transitions coverage, we need to cover all arrows between states. There are nine of them: InsertCard, InvalidCard, CardOK, three different transitions triggered by PinOK, and three different transitions triggered by InvalidPIN. So to achieve full valid transitions coverage, we need to cover nine coverage items with our test cases.

Regarding all states coverage, let us note that in the diagram from Fig. 1, we have three final states. So we will need at least three test cases, since reaching the final state terminates the test execution. Here are sample three test cases that achieve full all states coverage:

TC1: Welcome screen (InsertCard) Card validation (InvalidCard) End
TC2: Welcome screen (InsertCard) Card validation (CardOK) Ask for PIN (PinOK) Logged
TC3: Welcome screen (InsertCard) Card validation (CardOK) Ask for PIN (InvalidPIN) Ask for PIN second time (InvalidPIN) Ask for PIN third time (PinWrong) Card blocked

These three cases cover all states (TC1 covers three of them, TC2 covers additional two, and TC3 covers additional three), but do not cover all transitions. The two transitions not covered are:

Ask for PIN second time (PinOK) Logged
Ask for PIN third time (PinOK) Logged

We need to add two new tests that cover these two transitions, such as:

PT4: Welcome screen (InsertCard) Card validation (CardOK) Ask for PIN (InvalidPIN) Ask for PIN second time (PinOK) Logged
PT5: Welcome screen (InsertCard) Card validation (CardOK) Ask for PIN (InvalidPIN) Ask for PIN second time (InvalidPIN) Ask for PIN third time (PinOK) Logged

Again, note that we cannot achieve valid transitions coverage with less than five test cases, because we have five transitions directly reaching the end states. This means that the execution of such a transition ends the test case, so no two of these five transitions can be within a single test case.

Exercise 4.8
(FL-4.2.4, K3)

A) The system has three states (S, W, B) and five different events (Silence!, Bark!, Down!, SeesCat, IsPetted). So we have 3*5 = 15 combinations of (state, transition) pairs. Since we can see from the diagram that there are six valid transitions (six arrows between states), so there must be 15 − 6 = 9 invalid transitions:

- (S, Silence!)
- (S, Down!)
- (B, Bark!)
- (B, Down!)
- (B, SeesCat)
- (W, Silence!)
- (W, Bark!)
- (W, IsPetted)
- (W, SeesCat)

B) Valid transitions can be tested using one test case, for example:

S (Bark!) B (Silence!) S (SeesCat) B (IsPetted) W (Down!) S (IsPetted) W

This test case covers all six valid transitions. Since we have nine invalid transitions, we need to add nine test cases, one for each invalid transition. For example, in order to cover the invalid transition (W, Bark!), we can design the case:

S (IsPetted) W (Bark!) ?

After reaching the W state, we attempt to trigger the (invalid) "Bark!" event. If it is infeasible, or if the system ignores it, we assume that the test passes. If, on the other hand, the "Bark!" event can be invoked and the system changes its state, the test fails, since this is not the expected behavior.

Note that we always start in the initial state, so first we need to exercise the valid transitions to reach the desired state (in our case, the W state) and then attempt to trigger the invalid transition (in our case, Bark!). Similarly, we design test cases for the remaining eight invalid transitions.

Exercise 4.9
(FL-4.5.3, K3)

Here are some examples of test cases we can design:

- Successful registration with valid login (not yet used in the system) and valid password, e.g., login john.smith@mymail.com, password Abc123Def typed twice in both fields, expected result: the system accepts the data and sends an email to john.smith@mymail.com with a link to activate the account (this test verifies the acceptance criteria AC1 and AC5).
- Attempted registration with syntactically incorrect login (covers acceptance criterion AC1). Test data sets:
  - John.Smith@post (too short part after the @ sign)
  - JohnSmith-mymail.com (no @ sign)
  - @mymail.com (no text before the @ sign)
  - JohnSmith@mymail..com (two consecutive dots after the @ sign)

- Attempt to register with a valid but existing login. Expected result: registration denied, no email sent (this test verifies acceptance criterion AC2).
- Attempt to register with valid login, but wrong password. Expected result: registration denied, no email sent. Test data sets (covering AC3 and AC4 acceptance criteria):
  - Syntactically correct password, but different in both fields for the password (e.g., Abc123Def and aBc123Def)
  - Password too short (e.g., Ab12)
  - Password too long (e.g. ABCD1234abcd1234)
  - Password without digits (e.g. ABCdef)
  - Password without capital letters (e.g. abc123)

Solutions to Exercises from Chap. 5

Exercise 5.1
(FL-5.1.4, K3)

The last iteration of the poker for the optimistic value (a) yielded values of 3, 3, and 5, which, according to the procedure described, means that the experts reached a consensus on the optimistic value. It is 3, as this is the value indicated by the majority of experts.

Similarly, for the most likely value (m), the experts reached a consensus after the first iteration. The result is 5, as two of the three experts indicated this value.

For the pessimistic value (b), the experts reached consensus only in the third iteration. The result is 13, as it was indicated by all experts.

After these poker sessions, experts determined the values of the variables to be used in the three-point estimation technique, namely:

- Optimistic value: a = 3
- Most likely value: m = 5
- Pessimistic value: b = 13

Substituting these values into the formula from the three-point method, we get:

$$E = (3 + 4^{*}5 + 13)/6 = 36/6 = 6$$

meaning that the estimated effort is 6 person-days, with a standard deviation of

$$SD = (b - a)/6 = (13 - 3)/6 = 10/6 = 1.66.$$

This means that the final estimation result is 6 ± 1.66 person-days, which is between 4.33 and 7.66 person-days.

Exercise 5.2
(FL-5.1.4, K3)

In the completed project, the effort for the design phase was 20 person-days (because 4 people did the work in 5 days). Similarly, for the implementation phase, the effort was 10*18 = 180 person-days and for testing 4*10 = 40 person-days. Thus, the total effort was 240 person-days in the ratio design/programming/testing equal to 1:9:2.

Let x = number of days of work of designers and y = number of days of work of developers. Then 66 − x − y is the number of work days of testers (because the total project is expected to last 66 days). Given the number of designers, developers, and testers in the new project, the effort in the design, implementation, and testing phases is therefore 4x, 6y, 2*(66–x–y), respectively.

From the ratio, we have 4*x : 6*y : 2*(66–x–y) = 1:9:2. So, we have 2*4*x = 2*(66–x–y) and that 9*4*x = 6*y, since testing takes twice as much effort as designing and implementation takes nine times as much effort as planning. We must now solve the system of two equations:

$$8^{*}x = 132 - 2^{*}x - 2^{*}y$$

$$36^{*}x = 6^{*}y$$

From the second equation, we have y = 6x. Substituting into the first equation, we get:

$$8^{*}x = 132 - 2^{*}x - 12^{*}x, \text{ or } 22^{*}x = 132, \text{ from where we calculate } x = 132/22 = 6.$$

By substituting x = 6 into the relation y = 6x, we get that y = 6*6 = 36.

Since x, y, and 66−x−y represent the number of days spent on design, programming, and testing, respectively, we get the final answer: using the ratio method, that is, assuming that the effort in the new project will be distributed proportionally to the effort in the completed project:

- We must allocate x = 6 days for design.
- We must allocate y = 36 days for development.
- We must allocate 66–x–y = 66 − 42 = 24 days for testing.

Exercise 5.3
(FL-5.1.5, K3)

If we only considered priorities, the order of test case execution would be as follows:

TC001 → TC003 → TC002 → TC004.

However, we must also take into account logical dependencies—we cannot execute TC001 before TC002 and TC003.

Logical dependencies force us to start with TC002 (personal data entry), because it alone is not dependent on any other case. We still can't execute TC001 with the

highest priority, because it depends not only on 002 but also on 003. So we have to execute case 003 in the next step, which will unlock the possibility of executing TC001. At the very end, we run TC004.

Thus, the final order is: TC002 → TC003 → TC001 → TC004.

Exercise 5.4
(FL-5.1.5, K3)

If we were to rely solely on the client's priorities, we should implement and test requirements Req2, Req3, and Req6 first (high priority), then Req1 and Req5 (medium priority), and finally Req4 (low priority). However, regardless of the priorities, the first requirement to be implemented and tested must be Req1, as it is the only requirement that does not depend on other requirements. The only requirement that can be implemented and tested next is Req3 (because it is the only one that depends only on the already implemented and tested Req1). With Req1 and Req3 tested, we have to implement and test Req2, as it is the only requirement that depends on already implemented and tested requirements.

Note that up to this point, priorities have played no role. But now we can implement and test Req4 or Req5. Let us note that among the yet unimplemented and untested requirements, the requirement with the highest priority for the customer is Req6, and it is dependent on Req4. So we implement and test Req4 first, in order to "unlock" the possibility of implementing and testing the highly prioritized Req6 as soon as possible. At the very end, we implement and test Req5.

The final order of requirements is as follows:

Req1, Req3, Req2, Req4, Req6, Req5.

The order takes into account the customer's priorities while also taking into account the necessary logical relationships between requirements.

Exercise 5.5
(FL-5.5.1, K3)

The defect report should contain at least the information described in Table 6.

Table 6 Defect report content

Unique identifier	34.810
Title, summary of defect reported	Incorrect calculation of the amount to be paid
Date of report (= date of error discovery)	07.09.2023
Author	Carol Beer
Description of the defect to allow for reproduction and repair	Test case PT003
Actual result	$48
Expected result	$47.50
Priority for defect removal	Normal
A description of the nonconformity to help determine its cause	Seems that the system rounds the resulting amount to full $

Part IV
Official Sample Exam

This part of the manual contains a sample exam. This is the official sample exam published by ISTQB®. The duration of the exam is 60 minutes, or 75 minutes if the exam is not taken in the native language.

The original ISTQB® sample exam document contains the exam questions in the order of the corresponding learning objectives in the syllabus. This may make it easier to answer some of the questions. In the actual exam, the questions are arranged in random order. Therefore, in this chapter, the sample exam questions are placed in random order to best reflect the reality of the exam.

Exam Set A

Question #1 (1 point)
You are testing a system that calculates the final course grade for a given student. The final grade is assigned based on the final result, according to the following rules:

- 0–50 points: failed.
- 51–60 points: fair.
- 61–70 points: satisfactory.
- 71–80 points: good.
- 81–90 points: very good.
- 91–100 points: excellent.

You have prepared the following set of test cases:

	Final result	Evaluation final
TC1	91	Excellent
TC2	50	Failed
TC3	81	Very good
TC4	60	Fair
TC5	70	Satisfactory
TC6	80	Good

What is the 2-value boundary value analysis (BVA) coverage for the final result that is achieved with the existing test cases?

a) 50%
b) 60%
c) 33.3%
d) 100%

Select ONE option.

L. Stapp et al., *ISTQB® Certified Tester Foundation Level*,
https://doi.org/10.1007/978-3-031-42767-1_12

Question #2 (1 point)
Which of the following is a benefit of early and frequent feedback?

a) It improves the test process for future projects.
b) It forces customers to prioritize their requirements based on agreed risks.
c) It is the only way to measure the quality of changes.
d) It helps avoid requirements misunderstandings.

Select ONE option.

Question #3 (1 point)
Your favorite bicycle daily rental store has just introduced a new Customer Relationship Management system and asked you, one of their most loyal members, to test it. The implemented features are as follows:

- Anyone can rent a bicycle, but members receive a 20% discount.
- However, if the return deadline is missed, the discount is no longer available.
- After 15 rentals, members get a gift: a T-Shirt.

Decision table describing the implemented features looks as follows:

	R1	R2	R3	R4	R5	R6	R7	R8
Conditions								
Being a member	T	T	T	T	F	F	F	F
Missed deadline	T	F	T	F	T	F	F	T
15th rental	F	F	T	T	F	F	T	T
Actions								
20% discount		X		X				
Gift T-shirt			X	X				X

Based ONLY on the feature description of the Customer Relationship Management system, which of the above rules describes an impossible situation?

a) R4
b) R2
c) R6
d) R8

Choose ONE answer.

Question #4 (1 point)
You need to update one of the automated test scripts to be in line with a new requirement. Which process indicates that you create a new version of the test script in the test repository?

a) Traceability management.
b) Maintenance testing.
c) Configuration management.
d) Requirements engineering.

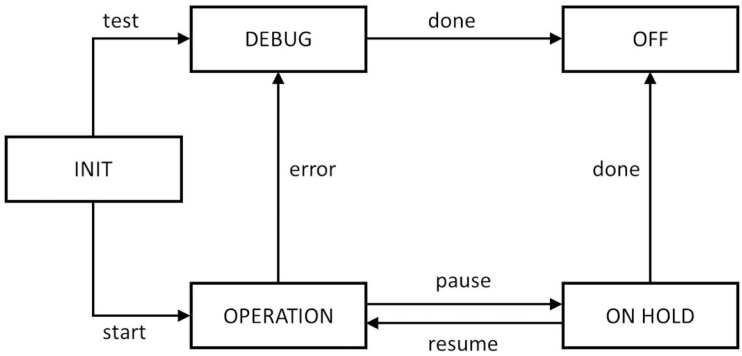
Select ONE option.

Question #5 (1 point)
Which of the following is a characteristic of experience-based test techniques?

a) Test cases are created based on detailed design information.
b) Items tested within the interface code section are used to measure coverage.
c) The techniques heavily rely on the tester's knowledge of the software and the business domain.
d) The test cases are used to identify deviations from the requirements.

Select ONE option.

Question #6 (1 point)
You test a system whose life cycle is modeled by the state transition diagram shown below. The system starts in the INIT state and ends its operation in the OFF state.



What is the MINIMAL number of test cases to achieve valid transitions coverage?

a) 4
b) 2
c) 7
d) 3

Select ONE option.

Question #7 (1 point)
You received the following defect report from the developers stating that the anomaly described in this test report is not reproducible:

Application hangs up

2022-May-03—John Doe—Rejected

The application hangs up after entering "Test input: $ä" in the Name field on the new user creation screen. Tried to log off, log in with test_admin01 account, same issue. Tried with other test admin accounts, same issue. No error message received; log (see attached) contains fatal error notification. Based on the test case TC-1305, the application should accept the provided input and create the user. Please fix with

high priority, this feature is related to REQ-0012, which is a critical new business requirement.

What critical information is MISSING from this test report that would have been useful for the developers?

a) Expected result and actual result.
b) References and defect status.
c) Test environment and test item.
d) Priority and severity.

Select ONE option.

Question #8 (1 point)
How do testers add value to iteration and release planning?

a) Testers determine the priority of the user stories to be developed.
b) Testers focus only on the functional aspects of the system to be tested.
c) Testers participate in the detailed risk identification and risk assessment of user stories.
d) Testers guarantee the release of high-quality software through early test design during the release planning.

Select ONE option.

Question #9 (1 point)
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.

Select ONE option.

Question #10 (1 point)
Which tool can be used by an agile team to show the amount of work that has been completed and the amount of total work remaining for a given iteration?

a) Acceptance criteria.
b) Defect report.
c) Test completion report.
d) Burndown chart.

Select ONE option.

Question #11 (1 point)
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.

Select ONE option.

Question #12 (1 point)
Which item correctly identifies a potential risk of performing test automation?

a) It may introduce unknown regressions in production.
b) Sufficient efforts to maintain testware may not be properly allocated.
c) Testing tools and associated testware may not be sufficiently relied upon.
d) It may reduce the time allocated for manual testing.

Select ONE option.

Question #13 (1 point)
Which TWO of the following options are the exit criteria for testing a system?

a) Test environment readiness.
b) The ability to log in to the test object by the tester.
c) Estimated defect density is reached.
d) Requirements are translated into given/when/then format.
e) Regression tests are automated.

Select TWO options.

Question #14 (1 point)
Which of the following is NOT a benefit of static testing?

a) Having less expensive defect management due to the ease of detecting defects later in the SDLC.
b) Fixing defects found during static testing is generally much less expensive than fixing defects found during dynamic testing.
c) Finding coding defects that might not have been found by only performing dynamic testing.
d) Detecting gaps and inconsistencies in requirements.

Select ONE option.

Question #15 (1 point)
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; i, iii, and v are not.
b) i, iii, and v are important; ii and iv are not.
c) i, ii, and v are important; iii and iv are not.
d) iii and iv are important; i, ii, and v are not.

Select ONE option.

Question #16 (1 point)
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 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.

Question #17 (1 point)
In your project, there has been a delay in the release of a brand-new application and test execution started late, but you have very detailed domain knowledge and good analytical skills. The full list of requirements has not yet been shared with the team, but management is asking for some test results to be presented.

Which test technique fits BEST in this situation?

a) Checklist-based testing.
b) Error guessing.
c) Exploratory testing.
d) Branch testing.

Select ONE option.

Question #18 (1 point)
Your team uses the three-point estimation technique to estimate the test effort for a new high-risk feature. The following estimates were made:

- Most optimistic estimation: 2 person-hours.
- Most likely estimation: 11 person-hours.
- Most pessimistic estimation: 14 person-hours.

What is the final estimate?

a) 9 person-hours.
b) 14 person-hours.
c) 11 person-hours.
d) 10 person-hours.

Select ONE option.

Question #19 (1 point)
Which of the following is NOT true for white-box testing?

a) During white-box testing, the entire software implementation is considered.
b) White-box coverage metrics can help identify additional tests to increase code coverage.
c) White-box test techniques can be used in static testing.
d) White-box testing can help identify gaps in requirements implementation.

Select ONE option.

Question #20 (1 point)
The reviews being used in your organization have the following attributes:

- There is the role of a scribe.
- The main purpose is to evaluate quality.
- The meeting is led by the author of the work product.
- There is individual preparation.
- A review report is produced.

Which of the following review types is MOST likely being used?

a) Informal review.
b) Walkthrough.
c) Technical review.
d) Inspection.

Select ONE option.

Question #21 (1 point)
You are testing a simplified apartment search form which has only two search criteria:

- Floor (with three possible options: ground floor; first floor; second or higher floor).
- Garden type (with three possible options: no garden; small garden; large garden).

Only apartments on the ground floor may have gardens. The form has a built-in validation mechanism that will not allow you to use the search criteria which violate this rule.

Each test has two input values: floor and garden type. You want to apply equivalence partitioning (EP) to cover each floor and each garden type in your tests.

What is the MINIMAL number of test cases to achieve 100% EP coverage?

a) 3
b) 4
c) 5
d) 6

Select ONE option.

Question #22 (1 point)
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.

Question #23 (1 point)
Which test activity does a data preparation tool support?

a) Test monitoring and control.
b) Test analysis and design.
c) Test implementation and execution.
d) Test completion.

Select ONE option.

Question #24 (1 point)
Your test suite achieved 100% statement coverage. What is the consequence of this fact?

a) Each instruction in the code that contains a defect has been executed at least once.
b) Any test suite containing more test cases than your test suite will also achieve 100% statement coverage.
c) Each path in the code has been executed at least once.
d) Every combination of input values has been tested at least once.

Select ONE option.

Question #25 (1 point)
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.

Question #26 (1 point)
Consider the following test categories (1–4) and agile testing quadrants (A–D):

1. Usability testing.
2. Component testing.
3. Functional testing.
4. Reliability testing.

A. Agile testing quadrant Q1: technology facing, supporting the development team.
B. Agile testing quadrant Q2: business facing, supporting the development team.
C. Agile testing quadrant Q3: business facing, critique the product.
D. Agile testing quadrant Q4: technology facing, critique the product.

How do the following test categories map onto the agile testing quadrants?

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

Select ONE option.

Question #27 (1 point)
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.

Question #28 (1 point)
Which of the following BEST describes the way acceptance criteria can be documented?

a) Performing retrospectives to determine the actual needs of the stakeholders regarding a given user story.
b) Using the given/when/then format to describe an example test condition related to a given user story.
c) Using verbal communication to reduce the risk of misunderstanding the acceptance criteria by others.
d) Documenting risks related to a given user story in a test plan to facilitate the risk-based testing of a given user story.

Select ONE option.

Question #29 (1 point)
Which of the following BEST describes the concept behind error guessing?

a) Error guessing involves using your knowledge and experience of defects found in the past and typical errors made by developers.
b) Error guessing involves using your personal experience of development and the errors you made as a developer.
c) Error guessing requires you to imagine that you are the user of the test object and to guess errors the user could make interacting with it.
d) Error guessing requires you to rapidly duplicate the development task to identify the sort of errors a developer might make.

Select ONE option.

Question #30 (1 point)
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) Report on achieved coverage.

Select TWO options.

Question #31 (1 point)
You are testing a mobile application that allows users to find a nearby restaurant based on the type of food they want to eat. Consider the following list of test cases, priorities (i.e., a smaller number means a higher priority), and dependencies:

Test case number	Test condition covered	Priority	Logical dependency
TC 001	Select type of food	3	None
TC 002	Select restaurant	2	TC 001
TC 003	Get direction	1	TC 002
TC 004	Call restaurant	2	TC 002
TC 005	Make reservation	3	TC 002

Which of the following test cases should be executed as the third one?

a) TC 003.
b) TC 005.
c) TC 002.
d) TC 001.

Select ONE option.

Question #32 (1 point)
You have been assigned as a tester to a team producing a new system incrementally. You have noticed that no changes have been made to the existing regression test cases for several iterations and no new regression defects were identified. Your manager is happy, but you are not. Which testing principle explains your skepticism?

a) Tests wear out.
b) Absence-of-errors fallacy.
c) Defects cluster together.
d) Exhaustive testing is impossible.

Select ONE option.

Question #33 (1 point)
During a risk analysis the following risk was identified and assessed:

- Risk: Response time is too long to generate a report.
- Risk likelihood, medium; risk impact, high.
- Response to risk:
  - An independent test team performs performance testing during system testing.
  - A selected sample of end users performs alpha and beta acceptance testing before the release.

What measure is proposed to be taken in response to this analyzed risk?

a) Risk acceptance.
b) Contingency plan.
c) Risk mitigation.
d) Risk transfer.

Select ONE option.

Question #34 (1 point)
Consider the following user story:

As an Editor
I want to review content before it is published
so that I can assure the grammar is correct

and its acceptance criteria:

- The user can log in to the content management system with "Editor" role.
- The editor can view existing content pages.
- The editor can edit the page content.
- The editor can add markup comments.
- The editor can save changes.
- The editor can reassign to the "content owner" role to make updates.

Which of the following is the BEST example of an ATDD test for this user story?

a) Test if the editor can save the document after deleting the page content.
b) Test if the content owner can log in and make updates to the content.
c) Test if the editor can schedule the edited content for publication.
d) Test if the editor can reassign to another editor to make updates.

Select ONE option.

Question #35 (1 point)
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.

Question #36 (1 point)
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.

Question #37 (1 point)
Which of these statements is NOT a factor that contributes to successful reviews?

a) Participants should dedicate adequate time for the review.
b) Splitting large work products into small parts to make the required effort less intense.
c) Participants should avoid behaviors that might indicate boredom, exasperation, or hostility to other participants.
d) Failures found should be acknowledged, appreciated, and handled objectively.

Select ONE option.

Question #38 (1 point)
Which of the following options shows an example of test activities that contribute to success?

a) Having testers involved during various software development life cycle (SDLC) activities will help 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 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.

Select ONE option.

Question #39 (1 point)
Which of the following statements describe a valid test objective?

a) To prove that there are no unfixed defects in the system under test.
b) To prove that there will be no failures after the implementation of the system into production.
c) To reduce the risk level of the test object and to build confidence in the quality level.
d) To verify that there are no untested combinations of inputs.

Select ONE option.

Question #40 (1 point)
Which of the following factors (i–v) have SIGNIFICANT influence on the test process?

i. The SDLC.
ii. The number of defects detected in previous projects.
iii. The identified product risks.
iv. New regulatory requirements forcing.
v. The number of certified testers in the organization.

a) i and ii have significant influence; iii, iv, and v have not.
b) i, iii, and iv have significant influence; ii and v have not.
c) ii, iv, and v have significant influence; i and iii have not.
d) iii and v have significant influence; i, ii, and iv have not.

Select ONE option.

Additional Sample Questions

ISTQB®'s general rule for publishing sample questions requires that at least one sample question be published for each learning objective. Thus, if there are more learning objectives than exam questions, questions covering learning objectives not included in the sample exam set are published separately as supplementary questions. This section contains the official supplementary questions.

Question #A1 (1 point)
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.

Question #A2 (1 point)
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.

L. Stapp et al., *ISTQB® Certified Tester Foundation Level*,
https://doi.org/10.1007/978-3-031-42767-1_13

Question #A3 (1 point)
A phone ringing in a neighboring cubicle distracts a programmer causing him to improperly program the logic that checks the upper boundary of an input variable. Later, during system testing, a tester notices that this input field accepts invalid input values.

Which of the following correctly describes an incorrectly coded upper bound?

a) The root cause.
b) A failure.
c) An error.
d) A defect.

Select ONE option.

Question #A4 (1 point)
Consider the following testware.

Test Charter #04.018 Session time: 1 h	
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.

Question #A5 (1 point)
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 in 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.