Certified Tester
Advanced Level
Agile Tester (CTAL-AT)
Sample Exam – Answers

v2.0

International Software Testing Qualifications Board

Copyright Notice

Copyright Notice © International Software Testing Qualifications Board (hereinafter called ISTQB®).

ISTQB® is a registered trademark of the International Software Testing Qualifications Board.

Copyright © 2026 (v2.0) the authors: Wim Decoutere, Michaël Pilaeten, Adam Roman, Tomas Tumasonis.

Copyright © 2014 (v1.0) the authors: Mette Bruhn-Pedersen, Debra Friedenberg, Jen Leger, Lloyd Roden, Lucjan Stapp, Patricia McQuaid, and the Foundation Agile Extension syllabus authors.

Revision History

Version	Date	Remarks
v1.0	2014/05/01	GA version published
v2.0	2026/04/17	GA version published

© International Software Testing Qualifications Board

Table of Contents

Copyright Notice 2

Revision History 3

Answer Key 6

Answers 7

Question #1 . . . 7
Question #2 . . . 8
Question #3 . . . 8
Question #4 . . . 9
Question #5 . . . 9
Question #6 . . . 9
Question #7 . . . 10
Question #8 . . . 10
Question #9 . . . 11
Question #10 . . . 11
Question #11 . . . 12
Question #12 . . . 12
Question #13 . . . 13
Question #14 . . . 13
Question #15 . . . 14
Question #16 . . . 14
Question #17 . . . 14
Question #18 . . . 15
Question #19 . . . 15
Question #20 . . . 15
Question #21 . . . 16
Question #22 . . . 16
Question #23 . . . 16
Question #24 . . . 17
Question #25 . . . 18
Question #26 . . . 19
Question #27 . . . 19
Question #28 . . . 20
Question #29 . . . 20
Question #30 . . . 21
Question #31 . . . 21
Question #32 . . . 22
Question #33 . . . 22
Question #34 . . . 23
Question #35 . . . 23
Question #36 . . . 24
Question #37 . . . 24
Question #38 . . . 25
Question #39 . . . 25

© International Software Testing Qualifications Board

Question #40 26

© International Software Testing Qualifications Board

Answer Key

Question Number (#)	Correct Answer	Learning Objective (LO)	K-Level	Number of Points
1	d	CTAL-AT-1.1.1	K2	1
2	d, e	CTAL-AT-1.1.1	K2	1
3	b, e	CTAL-AT-1.2.1	K2	1
4		CTAL-AT-1.3.1	K2	1
5	b	CTAL-AT-1.4.1	K2	1
6	b	CTAL-AT-2.1.1	K2	1
7	a, d	CTAL-AT-2.1.2	K2	1
8	b	CTAL-AT-2.1.3	K2	1
9	c	CTAL-AT-2.1.3	K2	1
10	c	CTAL-AT-2.2.1	K2	1
11	b, e	CTAL-AT-3.1.1	K2	1
12	d	CTAL-AT-3.1.2	K4	3
13	b	CTAL-AT-3.2.1	K2	1
14	a	CTAL-AT-3.2.1	K2	1
15	b	CTAL-AT-3.3.1	K2	1
16	d	CTAL-AT-3.4.1	K4	3
17	d	CTAL-AT-3.4.2	K2	1
18		CTAL-AT-4.1.1	K2	1
19	a	CTAL-AT-4.1.1	K2	1
20	b, d	CTAL-AT-4.1.2	K2	1
21	c	CTAL-AT-4.1.3	K2	1

Question Number (#)	Correct Answer	Learning Objective (LO)	K-Level	Number of Points
22		CTAL-AT-4.1.4	K2	1
23	d	CTAL-AT-4.1.4	K2	1
24	b	CTAL-AT-4.1.5	K3	2
25	d	CTAL-AT-4.1.5	K3	2
26	b	CTAL-AT-4.2.1	K2	1
27	b	CTAL-AT-5.0.0	K2	1
28	c	CTAL-AT-5.1.1	K2	1
29	b, d	CTAL-AT-5.1.2	K2	1
30	b	CTAL-AT-5.1.3	K2	1
31	a	CTAL-AT-5.1.4	K4	3
32	c	CTAL-AT-5.1.5	K3	2
33	a	CTAL-AT-5.1.5	K3	2
34	b, d	CTAL-AT-5.2.1	K2	1
35	b	CTAL-AT-5.2.2	K2	1
36	b	CTAL-AT-5.2.3	K2	1
37	b	CTAL-AT-5.3.1	K3	2
38	c	CTAL-AT-5.3.1	K3	2
39	c	CTAL-AT-6.1.1	K2	1
40	d	CTAL-AT-6.2.1	K2	1

© International Software Testing Qualifications Board

Answers

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
1	d	a) Incorrect. Basic non-functional testing (e.g., performance, usability, security) is performed during the iteration on new features. White-box testing (e.g., TDD) occurs during development, not only post-system testing. b) Incorrect. Black-box testing focuses mainly on verifying acceptance criteria during the iteration. White-box testing supports early defect detection (e.g., TDD) during coding. It is not primarily a post-iteration regression technique. c) Incorrect. Exploratory testing plays a critical role during the iteration. It complements structured black-box and white-box testing. It helps uncover unknown risks early. d) Correct. Functional testing primarily verifies that implemented features meet user story acceptance criteria and defined functional requirements. Once multiple features are completed, broader functional testing may occur, including integration testing across features (including those developed by other teams). This reflects that each iteration focuses on incremental functionality while broader validation occurs after the iteration, and often requires integrated environments.	CTAL-AT-1.1.1	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
2	d, e	a) Incorrect. Usability testing is non-functional, and can also be performed during the iteration. b) Incorrect. Security testing is non-functional, and can also be performed during the iteration. c) Incorrect. Performance testing is non-functional, and can also be performed during the iteration. d) Correct. Regression testing is an example of black-box testing performed after the iteration. e) Correct. End-to-end testing is an example of black-box testing performed after the iteration.	CTAL-AT-1.1.1	K2	1
3	b, e	a) Incorrect. Making end-to-end (E2E) testing the primary type is too costly and slow. b) Correct. End-to-end (E2E) testing is reserved for critical flows that provide the most value. c) Incorrect. End-to-end (E2E) testing is slow and provides limited diagnostic feedback. d) Incorrect. Regression testing is primarily addressed at lower test levels, not fully shifted to the higher test levels, which typically contain end-to-end (E2E) testing. e) Correct. End-to-end (E2E) testing is used in addition to unit tests and e.g. API tests, not as a substitute.	CTAL-AT-1.2.1	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
4		(1) Narrow focus and delayed discovery → A: formal testing often emphasizes predefined test results, which risks overlooking unexpected defects until late. (2) Subjective quality evaluation → B: holistic testing lacks fixed metrics, so stakeholder trust may be challenged. (3) Heavy emphasis on formal work products → A: formal testing relies on extensive documentation and formality, making adaptation harder. (4) Dependency on shared ownership and maturity → B: holistic testing requires a supportive culture; without it, adoption falters. Therefore, the correct grouping is: (1,3) → A and (2,4) → B.	CTAL-AT-1.3.1	K2	1
5	b	1 → B. Incremental regression testing provides rapid feedback by running subsets after code integrations. 2 → D. Risk-based regression testing identifies high-risk areas to focus regression testing effort. 3 → C. DevOps regression testing validates deployments with smoke tests and test monitoring in (pre-)production. 4 → A. Collaborative regression testing (e.g., bug bashes) adds value when test automation is incomplete, uncovering defects. b) is correct.	CTAL-AT-1.4.1	K2	1
6	b	a) Incorrect. Specialized expertise is still needed for complex or critical tasks. b) Correct. Generalizing specialists provide flexibility while retaining deep expertise, reducing bottlenecks and supporting collaboration. c) Incorrect. Agile software development promotes overlap and collaboration, not strict separation. d) Incorrect. Agile software development values adaptability over rigid role boundaries.	CTAL-AT-2.1.1	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
7	a, d	a) Correct. Dashboards that connect testing with business risks and readiness make quality visible and meaningful for Product Owners (PO). b) Incorrect. Verification during component testing is a developer responsibility, not motivating or aligned with the Product Owner's (PO) role. c) Incorrect. Excluding the Product Owner (PO) from acceptance criteria discussions reduces their engagement and weakens shared responsibility. d) Correct. Participating in collaborative test activities like bug bashes ties testing to user experience and business value, motivating Product Owners (PO). e) Incorrect. Assigning regression testing to Product Owners (PO) alone is neither efficient nor aligned with their role.	CTAL-AT-2.1.2	K2	1
8	b	a) Incorrect. Acceptance criteria should be clarified early, not after code is finalized. b) Correct. Continuous collaborative testing feedback gives developers confidence to refactor and correct defects earlier, lowering cost. c) Incorrect. Whole team testing removes silos; developers are expected to participate in testing, not avoid it. d) Incorrect. Business representatives help shape acceptance criteria and quality goals; excluding them contradicts the whole team approach.	CTAL-AT-2.1.3	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
9	c	a) Incorrect. This is an example of how Infrastructure as Code supports developers. b) Incorrect. This is an example of how version control systems support developers. c) Correct. Testers working closely with developers help clarify acceptance criteria through examples, leading to better alignment between the code and business expectations, which gives a clear, testable definition of "done" for user stories. d) Incorrect. This is an example of how test-driven development supports developers.	CTAL-AT-2.1.3	K2	1
10	c	a) Incorrect. Acceptance testing with actual users is a structured activity, not tissue testing. b) Incorrect. Formal usability test sessions in a usability lab go beyond the quick and informal purpose of tissue testing. c) Correct. Tissue testers are temporary, unbiased colleagues used early for quick feedback. d) Incorrect. Performance testing is unrelated to the informal nature of tissue testing.	CTAL-AT-2.2.1	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
11	b, e	a) Incorrect. Agile test planning avoids heavy documentation; it is adaptive and ongoing. b) Correct. Refining acceptance criteria collaboratively during backlog or iteration planning is a key Agile software development practice. c) Incorrect. Agile software development requires test environments and test data to be ready from day one of the iteration. d) Incorrect. Agile software development values both test automation (e.g., automated regression testing) and manual testing (e.g., exploratory, usability). e) Correct. Iteration planning includes selecting suitable test techniques for current user stories.	CTAL-AT-3.1.1	K2	1
12	d	a) Incorrect. Component testing is in Quadrant 1, which is technology-facing, not business-facing. b) Incorrect. Although UAT is in the correct quadrant (3), it is too early for acceptance testing. The product is in its early stage, undergoes rapid prototyping, and there is a high change frequency. c) Incorrect. Security testing is in Quadrant 4, which is technology, not business-facing. d) Correct. In early-stage products or systems undergoing rapid prototyping, strategy typically favors low-overhead, high-value feedback mechanisms. Exploratory testing from Quadrant 3 takes precedence over formal modeling or extensive coverage.	CTAL-AT-3.1.2	K4	3

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
13	b	a) Incorrect. Completion percentages are linked to phase-based test monitoring. b) Correct. Agile software development emphasizes lightweight, continuous feedback with charts, dashboards, and trend tracking. c) Incorrect. Real-time dashboards are a characteristic of Agile software development, where more sequential approaches favor periodic reporting. Also, sequential approaches typically do not have iterations. d) Incorrect. Metrics differ in purpose: Agile software development uses value-driven, actionable feedback instead of compliance measures.	CTAL-AT-3.2.1	K2	1
14	a	a) Correct. In Agile software development, test monitoring uses lightweight information such as burn-down charts to track test progress and product quality continuously. b) Incorrect. This example shows how retrospectives support test process improvement, not test monitoring. c) Incorrect. This example shows how iteration reviews support test control, not test monitoring. d) Incorrect. This example shows how iteration planning supports test planning, not test monitoring.	CTAL-AT-3.2.1	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
15	b	i) True. High code coverage does not guarantee meaningful testing or product quality. ii) True. Requirements coverage connects tests directly to user stories or acceptance criteria. iii) True. Exploratory testing coverage is reported with test charters or session-based test management. iv) False. Test type coverage is about balance across Agile testing quadrants, not acceptance criteria verification. v) False. This is a description of requirements coverage, not infrastructure coverage. Therefore, b is correct.	CTAL-AT-3.3.1	K2	1
16	d	a) Incorrect. Coverage is OK. There is no need to strengthen the test automation b) Incorrect. This is done when the failure rate is high, with low defect yield, which is not the case in this scenario. c) Incorrect. Test automation refactoring practices are helpful when the number of flaky tests is high, but this metric is at the expected level. d) Correct. High cycle time for resolving defects often points to weak collaboration across roles. A possible improvement is to adopt a whole-team approach to defect triage.	CTAL-AT-3.4.1	K4	3
17	d	1 → C: Test-focused retrospectives identify testing challenges and build shared ownership. 2 → A: Self-assessments help establish a baseline and identify gaps. 3 → D: Risk workshops ensure alignment with business risks and product risks. 4 → B: Test-first approaches (ATDD/BDD) strengthen feedback loops. Correct option: d.	CTAL-AT-3.4.2	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
18		Automated acceptance tests belong to category A because they act as executable specifications of behavior. Test charters belong to category B because they guide exploratory testing based on emerging risks or questions. Visual models belong to category A because diagrams define flows or states that act as reference points for development. Exploratory test session sheets belong to category B because they document observations that refine requirements as the Agile software development team learns. Therefore: 1 and 3 → A; 2 and 4 → B.	CTAL-AT-4.1.1	K2	1
19	a	a) Correct. Examples are derived from conversations between the team and business stakeholders, and they are often documented in a format that can be executed as tests. b) Incorrect. Test charters are used in exploratory testing which is typically manual, not automated. c) Incorrect. Build scripts are used to build the release, not to test it. d) Incorrect. Defect reports are not a basis for automated testing.	CTAL-AT-4.1.1	K2	1
20	b, d	a) Incorrect. Storyboarding and testboarding promote early collaboration, not late involvement. b) Correct. Storyboarding reveals gaps, risks, and assumptions. c) Incorrect. Storyboarding and testboarding are iterative and adapt to scope changes. d) Correct. Testboarding maps scenarios, aligning testers, developers, and product owners. e) Incorrect. Regression testing automation is not the purpose of storyboarding and testboarding.	CTAL-AT-4.1.2	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
21	c	a) Incorrect. Example mapping does not require detailed test scripts; it fosters shared discovery instead. b) Incorrect. Example mapping goes beyond functional requirements by surfacing hidden assumptions and conflicts. c) Correct. Concrete examples help align understanding, clarify acceptance criteria, and strengthen the test basis. d) Incorrect. Example mapping is used during refinement to evolve acceptance criteria, not to finalize them.	CTAL-AT-4.1.3	K2	1
22		Confirmation bias → Examples 1 and 4. Designing only happy-path tests, verifying expected results. Anchoring effect → Examples 2 and 3. Fixation on early requirements, sticking to dominant interpretations.	CTAL-AT-4.1.4	K2	1
23	d	a) Incorrect. This is an example of an anchoring effect, not a confirmation bias. b) Incorrect. This is an example of a conformity bias, not a confirmation bias. c) Incorrect. This is an example of a conformity bias, not a confirmation bias. d) Correct. Confirmation bias may lead to designing and executing tests that only verify expected behavior, neglecting potential failure paths or unusual conditions.	CTAL-AT-4.1.4	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
24	b	a) Incorrect. Statement iv contradicts the syllabus guidance that slicing by scenario should start with the most common scenario and only later cover rare or complex cases. b) Correct. Statement i reflects workflow slicing where each step produces observable behavior with clear acceptance criteria. Statement ii reflects slicing by data complexity where a simple happy path is created first. Statement iii reflects slicing by interface where backend logic is validated before UI workflows. c) Incorrect. Statement v overgeneralizes vertical slicing. The syllabus states vertical slices are useful but only where appropriate, and if end-to-end scope becomes unmanageable, Agile software development teams should revert to slicing by scenario or workflow slicing. d) Incorrect. Statement iv is wrong because rare exceptions should not be the starting point, and statement v is wrong because vertical slicing is not required for every user story.	CTAL-AT-4.1.5	K3	2

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
25	d	a) Incorrect. Slicing by data complexity begins with a single happy path example using the simplest data and defers alternative data conditions or edge cases to separate user stories. The two sliced user stories do not use any alternative data conditions or edge cases. b) Incorrect. Slicing based on constraints focuses first on general assumptions and then considers conditions derived from known constraints. The scenario provides no constraints. c) Incorrect. Slicing by interface separates backend processing logic from UI concerns when they can be tested independently. The two sliced user stories do not separate backend logic from UI. d) Correct. Vertical slicing ensures that each user story includes a complete thin slice of functionality across layers, from database to UI, where appropriate. First sliced user story focuses on the search and product detail workflow across backend (search API) and frontend (product detail page). Second one focuses on the add-to-cart functionality across backend (cart API) and frontend (cart UI).	CTAL-AT-4.1.5	K3	2

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
26	b	a) Incorrect. User stories are described in the syllabus as free-form text capturing user value, not state transition diagrams. BPMN is noted for process modeling, not narrative description. b) Correct. User stories capture user value in a concise narrative (B). BPMN shows structured workflows for process consistency (C). State transition diagrams visualize system behavior under different conditions (D). Glossaries ensure consistent terminology in acceptance criteria (A). c) Incorrect. Glossaries are not process models; they are explicitly mentioned as tools to ensure shared understanding of domain terms. User stories are not structured workflows, and state transition diagrams are not narrative descriptions. d) Incorrect. User stories do not serve as glossaries. BPMN diagrams are for workflows, not state transitions. State transition diagrams, not BPMN, model conditional system behavior.	CTAL-AT-4.2.1	K2	1
27	b	a) Incorrect. Test heuristic is a generally recognized rule of thumb that helps to achieve a test objective. b) Correct. Test charter is a documentation of the test objective for a test session. It usually covers test mission, focus areas, resources and timebox. c) Incorrect. Test tour is a set of exploratory tests organized around a special focus. This document is not a set of tests, but an example of a test charter, which is a basis for designing such tests. d) Incorrect. Test mnemonic is a memory aid in the form of an abbreviation, used to help testers remember key concepts or test practices.	CTAL-AT-5.0.0	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
28	c	a) Incorrect. This option mismatches checklists and guidelines. According to the syllabus, guidelines are general principles such as "test early," while checklists are structured prompts. b) Incorrect. This option incorrectly places guidelines as rules of thumb. The syllabus defines rules of thumb as experiential insights, such as defects clustering in components, not general guiding principles. c) Correct. Guidelines are broad principles like "test early." Generic checklists are structured lists such as usability or security risk reminders. Rules of thumb include insights like "if it broke before, it will probably break again." Analogies and metaphors include test tours that simulate familiar scenarios such as traveling through a city. d) Incorrect. This option incorrectly associates guidelines with checklists. Checklists are reusable prompts across areas such as quality characteristics, not broad general principles.	CTAL-AT-5.1.1	K2	1
29	b, d	a) Incorrect. Reporting a defect is documentation, not test mnemonic usage. b) Correct. Choosing regression testing focus areas (e.g., recent, core, risky) is an example of applying a regression test mnemonic RCRCRC. c) Incorrect. Test session sheets from an exploratory testing session are outputs of testing, not test mnemonic-based activities. d) Correct. Exploring system dimensions like data and interfaces reflects the use of a system-analysis test mnemonic SFDIPOT. e) Incorrect. Automated tests are regression testing assets, not test mnemonic usage.	CTAL-AT-5.1.2	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
30	b	a) Incorrect. Business district relates to core features most relied upon by customers. b) Correct. Tourist district relates to functions attractive to novice users but less useful for experts. c) Incorrect. Historical district relates to legacy code and defect-prone functions. d) Incorrect. Areas related to negative testing and vulnerabilities.	CTAL-AT-5.1.3	K2	1
31	a	a) Correct. This test charter correctly reflects the user story and all acceptance criteria: it focuses on the online shopper, the wishlist functionality (add/remove), and considers session persistence (logged-in/logged-out scenarios). b) Incorrect. This test charter focuses on checkout/payment, not wishlist functionality. c) Incorrect. This test charter focuses on inventory/stock, which is outside the scope of the wishlist story. d) Incorrect. This test charter focuses on admin tasks, not the user-facing wishlist behavior.	CTAL-AT-5.1.4	K4	3

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
32	c	a) Incorrect. The charter explicitly targets first-time users and password reset via email. Password change and password reset are different workflows with different risks and UI paths. b) Incorrect. This test focuses on failed login attempts, account lockout behavior, and lockout messaging. While related to authentication, it does not test password reset via email or the first-time reset experience described in the charter. This is closer to a security or login-protection test, not a password-reset workflow exploration. c) Correct. This scenario directly exercises the end-to-end password-reset flow described in the charter. d) Incorrect. This option tests system performance, not exploratory workflow behavior.	CTAL-AT-5.1.5	K3	2
33	a	a) Correct. This exploratory testing session follows the test charter exactly. b) Incorrect. This exploratory testing session does not follow the test charter – it is similar, but has different focus (payment method verification, not the cart features verification). c) Incorrect. This exploratory testing session does not follow the test charter – it is about filtering and sorting, not about the cart features. d) Incorrect. This exploratory testing session follows the test charter goal, but it does not follow the timebox (the session should last 45 minutes, but in fact it lasts 1 hour).	CTAL-AT-5.1.5	K3	2

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
34	b, d	a) Incorrect. According to the syllabus, the Driver's role is to act as the typist and perform "no thinking." The Driver does not make decisions — this responsibility belongs to the Navigator. b) Correct. The syllabus defines the Navigator as the "main tester," who considers insights from the mob but has the final decision on what action to take. c) Incorrect. The syllabus specifies that the Moderator "sits in the back and does not rotate." They ensure the mob testing approach is followed and may pause the test session if needed, but they do not type. d) Correct. After a mob testing session, the Moderator collects feedback on sticky notes, groups them on a whiteboard, and helps the Agile software development team identify patterns for continuous improvement. This retrospective is an essential part of mob testing. e) Incorrect. The syllabus emphasizes that mob testing is a collaborative test approach where everyone works together on the same system at the same time. Working individually contradicts the concept of swarming and mobbing.	CTAL-AT-5.2.1	K2	1
35	b	a) Incorrect. Pair testing does not involve splitting work; both testers focus on the same task at the same time. b) Correct. Pair testing improves detection and coverage because two testers bring different perspectives and react to findings instantly. c) Incorrect. Pair testing is not a review step led by senior staff; it is a collaborative activity between partners. d) Incorrect. Pair testing often enhances exploratory work and does not replace it with scripted test steps.	CTAL-AT-5.2.2	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
36	b	a) Incorrect. Vibe testing is not about creating detailed scripted tests in advance; it emphasizes intent-first validation rather than predefined test scripts. b) Correct. Vibe testing ensures the system aligns with what users intend, not just what the AI-generated code outputs. c) Incorrect. Code reviews may still occur, but vibe testing focuses on intent-first behavior validation, not line-by-line inspection. d) Incorrect. Automated unit tests can help but do not capture the intent-driven essence of vibe testing.	CTAL-AT-5.2.3	K2	1
37	b	a) Incorrect. Combining setup and verification into one test step does not address the missing dependency documentation. b) Correct. This is a “Hidden Dependencies” smell; the fix is to make dependencies explicit in the preconditions. c) Incorrect. Adding test steps to test the database record creates unnecessary complexity instead of clarifying dependencies. d) Incorrect. Merging test cases introduces interdependence (“Interdependent Tests”), not a solution to hidden dependencies.	CTAL-AT-5.3.1	K3	2

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
38	c	a) Incorrect. One More Step smell means that expected results contain more test steps (e.g., using words such as "check", "verify", "see"). This is not the case, since each test case has an atomic test step for the expected results. b) Incorrect. Call on Me smell means extensive calling of other test cases or parameters. This is not the case, because none of the two test cases call any other. c) Correct. Interdependent Tests smell means that one test's success depends on another having run first (or having left data behind). The second test case's success depends on the first one, otherwise the user database would be empty and the second test case couldn't be successfully run. d) Incorrect. Hotstepper smell means separate test step for every action where an action has no effect. This is not the case, because all actions described in the test steps have effects and can be verified.	CTAL-AT-5.3.1	K3	2
39	c	a) Incorrect. Exploratory testing depends on human insight and emergent behavior, making it unsuitable for automation. b) Incorrect. Although the statement sounds plausible (system coverage argument), it reverses the pyramid strategy. c) Correct. This option correctly distinguishes the automation strategy used in Agile and described in the syllabus. d) Incorrect. Regression automation may evolve later, but automation is not primarily introduced after the iteration, and not all tests should be automated.	CTAL-AT-6.1.1	K2	1

© International Software Testing Qualifications Board

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
40	d	a) Incorrect. Exploratory tools provide test session sheets but not production usage data. b) Incorrect. Task boards show planned work but not actual customer behavior. c) Incorrect. Defect management tools focus on internal quality defects, not customer usage. d) Correct. Monitoring and analytics tools capture production-level metrics such as usage patterns, defect rates, and performance data, which help refine backlog priorities.	CTAL-AT-6.2.1	K2	1

© International Software Testing Qualifications Board