Sample Exam – Answers

Sample Exam set B

Version 1.0

ISTQB® Test Management Syllabus Advanced Level

Compatible with Syllabus version 3.0

International Software Testing Qualifications Board

Copyright Notice

Document Responsibility

The ISTQB® Examination Working Group is responsible for this document.

This document is maintained by a core team from ISTQB® consisting of the Syllabus Working Group and Exam Working Group.

Acknowledgements

This document was produced by a core team from ISTQB®: Horst Pohlmann (Product Owner, Vice Chair ALWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma, Anna Menkova and Miroslav Renda.

This exam paper was produced by Ecaterina Irina Manole and Anna Menkova supported by Horst Pohlmann.

Following reviewers participated in the internal EWG BETA review: Lucjan Stapp (CaSTB), Jean-Baptiste Crouigneau (EWG-Rep in TF-TM), Andrea Horvath, Tal Peer, Klaus Skafte, Michael Stahl and Reifa Tangon.

The core team thanks the Exam Working Group review team, the Syllabus Working Group and Member Board for their suggestions and input.

Revision History

Sample Exam – Answers Layout Template used:	Version 2.11 Date: October 16, 2023

Version	Date	Remarks
1.0	May 30, 2025	Approval for internal EWG BETA REVIEW [NOT published]
1.0	Sept. 15, 2025	After Internal EWG BETA Review and Rework [NOT published]
1.0	Dec 10, 2025	Post Rework after ISTQB Member Board Review [NOT publisded]
1.0	March 28, 2026	Compliance Check to TM 3.0.0 Syllabus incl. corrections and Appendix inserted to be ISTQB rules compliant. [NOT published]

Table of Contents

Introduction

Purpose of this document

The example questions and answers and associated justifications in this sample exam have been created by a team of subject matter experts and experienced question writers with the aim of:

- Assisting ISTQB® Member Boards and Exam Boards in their question writing activities
- Providing training providers and exam candidates with examples of exam questions

These questions cannot be used as-is in any official examination.

Note that real exams may include a wide variety of questions, and this sample exam ***is not*** intended to include examples of all possible question types, styles, or lengths, also this sample exam may both be more difficult or less difficult than any official exam.

Instructions

In this document you may find:

- Answer Key table, including for each correct answer:
  - K-level, Learning Objective, and Point value.
- Answer sets, including for all questions:
  - Correct answer
  - Justification for each response (answer) option
  - K-level, Learning Objective, and Point value.
- Additional answer sets, including for all questions [does not apply to all sample exams]:
  - Correct answer
  - Justification for each response (answer) option
  - K-level, Learning Objective, and Point value.

- *Questions are contained in a separate document.*
- The main part covers a full Sample exam paper acc. to CTAL-TM v3.0 Structure and Rules.

Answer Key

Question Number (#)	Correct Answer	LO	K-Level	Points
1	c	TM-1.1.1	K2	1
2	b	TM-1.1.2	K2	1
3	c	TM-1.2.1	K2	1
4	c	TM-1.2.2	K2	1
5	a	TM-1.2.3	K2	1
6	c	TM-1.2.5	K2	1
7	a, d	TM-1.2.7	K4	3
8	c	TM-1.2.7	K4	3
9	c	TM-1.3.1	K2	1
10	c	TM-1.3.2	K2	1
11	d	TM-1.3.4	K4	3
12	a	TM-1.3.4	K4	3
13	a	TM-1.3.5	K2	1
14	b	TM-1.4.1	K2	1
15	a	TM-1.4.2	K4	3
16	a	TM-1.4.2	K4	3
17	b	TM-1.4.3	K3	2
18	a	TM-1.4.3	K3	2
19	b	TM-1.5.1	K2	1
20	a	TM-1.5.3	K2	1
21	a, c	TM-1.5.4	K3	2
22	a	TM-1.5.4	K3	2
23	b	TM-1.6.1	K2	1
24	b, c	TM-1.6.3	K4	3
25	c	TM-1.6.3	K4	3
26	c	TM-1.6.4	K2	1
27	b	TM-2.1.1	K2	1
28	a	TM-2.1.2	K2	1
29	b, c	TM-2.1.3	K4	3
30	c	TM-2.1.3	K4	3
31	c	TM-2.2.2	K2	1
32	b, d	TM-2.2.3	K4	3
33	c	TM-2.2.3	K4	3
34	c	TM-2.3.1	K3	2
35	b	TM-2.3.1	K3	2
36	b	TM-2.3.2	K2	1
37	a	TM-2.3.3	K2	1
38	a	TM-2.3.4	K2	1
39	c	TM-2.3.5	K3	2
40	a, e	TM-2.3.5	K3	2
41	c	TM-2.3.6	K2	1
42	c	TM-3.1.1	K2	1
43	b, d	TM-3.1.2	K4	3
44	a	TM-3.1.2	K4	3
45	d	TM-3.1.3	K2	1
46	c	TM-3.1.4	K2	1
47	b	TM-3.1.5	K2	1
48	b	TM-3.2.1	K2	1
49	b	TM-3.2.2	K3	2
50	a, c	TM-3.2.2	K3	2

Answers

Question Number (#)	Correct Answer	Explanation / Rationale	Learning Objective (LO)	K-Level	Number of Points
Section: Test Process					
1	c	a) Is not correct. This falls under release test planning resource allocation, which is a core activity during release test planning. Assigning testers to specific responsibilities is essential to ensure test execution is feasible and efficient. Reference: Section 1.1.1 describes test planning as including activities such as “identifying the activities and resources required to achieve the test objectives” (which includes allocating test staff to planned test activities). b) Is not correct. Estimating infrastructure or tool-related costs is part of test planning. The release plan must address resource allocation, including environments and the associated effort/cost drivers needed for budgeting and feasibility. Reference: Section 1.1.1 states that test planning includes “estimating the required test resources such as test staff, test tools, test environments, and test data” (and therefore also supports estimating the cost impact of these resources). c) Is correct. Test policy is a high-level organizational artifact that sets overall principles and goals for testing. It is not created or documented during release planning; rather, release-level plans are based on an existing test policy defined at organizational level, not per release. Syllabus (Section 1.1.1 on test planning) describes how test plans are derived from existing higher-level guidance (such as organizational objectives, strategies, and constraints) and focus on how testing will be performed in a given project or release, not on defining that higher-level guidance. Therefore, creating and documenting the test policy is not a valid release planning activity. d) Is not correct. Identifying and analyzing risks during backlog refinement or release planning helps ensure test priorities and coverage align with business goals and supports risk-based decisions for the release scope. Reference: Section 1.1.1 notes that risk analysis in test planning “involves identifying and assessing the potential impact and likelihood of risks as part of test planning.”	TM-1.1.1	K2	1

2	b	a) Is not correct. This describes test execution and result evaluation, not the purpose of test monitoring. While individual test case outcomes are part of progress data, monitoring focuses on collecting, recording, and comparing actual progress/results to the plan and identifying deviations, rather than reviewing pass/fail in isolation. "Test monitoring involves collecting and recording test results, identifying deviations from planned testing…" (Section 1.1.2) b) Is correct. This is a contextually appropriate monitoring activity: measuring actual progress against the planned schedule and communicating deviations early—especially when critical compliance milestones are at risk. It directly supports control and timely decision-making under pressure. "Test monitoring involves collecting and recording test results, identifying deviations from planned testing…" (Section 1.1.2) c) Is not correct. Postponing monitoring undermines its purpose. Test monitoring should begin as soon as testing starts (and continue throughout) to detect deviations early and mitigate potential compliance and schedule impacts. d) Is not correct. Delaying reporting until the end of the sprint contradicts the goal of early deviation detection. In time-sensitive, compliance-driven projects, monitoring information must be reported/communicated promptly so corrective actions can be taken in time.	TM-1.1.2	K2	1

Section: Context of Testing					
3	c	a) Is not correct. These are primarily coordination and process-support roles. While they help organize work and enable execution, they typically do not own business priorities and therefore are not the primary stakeholders for early risk discussions that define test objectives and required coverage. b) Is not correct. These stakeholders contribute to documentation, tooling support, and operational enablement, but they generally do not influence business priorities or acceptance expectations. As a result, their input is less relevant when the goal is to define risk-based test coverage early. c) Is correct. Product owners, project managers, and business users are the most relevant stakeholders for early test planning and risk discussions because they: • Represent business priorities and customer value, ensuring testing focuses on what matters most to end users and stakeholders. • Provide insight into business and domain risks (e.g., regulatory, operational, reputational) that strongly influence test scope and coverage. • Help define quality expectations and acceptance criteria, which drive test objectives and coverage needs. • Can make or influence prioritization decisions, enabling the test manager to align the test strategy and coverage with perceived risks. This makes them the best choice for early involvement, unlike roles focused mainly on coordination or implementation. d) Is not correct. These are implementation-focused roles (building, deploying, and executing technical work). They are valuable for feasibility and execution planning, but they are not the primary decision-makers for business-risk prioritization and therefore are not the most appropriate group to define required test coverage based on perceived business risks early on.	TM-1.2.1	K2	1

4	c	a) Is not correct. While software developers may interact with the tool and may be involved in technical rollout/integration, they are often not the primary high-influence decision makers for tool selection in a regulated healthcare project. Tool approval typically requires stakeholders who influence compliance, traceability, and integration strategy. Developers' influence on these final selection/approval decisions can vary by organization, but is often lower than that of Product Owners or Test Leads/QA Managers, who must ensure regulatory alignment and drive adoption within the test organization. b) Is not correct. Quality Assurance Auditors are critical for compliance and may provide input on compliance criteria, but they typically focus on auditing and independent assessment, not on driving initial tool selection or being involved in implementing the tool. Their influence can be high in compliance validation, but their day-to-day involvement in tool rollout and usage enablement is typically lower during selection. c) Is correct. Test Leads/QA Managers have both high influence and high interest in selecting a test execution tool for a regulated healthcare application because they are responsible for ensuring that testing supports compliance, traceability, and product risk mitigation. These responsibilities make them likely to be mapped to the Promoter quadrant described in TM-1.2.2, which states that high-influence, high-interest stakeholders are critical in shaping test strategy and tool decisions. Test Leads/QA Managers typically have high influence and high involvement in implementing the tool (process integration, rollout planning, training, governance, and ensuring it is effectively used in system and integration testing). Together, these roles form the group with the most influence and involvement in tool selection and implementation. Syllabus reference: "Promoters (High Influence, High Interest) are vital for shaping the test strategy and plan." (TM-1.2.2 Importance of Stakeholders' Knowledge) d) Is not correct. Marketing and Sales teams may have high interest in the product's commercial success, but they have low influence on technical tool decisions and typically low involvement in implementing a test execution tool, so they are often mapped as lower-influence stakeholders for this type of decision, even if their interest is high.	TM-1.2.2	K2	1

5	a	a) Is correct. Providing workshops aligns with syllabus guidance: test managers should support the team in combining structured processes with Agile flexibility and enhance their collaboration capabilities in hybrid environments (e.g., helping the team synchronize continuous testing in sprints with the formal documentation/traceability needs of traditional phases). Reference: Syllabus, Section 1.2.3 ("Ensuring the team is adept at combining structured processes with Agile flexibility" and "Enhancing collaboration between the test team and stakeholders to better manage testing within sprints and traditional test phases "). b) Is not correct. Simply extending deadlines without training does not ensure skill development or successful integration of Agile practices and does not actively build the team's capability to work effectively in a hybrid model. c) Is not correct. Delaying Agile adoption goes against the purpose of a hybrid model, which is to blend traditional and Agile elements to ease transition, not to postpone it (hybrid models are used to ease the transition by combining structure with Agile flexibility). d) Is not correct. While external Agile testers could, in principle, provide examples or informal learning, the option explicitly states that they "temporarily manage the continuous testing work while the current team focuses on traditional phases." This setup sidelines the existing team from practicing continuous testing within the hybrid model and delays their adaptation. According to Section 1.2.3, test management in a hybrid model should develop the team's capability to combine traditional and Agile practices and improve collaboration across sprint-based and traditional phases, not outsource the Agile part while the team remains in its old way of working.	TM-1.2.3	K2	1

6	c	a) Is not correct. Ensuring the scope and objectives of system integration testing are clear and attuned to risk assessment and quality targets is explicitly defined as a test management activity at the system integration test level (see Syllabus, Section 1.2.5) b) Is not correct. Coordinating the availability of test environments and test interfaces is a core system integration test management responsibility, as the test manager must ensure that the technical prerequisites for integration testing are available and aligned with the integration sequence to support maintaining oversight of progress, outcomes, and issue management during system integration testing. c) Is correct. This relates to product validation and business readiness, which belong to the acceptance level, not integration. (see Syllabus, Section 1.2.5) d) Is not correct. Defining which test techniques will be used to verify interfaces at the system integration test level is part of test planning and management at the integration level, since the test manager is responsible for selecting and aligning test techniques with the test objectives of that test level as part of defining the test approach (including the selection and combination of test techniques).	TM-1.2.5	K2	1

7	a, d	a) Is correct. This option aligns with key test management principles: Risk-based testing prioritizes testing where quality risks are highest optimized resource allocation ensures efficient team and tool usage monitoring execution metrics provides insights for better control. Reference: "Test monitoring should include risk monitoring. In addition to monitoring the evolution of known quality risks, it should include analyzing any new quality risks and adjusting the risk register." (Section 1.3.1). In addition, test planning includes estimating/allocating test resources and test monitoring/control requires a monitoring framework with measures and targets (Sections 1.1.1 and 1.1.2). This option is the best because it directly addresses risk, resource management, and monitoring. b) Is not correct. This option is incorrect because it focuses on test quantity rather than risk-based prioritization and control. Simply increasing exploratory testing and executing more test cases does not resolve resource allocation, risk management, or monitoring issues. While increasing exploratory testing can add value, simply executing more test cases does not resolve the identified issues in the scenario (unanticipated risks, insufficient resource planning, inadequate monitoring/test control). Risk-based testing is more effective than focusing solely on test volume. Monitoring and control are key missing elements. References: " Risk Assessment and Mitigation Plan: Integral to the test plan is a robust risk management framework. Test managers must undertake a detailed risk analysis, pinpointing potential vulnerabilities and challenges that could impact both the project workflow and the end product." (Section 1.2.7) c) Is not correct. This option is incorrect because automation alone does not ensure risk mitigation or effective test monitoring and test control. Automation without monitoring and control is insufficient. Agile testing requires continuous monitoring of test automation outcomes. The syllabus explicitly emphasizes continuous monitoring and integration of tools/environments within CI/CD and ongoing monitoring/control activities. References: "Continuous monitoring ensures that test tools and environments are effectively integrated within the CI/CD pipeline, facilitating continuous testing and immediate feedback loops." (Section 1.2.7) d) Is correct. This option is correct because continuous monitoring and adaptive control improve test effectiveness and Agile adaptability. Continuous test monitoring ensures proactive identification of emerging risks and defects. In addition, test control uses the information from test monitoring to provide guidance and corrective actions and revisits test planning activities when necessary, which directly supports adapting the test process based on project insights. Aligns with Agile methodology where testing needs to evolve rapidly. Reference: "Continuous monitoring ensures that test tools and environments are effectively integrated within the CI/CD pipeline, facilitating continuous testing and immediate feedback loops that are vital for the Agile development process." (Section 1.2.7) and 1.1.2 / 2.1.2 for „corrective actions"/„revisits planning". e) Is not correct. This option is incorrect because regression testing alone does not address risk, and test monitoring and control are key test management activities that	TM-1.2.7	K4	3

		require a monitoring framework and ongoing oversight; development collaboration is important but does not replace test management monitoring/control. Reference: The test management role "includes… planning the test activities, monitoring and controlling the test progress…" (Section 0.11). In addition, the syllabus highlights test monitoring and control as a key test management activity (Section 1.1) and describes test monitoring/control as ongoing activities supported by a monitoring framework (Section 1.1.2).			
8	c	a) Is not correct. Automation and defect management are necessary parts of the project's workflow, but they are not the primary focus compared to risk-based planning/monitoring/control emphasized by functional safety and cybersecurity standards. b) Is not correct. Configuration and usability concerns are valid, especially in distributed teams, but this combination does not address the project's primary test management focus: risk mitigation in a safety-critical system. c) Is correct. The system is safety-critical, must comply with functional safety and cybersecurity standards, and is entering the initial maintenance phase with frequent updates and thorough regression testing. In this context, test management must focus on continuous risk identification and control (for new changes and for regression) and on tracking progress against risk mitigation priorities over multiple iterations. A risk-based test plan that is maintained throughout the maintenance phase, and regression and progress tracking aligned with those risks, directly supports ongoing compliance, traceability, and control in a regulated environment using CI/CD and continuous monitoring. d) Is not correct. Exploratory testing and retrospectives are useful, but they do not sufficiently address the test planning and monitoring needs for continuous compliance in regulated environments like this one.	TM-1.2.7	K4	3

Section: Risk-based testing					
9	c	a) Is not correct. Overestimating all non-functional risks defeats the purpose of risk prioritization and leads to resource inefficiency. b) Is not correct. Too rigid; risk-based testing does not require always testing everything, but rather prioritizing based on current risk levels. c) Is correct. Risk-based testing involves identifying, assessing, and responding to risks. In this scenario, performance risk was deprioritized based only on low likelihood, even though the impact was clearly high, which led to business loss. Proper risk prioritization must always consider the combined risk exposure, not likelihood alone (Section 1.3.3 and 1.3.1). d) Is not correct. Waiting until the end removes the value of early risk mitigation, which is central to risk-based testing.	TM-1.3.1	K2	1
10	c	a) Is not correct. Expert interviews involve one-on-one sessions gathering individual expert input, not collaborative group work in a structured setting. b) Is not correct. Brainstorming is a technique for generating ideas and can be used to collect potential risks. However, the scenario describes a structured, facilitated group session with multiple stakeholders to jointly identify and categorize product risks. This best matches a risk workshop; brainstorming may be used within such a workshop, but it is not the most appropriate description here. Reference: Syllabus, Section 1.3.2. c) Is correct. Organizing a structured, collaborative session with multiple stakeholders to jointly identify and categorize product risks matches the definition of a risk workshop. Reference: Syllabus, Section 1.3.2. d) Is not correct. Referring to past experience typically involves looking at historical data or lessons learned, not actively working in a group to assess current product risks.	TM-1.3.2	K2	1

11	d	a) Is not correct. Because it *accepts Risk B* despite its high exposure and compliance relevance. In addition, for Risk A the mitigation should explicitly include timing/latency acceptance criteria (SLA-style checks), which is not made explicit in a) (Syllabus, section 1.3.4). Also, given the limited access to production-like environments, broad performance and load testing as stated for Risk D is unlikely to be feasible or effective, whereas a targeted approach is more appropriate under these constraints. b) Is not correct. Stakeholder workshops for Risk A do not replace technical system validation. Risk B (external data corruption) requires integration/system-level validation, not only unit testing. Risk D (€400k exposure) is incorrectly deferred, despite being higher priority than Risk C. This option misaligns risk exposure to test level selection. c) Is not correct. Exploratory testing for data corruption (Risk B) is insufficient for integrity validation. Risk C (€160k exposure) is the lowest priority risk, yet receives early testing focus. Risk D (€400k) is unjustifiably deprioritized. This violates risk-based prioritization and rigor selection. d) Is correct. Risk A (€900k) is correctly mitigated via end-to-end validation with timing. Risk B (€600k) is correctly mitigated using robustness and data integrity testing. Risk C (€160k) is correctly accepted as residual due to its lower business impact. Risk D (€400k) is correctly addressed using targeted peak-load tests. This is the only option that correctly aligns risk exposure, test technique rigor, system constraints and regulatory demands in line with risk-based testing principles (Syllabus, section 1.3.4).	TM-1.3.4	K4	3

12	a	a) Is correct. This option directly applies the risk profiles provided in the stem. Payment Processing (High–High–High): requires the most skilled testers and the most rigorous techniques. User Authentication (High security): needs focused security testing and review. Shopping Cart (High performance): requires performance-focused effort. Recommendations (Low overall risk): appropriate to assign lighter regression. It correctly demonstrates proportional allocation of techniques, effort, and tester expertise based on specific risk levels, which is the essence of risk-based testing. b) Is not correct. Basing prioritization mainly on past defects rather than the explicit risk levels in the table contradicts risk-based testing. In a high-risk environment (e.g., Payment Processing with High–High–High risk), waiting for defects to appear before increasing test depth can lead to severe impact. Applying the same regression depth to all modules ignores the known differences in security, performance, and functionality risk. Assigning testers for workload balance rather than according to risk further misaligns testing effort with the areas needing the most mitigation. c) Is not correct. Focusing on visible UI changes disregards the risk table. Reusing the same regression suite across modules fails to account for differing security, performance, and functionality risks, especially the High–High–High risk of Payment Processing or the High security risk of User Authentication. Tester assignment based on deadlines instead of risk further misaligns priorities. d) Is not correct.	TM-1.3.4	K4	3

		Beginning with broad exploratory sessions and postponing risk-based decisions ignores the explicit, known risk differences across modules. The risk table already indicates where security, performance, and functional risks are high, so delaying prioritization prevents proper application of risk-based planning. This does not demonstrate mitigation aligned to the provided risk levels.			
13	a	a) Is correct. It captures the essential distinction: heavyweight risk-based techniques (e.g., FMEA, Hazard Analysis) are more formal and more thorough, typically involving more extensive documentation and broader stakeholder involvement. In contrast, lightweight techniques are less formal and lower-effort, often relying on faster stakeholder input and reduced documentation/analysis depth, which fits Agile projects with short iterations and evolving, incrementally captured requirements. b) Is not correct. It overgeneralizes. Lightweight approaches are often appropriate in Agile contexts but not always “superior” in every environment — project characteristics, safety requirements, and compliance factors still matter. c) Is not correct. It falsely states equivalence. Heavyweight and lightweight approaches differ significantly in formality, documentation, and stakeholder involvement. d) Is not correct. It ignores objective project factors (iteration length, regulatory needs, risk profile) that guide the choice. Team preferences alone are insufficient.	TM-1.3.5	K2	1

Section: Project Test Strategy					
14	b	a) Is not correct. This option focuses more on logistics and tools, not the strategy-level structure of the test approach. b) Is correct. Section 1.4.1 states that key decisions in choosing a test approach include selecting appropriate test levels, test types, and test techniques, tailored to the project context and risks. This combination reflects the core elements of a test approach (Syllabus v3.0, Section 1.4.1). c) Is not correct. These are planning and management concerns, not defining the test approach itself. d) Is not correct. Review methods and KPIs are supporting practices, not the central elements of a test approach.	TM-1.4.1	K2	1

15	a	a) Is correct. This option represents a comprehensive test approach aligned with safety-critical contexts. It includes early static analysis, formal reviews, both white-box and black-box testing, full traceability across all test levels, defect taxonomy use, and targeted automation. These elements are all consistent with the analytical, requirements-based, and risk-based approaches emphasized in the syllabus for regulated projects (Syllabus, Section 1.4.2 and Section 1.3.4). b) Is not correct. This approach appears strong—covering full traceability, reviews, automation, and root cause analysis. However, delaying static analysis until after integration is not acceptable in a safety-critical project. Functional safety standards usually demand that static techniques be applied early in the lifecycle to prevent defect injection. Postponing them reduces preventive value and increases rework risk. c) Is not correct. While it includes structured test specifications, integration interface coverage, and defect classification, it is not rigorous enough for a regulated safety-critical V-model context. It explicitly links only high-level requirements to test results and states that traceability is not consistently maintained across all test levels, which creates gaps in end-to-end traceability—critical for auditable evidence, especially given supplier reluctance and prior audit findings. In addition, applying only partial static analysis weakens early defect prevention; in safety-critical projects, static techniques (reviews/static analysis) should be applied early to requirements/design/code and the overall rigor (levels/types/techniques and reviews) must be tailored to the project risks and regulatory requirements d) Is not correct. Risk-based prioritization is valid, and applying automation plus reviews is appropriate. However, limiting formal reviews to only system-level and relying on lightweight peer reviews at component and integration levels is insufficient in a safety-critical setting. Applying code analysis only to selected items undermines systematic defect prevention.	TM-1.4.2	K4	3

16	a	a) Is correct. It aligns with the organizational test strategy (risk-based, regulatory compliance emphasis), the quality goals (defect detection, security focus), and the project constraints (budget, time, FDA - U.S. Food and Drug Administration- approval). Collaboration between developers and testers plus targeted automation fits the organizational maturity (intermediate automation). Risk-based prioritization directly supports compliance-first goals. b) Is not correct. A predominantly checklist-driven, manual approach across all levels is unlikely to provide sufficient depth and efficiency for a compliance-critical healthcare app under a fixed budget and hard regulatory deadline. It also risks inadequate coverage of high-impact privacy/security and integration risks (e.g., with the three hospital systems) and may not produce the level of robust, risk-focused evidence expected for regulatory validation. c) Is not correct. Lack of explicit test levels contradicts the organizational need for a structured, compliance-first test strategy. Exploratory-only testing cannot ensure FDA (U.S. Food and Drug Administration) validation and regulatory approval. d) Is not correct. Although risk-based ideas are mentioned, reducing the test approach to only two levels and relying on exploratory testing for system testing is insufficient for a regulated healthcare context. System testing requires structured, traceable techniques, coverage of all acceptance criteria, and demonstrable compliance — far more than exploratory sessions on “done” stories. In fact, a story is only considered done after testing, so system testing cannot depend solely on exploratory checks performed after developers mark a feature complete. This option therefore fails to meet the required organizational test strategy for regulatory standards and does not provide the rigor expected at system-test level.	TM-1.4.2	K4	3

17	b	a) Is not correct. This option is Measurable (M) but fails the Achievable (A) criterion of S.M.A.R.T. objectives. While response time and vulnerabilities are quantified, requiring zero vulnerabilities and a 1-second response time without context is unrealistic for a complex mobile banking system. Reference: S = specific. A project test objective and exit criterion should be clear and unambiguous. (Syllabus 1.4.3) A = achievable. It should be feasible considering the available resources, timeframe and capabilities. (Syllabus 1.4.3) b) Is correct. This option aligns with all S.M.A.R.T. criteria: Specific (S): Clearly defines security and performance benchmarks. Measurable (M): Provides quantifiable criteria (e.g., 0 open security flaws, 100 transactions per second, responses under 2 seconds). Achievable (A): Reasonable given the project's constraints. Relevant (R): Directly supports the application's quality goals. Timely (T): Has a defined deadline ("by the end of system testing"). c) Is not correct. Although this option contains a measurable comparison, it still fails the Specific (S) and Achievable (A) aspects of S.M.A.R.T. because competitors are undefined and "more secure" has no objective security criteria. Reference: "A project test objective and exit criterion should be clear and unambiguous." (Syllabus 1.4.3) d) Is not correct. This option introduces a partial Measurable (M) element ("three benchmarks"), but it still lacks Specific (S) and a clearly defined Time-bound (T) element, as no concrete standards, target values, or explicit timeframe/phase are defined (and "before launch" is not sufficiently precise). Reference: "It should be quantifiable and have specific criteria for measuring progress to determine whether it has been reached" (Syllabus 1.4.3).	TM-1.4.3	K3	2

18	a	a) Is correct. This test objective meets all S.M.A.R.T. criteria and is measurable during the system test phase: Specific: Focuses on response time of a key feature (audio lesson loading). Measurable: Quantified by a 2-second maximum on 95% of devices. Achievable: Feasible considering the available resources and test environment/device coverage. Relevant: Tied to user experience and stakeholder expectations for performance. Time-bound: Must be met by the end of the system test phase. This reflects proper application of the S.M.A.R.T. methodology as described in the syllabus section 1.4.3. b) Is not correct. Even though it mentions usability, the objective is still too vague and lacks key S.M.A.R.T. criteria. There is no measurable usability metric, no acceptance threshold, and no timeline. “Improving user experience” or “enhancing usability” without quantification cannot be validated through testing. c) Is not correct. Although relevant and well-intentioned, it lacks measurable and time-bound elements. No defined metric for what constitutes validation is provided, and timeline and success criteria are missing. Therefore, it does not fulfill M and T (and makes A difficult to assess). d) Is not correct. Adding “usability improvements” does not make this a S.M.A.R.T. objective. The completion rate target is not defined in a testable way within the test process (no measurement method, acceptance threshold, device scope, or clear time constraints). It therefore lacks clear, testable usability criteria and a timeframe.	TM-1.4.3	K3	2

Section: Improving the Testing Process					
19	b	a) Is not correct. This activity—finalizing scope and gaining buy-in—belongs to the Initiating phase. In the scenario, this has already happened (budget approved and stakeholder agreement obtained). “At the start of the improvement process, the objectives and scope... are agreed upon by stakeholders.” (Initiating Phase, Section 1.5.1) b) Is correct. After an assessment (diagnosing), the next step in the IDEAL model is to establish a test process improvement plan, including prioritization of actions. “A test process improvement plan can be a formal document… The list of possible process improvements should be prioritized.” (Establishing Phase, Section 1.5.1) c) Is not correct. Diagnosing has already been completed per the scenario: the assessment was just finalized. This option represents a step already done. “The current test process is assessed to identify possible improvements… typically made against a standard framework… or specific metrics...” (Diagnosing Phase) d) Is not correct. Training and piloting are part of the Acting phase, which comes after the plan is created and approved. “This typically includes training and piloting of changed processes and their full deployment in the project or team.” (Acting Phase, Section 1.5.1)	TM-1.5.1	K2	1

20	a	a) Is correct. TMMi is a maturity model designed for structured, staged process improvement, requiring formal assessments, defined maturity levels, and long-term commitment. In this scenario, management explicitly requires lightweight, short-term improvements at project level only, making TMMi disproportionately heavy and slow for the intended goal. Syllabus describes maturity models as frameworks for systematic, long-term test process improvement, which does not match the constraints of this situation. b) Is not correct. TPI Next supports flexible selection of process areas and is well-suited for incremental improvements at the project level. TPI Next provides a key area-based structure and supports selecting subsets of improvement areas, making it suitable for project-specific improvements. The syllabus (section 1.5) emphasizes that test process improvement should be adaptable to the organizational and project context, and that incremental approaches are appropriate when rapid, focused improvement is required rather than long-term maturity growth. c) Is not correct. Agile-aligned practices and context-driven improvements are appropriate for short-term, flexible changes. d) Is not correct. Kanban principles help with team-level workflow visualization and continuous improvement, well-suited for Agile projects.	TM-1.5.3	K2	1

21	a, c	a) Is correct. This supports root cause analysis, a core element of retrospectives. Analyzing where the test process failed allows the team to address gaps and improve future testing. "Root cause analysis can be applied to identify root causes of identified problems…" (see syllabus section 1.5.4 Retrospectives) b) Is not correct. Updating test data is an operational planning task, not a core retrospective activity focused on evaluation and improvement of testing methods. "Retrospectives aim at generating lessons learned… to better manage future projects." (see syllabus section 1.5.4 Retrospectives) c) Is correct. This aligns with the data collection phase of retrospectives. Quantitative metrics like requirement/test coverage help the team evaluate test effectiveness. "Quantitative data... for test progress, defect detection, test effectiveness... can provide an objective insight into the testing of the project or iteration." (see syllabus section 1.5.4 Retrospectives) d) Is not correct. Investigating whether customer or end-user mistakes caused the problem shifts the focus outside the team, while retrospectives—as described in Section 1.5.4—concentrate on improving the team's own processes, collaboration, tools, and testing practices. Even though UX insights may be valuable, determining whether users made mistakes is a product-analysis activity, not a retro objective. Retrospectives aim to generate lessons learned about how the team worked and how the test process can be improved, not to analyze external user behaviour. e) Is not correct. Reviewing usability alignment with branding is a product improvement task, not part of retrospective evaluation of the test process itself. "Retrospectives address topics such as process, people, organization, collaboration, and tools." (see syllabus section 1.5.4 Retrospectives)	TM-1.5.4	K3	2

22	a	a) Is correct. This option aligns with the structured retrospective approach described in the Syllabus section 1.5.4. The retrospective should collect data (e.g., defect trends, test execution issues), analyze root causes, and define improvement actions with clear ownership. It follows best practices by ensuring data-driven decision-making and continuous improvement. b) Is not correct. Retrospectives are meant to be team-driven, focusing on open discussions without external pressure: "Retrospectives are performed by the entire team and thus support the whole team approach and foster continuous improvement." (Syllabus 1.5.4). Involving senior management might discourage open communication, as team members may feel hesitant to discuss challenges freely. Instead, retrospective outcomes can be summarized and shared with management afterward. c) Is not correct. While major defects are important, a comprehensive retrospective should analyze both critical failures and smaller issues that could impact future test cycles. Ignoring smaller inefficiencies limits the opportunity for continuous improvement and may lead to recurring issues in future iterations. d) Is not correct. While best practices are useful, retrospectives should be context-specific, addressing actual challenges faced during the project or iteration. A rigid focus on external models, without evaluating real test execution data, may overlook team-specific bottlenecks and process inefficiencies.	TM-1.5.4	K3	2

Section: Test Tools					
23	b	a) Is not correct. A rollout after validation is aligned with the best practices described in Syllabus v3.0, Section 1.6.1: “Generic good practices in the adoption and roll out of a tool include: Run a pilot project to validate the selection criteria and requirements and to evaluate how the tool fits with existing processes and practices. Adapt and improve processes to fit with the use of the tool, also adapt the tool to existing processes, if necessary. Define guidelines for the use of the tool. Provide training, coaching, and mentoring for tool users. Roll out the tool to the organization in increments.” b) Is correct. Replacing a tool without process alignment or assessment violates key principles from the syllabus, which emphasizes evaluation against organizational needs (see Syllabus, Section 1.6.1) c) Is not correct. Using objective, requirement-based evaluation criteria is a core recommendation in tool selection. d) Is not correct. A pilot project helps verify tool fit and gain feedback before wider rollout, and is a clearly defined best practice in the syllabus. Reference: “Generic good practices in the adoption and roll out of a tool include: Run a pilot project to validate the selection criteria and requirements and to evaluate how the tool fits with existing processes and practices” (Section 1.6.1)	TM-1.6.1	K2	1

24	b, c	a) Is not correct. How well the tool integrates with both Agile and V-model testing approaches (note: stakeholders have already confirmed that both shortlisted tools support this). Therefore, this factor does not differentiate between the shortlisted tools and should not guide the final decision**.** References: "Hybrid software development models integrate elements from both traditional sequential approaches and Agile practices to suit specific project needs or organizational transitions." (Section 1.2.3) "In a hybrid setting, test management activities may include evaluating the team's ability to transition between traditional and Agile methodologies and ensuring structured processes coexist with Agile flexibility."(Section 1.2.4) b) Is correct. This option is the best answer because compliance and Agile flexibility are critical in a dual-model environment. A test management tool must meet safety-critical regulatory requirements and must also be flexible enough to support Agile processes for cloud-based fleet management applications. Regulatory compliance directly impacts legal liability, safety, and the ability to sell products in the automotive industry. Reference: “Organizations that develop safety-critical or mission-critical software, or are subject to regulatory compliance, may prefer commercial tools as they more often meet the required standards and often possess appropriate certification.” (Section 1.6.2) c) Is correct. This option is valid and important for the final decision because the scenario explicitly allows using total cost of ownership (TCO). Therefore, both recurring costs (e.g., licensing, maintenance) and non-recurring costs (e.g., training, support, configuration effort) must be considered. Reference: “A cost-benefit analysis should be performed before acquiring or building a tool… This analysis should take both recurring and non-recurring costs into account.” (Section 1.6.3)	TM-1.6.3	K4	3

		d) Is not correct. The ability of the organization to adapt its test processes to a new tool (even though the organization will not change its test processes to fit the tool). Because the scenario explicitly states that the organization will not change its test processes to fit the tool, process adaptability cannot be used as a guiding factor for the final selection decision. While adaptability is useful, regulatory compliance requirements are non-negotiable. In safety-critical industries, processes must be aligned with regulations first, not adapted for convenience. Test tools must meet process and compliance needs, rather than forcing organizations to adapt around a non-compliant tool. References: "A test manager must evaluate how a tool integrates into existing organizational processes, but regulatory compliance takes precedence over process adaptation." (Section 1.6.2) e) Is not correct. The scenario explicitly states that the vendor roadmap and future feature promises must not be considered; the final decision must be based only on verified current capabilities and total cost of ownership. Therefore, roadmap items cannot guide the final selection decision.			

25	c	a) Is not correct. While Tool X is cheapest and easiest to onboard and **can** achieve a positive ROI within two years based on the projected manual effort reduction, **it** does not meet the compliance must-haves (only partial traceability and no security testing support). Therefore, it is not a suitable final choice despite its cost and usability advantages. b) Is not correct. Tool Y offers balanced value and positive ROI, but it lacks explicit support for security testing. This conflicts with compliance team priorities. c) Is correct. Tool Z addresses regulatory compliance through security test support, delivers the highest measurable benefits (manual effort reduction, cycle-time, coverage), and achieves ROI despite higher cost and usability challenges. It best aligns with management's ROI goals, compliance needs, and long-term organizational value. d) Is not correct. Oversimplifies the choice. ROI and compliance are not "negligible" factors — they decisively favor Tool Z.	TM-1.6.3	K4	3
26	c	a) Is not correct. This describes acquisition activities (assigning tool owner, defining conventions). b) Is not correct. This belongs to support and maintenance (ensuring interoperability, backups). c) Is correct. Archiving or preserving data is explicitly part of the retirement stage, when the tool is replaced and data must remain accessible. d) Is not correct. This describes evolution, where business or vendor changes require adapting the tool.	TM-1.6.4	K2	1

Section: Test Metrics					
27	a	a) Is correct**.** Metric 1 is primarily used for test monitoring and control (progress vs. plan) and metric 2 for test completion (achievement of objectives/exit criteria); metric 3 reports product risk status and metric 4 reports defect status. b) Is not correct. Metric 2 is a test completion metric (not product risk status reporting), and metric 3 is explicitly a product risk reporting metric (not a completion metric). c) Is not correct. Metric 1 belongs to test monitoring and control, and metric 2 belongs to test completion—so 1B and 2A are swapped against the syllabus examples. d) Is not correct. Metric 3 reports product risk status (not defect status), and metric 4 reports defect status (not product risk status).	TM-2.1.1	K2	1

28	a	a) Is correct. Test metrics are primarily used to measure and monitor test progress against planned objectives and exit criteria. They help assess whether testing goals have been met, aligning with the Syllabus, Section 2.1.2: "Test metrics are indicators that show how far the test has progressed and whether the exit criteria or the related test tasks have been achieved." b) Is not correct. This describes test control, which uses metrics to guide decisions. Metrics alone do not recommend corrective actions; they provide information upon which actions can be based. "Test control uses the information from test monitoring to provide guidance and corrective actions." (see the Syllabus, Section 2.1.2) c) Is not correct. This describes part of test completion, not the primary role of test metrics during active monitoring. While historical metrics are collected at closure, the key goal of test metrics is progress tracking during execution. "Test completion collects data from completed test activities." (see the Syllabus, Section 2.1.2) d) Is not correct. Metrics support test prioritization decisions but do not automatically trigger reprioritization. Decision-making belongs to test control processes, not to metrics themselves. "Examples of test control directives include reprioritizing tests when an identified risk becomes an issue." (see the Syllabus, Section 2.1.2)	TM-2.1.2	K2	1

29	b, c	a) Is not correct. Focuses on technical depth rather than decision value — more detail does not equal clarity for stakeholders. (Section 2.1.3) b) Is correct. The option directly links test progress to risk mitigation, which empowers stakeholders to make informed release decisions. (Sections 2.1.3) c) Is correct. The option translates technical results into stakeholder-friendly insights. Reference: "Of particular importance is the need to relate the status of test work products and activities in a manner that is understandable and relevant to the project and business stakeholders. (Section 1.1.2), additionally section 2.1.3) d) Is not correct. Replacing metric-based reporting with subjective stability ratings reduces measurability against test objectives/exit criteria; expert judgment may complement metrics, but should not substitute them for transparent decision-making. (section 2.1.2) e) Is not correct. Overloads the report with low-level data, irrelevant for decision-making. Reference: "Of particular importance is the need to relate the status of test work products and activities in a manner that is understandable and relevant to the project and business stakeholders." (Sections 1.1.2) additionally section 2.1.3)	TM-2.1.3	K4	3

30	c	a) Is not correct. While product risks, defect trends, and test environment stability can all provide useful reporting information, this option is less complete for stakeholder decision-making in a regulated, document-centric sequential project. It does not explicitly include requirement coverage, which is especially important where compliance and documented verification of requirements are mandatory. In addition, overall test execution effort is less precise than comparing actual versus planned test effort/resources in formal project reporting. b) Is not correct. Although defect severity trends and test progress are valid reporting elements, this option is less appropriate for system-level stakeholder reporting in a regulated sequential project. Code coverage is more relevant to lower test levels and technical analysis than to stakeholder-focused system test reporting. In addition, requirements change frequency is less important in this context because the scenario states that requirements are stable, while requirement coverage is far more relevant for demonstrating compliance and verification completeness. c) Is correct. This option aligns fully with the needs of a government, document-centric project: Product risk metrics show whether identified high-risk areas (e.g., security, privacy) are sufficiently tested. Defect detection and resolution rates reflect system quality and the maturity of the product toward release. Test execution progress allows monitoring whether testing activities meet planned timelines. Requirement coverage ensures that stakeholder and regulatory requirements are properly validated. Test effort tracking supports resource and budget control — critical in public sector projects. "Product risks, defects, test progress, coverage, and related cost and test effort are measured and reported in specific ways at the end of the project." (Syllabus Section 2.1.3) d) Is not correct. While defects, test progress, overall test coverage, and residual risks in not-yet-tested areas are all relevant reporting elements, this set is less complete than c). It does not explicitly include requirement coverage and defect resolution, both of which are important for stakeholder decision-making in regulated projects. It also does not explicitly address test effort as actual versus planned resource hours, which supports management oversight in formal project reporting.	TM-2.1.3	K4	3

Section: Test Estimation					
31	c	a) Is not correct. The planned test approach and degree of test automation are valid inputs for test effort estimation because they are development-process-related factors (e.g., use of tools/automation and required setup) that directly influence the amount of effort needed. (Section 2.2.2) b) Is not correct. The scope and technical complexity of the product under test is a standard factor in estimation. Larger and more complex systems require proportionally more test effort. (Section 2.2.2) c) Is correct. The number of improvement actions identified in the last retrospective is not a standard or reliable estimation input. Retrospectives often mix product, process, and communication issues. Even if some defect counts appear, they lack the systematic context (e.g., severity, density, injection phase) that makes defect metrics useful for estimation. Therefore, this metric cannot be considered a valid input. d) Is not correct. The experience level and domain knowledge of the assigned testers directly affects productivity and accuracy. As the syllabus states, “the skills and experience of the people involved” must be considered in test effort estimation (Section 2.2.2).	TM-2.2.2	K2	1

32	b, d	a) Is not correct. The stem states that only some scripted regression checks are part of the approach. Planning for full end-to-end regression coverage with detailed test scripts contradicts this and would not reflect the described testing strategy. b) Is correct. The stem highlights risk-based testing for core system components. Using historical data from previous sprints to calculate average effort per risk item directly supports effort estimation in this context. c) Is not correct. The stem specifies a combination of risk-based testing, exploratory testing, and scripted regression checks. Assuming exploratory testing alone is sufficient ignores the other test activities and underestimates effort. d) Is correct. The stem mentions exploratory testing for frequently changing modules. Allocating fixed testing slots based on test charters helps structure and estimate this exploratory effort, making it consistent with the scenario. e) Is not correct. The stem explicitly associates frequently changing modules with exploratory testing. Focusing estimation only on new features and assuming regression remains stable overlooks both the exploratory effort and risk-based activities, leading to incomplete estimation.	TM-2.2.3	K4	3

33	c	a) Is not correct. Three-point estimation requires optimistic, pessimistic, and most-likely values that are typically derived from reliable expert judgment. In this project, the team is not yet fully assigned and direct expert input is limited, so producing credible ranges is difficult at this stage; therefore, this approach is not the MOST appropriate here. b) Is not correct. Planning Poker is an Agile technique, designed for story-based planning with evolving scope and iterative delivery. Here the project has fixed scope, a sequential model, and a need for an early structured estimate rather than incremental, collaborative guessing. This makes Planning Poker unsuitable. c) Is correct. Data-driven estimation is the most reliable method in this context. The project has detailed requirements, a fixed scope, and some historical data from similar regulatory projects, making requirement-based sizing with historical averages the most objective, repeatable, and defensible approach. This technique supports early estimation even when the full team is not yet assigned, which matches the project's constraints. d) Is not correct. While the Wideband Delphi method can be useful, it relies on the availability of experienced experts to iteratively refine estimates through structured consensus. In this project, the team composition is not yet finalized and expert input is limited, making Delphi impractical; since historical data from similar projects exists, an analytical, data-driven approach is preferable.	TM-2.2.3	K4	3

Section: Defect Management					
34	c	a) Is not correct. An environment/configuration-related issue is still an anomaly that should be handled within the defect management process (e.g., to ensure traceability and prevent recurrence), so it should not be tracked outside defect reporting. (section 2.3.1, section 2.3.5) b) Is not correct. Even if the test case is not part of the approved test basis, the anomaly should be documented and may then be rejected with a clear reason (e.g., incorrect test/invalid information), so it is still processed via a defect report. (section 2.3.1) c) Is correct. In automated component tests used for test-driven development, failures before the functionality is implemented are expected and are typically not tracked via a defect report, but within the development task. (section 2.3.1) d) Is not correct. This situation should be handled within the defect workflow by referencing/updating (and if applicable re-opening) the existing defect report, not by tracking it outside defect reporting. (section 2.3.1)	TM-2.3.1	K3	2

35	b	a) Is not correct. Although several transitions are correct, Deferred → Rejected is not supported by the rules. Deferral should only lead back to Open. b) Is correct. All four transitions fully comply with the established rules: Open → Deferred ensures only unfixed defects can be postponed. RESOLVED → RE-OPENED is valid because a defect marked as RESOLVED may still require further work (e.g., verification fails), so it returns to active fixing. Deferred → Open is valid since postponed defects may later be reconsidered. Re-Opened → In Progress is valid since reopened defects should return to active fixing. c) Is not correct. It incorrectly uses CLOSED → RE-OPENED instead of RESOLVED → RE-OPENED, and it allows DEFERRED → CLOSED (not supported) and RE-OPENED → CLOSED (skips IN PROGRESS). d) Is not correct. It includes RE-OPENED → OPEN (should return to IN PROGRESS) and DEFERRED → CLOSED (not supported), and it also uses CLOSED → RE-OPENED instead of RESOLVED → RE-OPENED.	TM-2.3.1	K3	2

36	b	a) Is not correct. In the simple defect workflow, CLOSED is the terminal state; this sequence ends in RE-OPENED, and it also implies an unsupported transition from REJECTED to RE-OPENED, so it is not a valid complete lifecycle leading to the terminal state. b) Is correct. This matches the typical, complete defect lifecycle: OPEN: Defect detected and logged. IN PROGRESS: Being addressed by responsible parties (development, etc.). RESOLVED: Fixed and pending confirmation. CLOSED: Confirmed fixed and archived. Good defect workflows ensure clear resolution and closure, and involve proper communication among cross-functional stakeholders (testers, developers, product owners). c) Is not correct. A defect cannot be IN PROGRESS before it is even opened. OPEN must precede IN PROGRESS in any standard workflow. d) Is not correct. DEFERRED is not part of the simple defect workflow, and the sequence does not end in the terminal state CLOSED; therefore it is not a valid complete lifecycle to a terminal state.	TM-2.3.2	K2	1

37	a	a) Is correct. In Agile teams, it is common practice to discuss defects informally with developers first. According to Syllabus, Section 2.3.3 information about a defect is often exchanged informally among testers and developers without a formal defect report, particularly when teams are co-located or have efficient communication. Only after clarification should a formal defect report be created if needed. b) Is not correct. Although focusing on the sprint backlog is important, this is not a valid reason to delay defect reporting. Defects that block current work, cannot be fixed immediately, or need further prioritization must still be tracked properly according to the syllabus, Section 2.3.3. c) Is not correct. A defect's perceived "minor" impact is not a sufficient reason to delay tracking; if it cannot be resolved within the same iteration (or the team's agreed short timeframe), it should be recorded and typically added to the product backlog for later prioritization. (Section 2.3.3) d) Is not correct. If the defect cannot be resolved within the same iteration, common practice is to add it to the product backlog so it can be prioritized for a later iteration; postponing documentation until after the next sprint planning reduces transparency. (Section 2.3.3)	TM-2.3.3	K2	1

38	a	a) Is correct. The syllabus notes that hybrid/multi-team environments require alignment on defect attributes and tools to maintain consistency and traceability; ideally all teams use one defect management tool, and if multiple tools are used, synchronization between them (preferably automated) should be established. Reference: "In an ideal scenario all teams use one defect management tool. In practice it is common for each team to use a different defect management tool… In such cases it is good to establish synchronization between the defect management tools (preferably automatically). "(Syllabus, Section 2.3.4) b) Is not correct. While autonomy is valued in Agile, completely independent defect handling creates fragmentation. Cross-team coordination is required for reliable defect management. Reference: "Such a multi-team environment poses various challenges: Alignment on defect attributes and tools to be used for defect management…"(Syllabus, Section 2.3.4) c) Is not correct. Centralizing defect handling under the sequential team disregards the faster feedback loops required in Agile. Reference: "Defect management meetings should be held more often with Agile software development than in sequential development models to keep pace with the Agile team's faster rate of product increment delivery."(Syllabus, Section 2.3.4) d) Is not correct. The syllabus emphasizes context-driven defect management, where formality is adapted to team setup, distribution, and maturity — not enforced unilaterally from one process model (see Syllabus, Section 2.3.3)	TM-2.3.4	K2	1

39	c	a) Is not correct. A brief summary describing the defect condition is mandatory. "To manage the defect report in most environments, the following are mandatory: A defect title with a short summary of the anomaly." (Syllabus, Section 2.3.5) b) Is not correct. A detailed defect description, including steps to reproduce and observed behavior, is mandatory. "A detailed description of the anomaly preferably including steps to reproduce the failure" is listed as a mandatory field for defect management. (Syllabus, Section 2.3.5) c) Is correct. The subsystem or component in which the defect lies is helpful for defect resolution but not mandatory for defect management. "To help defect resolution: The subsystem or component in which the defect lies..." is listed as additional information depending on the context, not core mandatory data. (Syllabus, Section 2.3.5) d) Is not correct. The priority indicating how soon the defect should be addressed is mandatory. "Priority to fix the anomaly" is listed among the mandatory fields required for defect management. (Syllabus, Section 2.3.5)	TM-2.3.5	K3	2

40	a, e	a) Is correct. Essential for reproducibility, especially in multi-vendor and complex environments. (Section 2.3.5) b) Is not correct. Business impact can be useful depending on context, but the supplier's concern is insufficient detail for triage/analysis; the most impactful missing elements are reproducibility (steps/expected vs. actual) and environment/configuration. (Section 2.3.5) c) Is not correct. The software component where the defect was detected is useful information, but it is less important here than the missing details needed for effective triage and analysis across multiple teams and suppliers. In addition, the scenario states that the defect reporting tool already enforces recording the affected component/subsystem, so selecting c) would not improve the completeness of the defect reports in this case. (Section 2.3.5) d) Is not correct. References to connected defects are useful and explicitly mentioned, but in this scenario they are already maintained automatically by the tool, so selecting d) would not improve report completeness. (Section 2.3.5.) e) Is correct. Crucial for understanding whether observed behavior is truly faulty. Allows developers and third parties to replicate and debug accurately. (Section 2.3.5)	TM-2.3.5	K3	2

41	c	a) Is not correct. The number of open defects per release may help assess backlog health or release readiness, but it does not indicate when or where in the process those defects were introduced or missed. Therefore, it is less useful for improving early defect detection b) Is not correct. Severity levels of production defects lack insight into the effectiveness of earlier test stages c) Is correct. The defect detection rate at each test level shows how effectively defects are found during unit testing, integration testing, and system testing before they escape to later stages. By comparing these rates across levels, the test manager can identify where defects are being detected too late and which earlier activities need improvement to shift detection left and reduce late failure costs. This directly supports process improvement as described in Syllabus Section 2.3.6, which states that defect statistics should be used to identify weaknesses in the test process and improve early defect detection. d) Is not correct. The percentage of reopened defects is a valid quality indicator related to resolution accuracy and communication, but it does not provide visibility into how effective early test levels are at catching defects.	TM-2.3.6	K2	1

Section: Test Team					
42	c	a) Is not correct. Evaluating test coverage based on requirements traceability is an example of professional competence. It deals with a specialized task requiring specific testing knowledge. b) Is not correct. Resolving conflicts between developers and testers is an example of social competence, involving interpersonal communication and conflict management skills. c) Is correct. The ability to independently analyze complex problems and develop solutions exemplifies methodological competence. As per the syllabus, methodological competence includes analytical, conceptual, and judgmental skills enabling independent performance of complex tasks. (see syllabus, Section 3.1.1) d) Is not correct. Updating automated test scripts falls under professional competence, as it pertains to technical, task-specific testing skills.	TM-3.1.1	K2	1

43	b, d	a) Is not correct. Automation frameworks and data generation are valuable for component or regression testing, but not as directly applicable to manual/system-level test design and traceability. Automation may assist execution, but test design and documentation are the key challenges in this role b) Is correct. The project demands compliance with EN 50128, requiring traceability, auditability, and formal documentation. A tester must be able to work within the structured, regulated testing context of the V-model c) Is not correct. Collaboration with Agile teams is useful, but it is not the most critical skill for this particular replacement role. The responsibilities listed in the question emphasize: - Designing and documenting risk-based system tests from safety and interface requirements - Maintaining traceability of test artifacts for regulatory audit - Working in a safety-critical, standards-driven environment (e.g., EN 50128) - These require strong capabilities in safety standards and formal documentation (option b) and in risk-based, requirements-based black-box test design (option d). Experience collaborating on user stories and acceptance criteria alone would not be sufficient to cover the core needs of this role, especially around compliance and traceability. d) Is correct. Risk-based testing and black-box test design are core to the role described: system-level test design, using black-box techniques from documented requirements. Risk-based prioritization is crucial in safety-critical environments e) Is not correct. Interface testing and cross-team coordination are helpful, but not the top priority in this context, especially in a hybrid model. However, it lacks the emphasis on formal safety-compliant documentation and systematic test design techniques, which are higher priority here.	TM-3.1.2	K4	3

44	a	a) Is correct. This option reflects the competence areas that are most relevant in a hybrid project with both Agile and compliance-driven components, as described in the stem. Automation and scripting are needed because hybrid environments typically include reusable regression checks and integration-level workflows where automation reduces manual effort and supports consistency across releases. Communication and collaboration are essential due to the hybrid nature of the project, where testers must work with multiple stakeholder groups and align practices across different development styles. Documentation and traceability are required for the compliance-driven parts of the project, where auditability and evidence-based reporting are mandatory. Security testing competence is critical because the project handles sensitive personal data. Adaptability is necessary in hybrid contexts, where testers must switch between Agile-oriented collaboration and structured, regulated testing activities. Together, these elements represent a balanced set of professional, methodological, social, and personal competences as outlined in Section 3.1.1 (areas of competence) and applied via Section 3.1.2 (context-based skill analysis) of the syllabus, making this option the best fit. b) Is not correct. While this option covers compliance/traceability and includes basic scripting, it does not explicitly address security and privacy testing, which is a central risk driver in the stated context (sensitive personal data and strong security/privacy requirements). In addition, it places less emphasis on adaptability to frequent change and distributed collaboration needs highlighted in the stem, making it a less complete match than option a). c) Is not correct. This option emphasizes performance and customer feedback loops, which are less central in this context.	TM-3.1.2	K4	3

		It omits critical elements like traceability, regulatory understanding, and security testing, making it an incomplete match. d) Is not correct. This option is better suited to a sequential legacy system context. It fails to address collaboration in Agile teams, automation, evolving architecture, or security—key factors in this hybrid and security-sensitive environment.			

45	d	a) Is not correct. Drafting a test strategy outline relates to professional competence, but the second part of the option misclassifies the competence involved in handling feedback. According to Section 3.1.1, social competence includes communication, cooperation, and interaction with others. Handling feedback during a peer discussion requires these interaction and communication skills, making it primarily a matter of social competence. Personal competence includes openness to criticism, but this refers to an individual attitude, not the interactive communication process during a feedback exchange. Therefore, the option mixes competence types and is not aligned with the syllabus definitions. b) Is not correct. This option misclassifies competencies. Collaboration skills are indeed part of social competence, but technical expertise is clearly defined in the syllabus as part of professional competence, not personal. Personal competence focuses more on self-management and openness to learning, not technical knowledge. Therefore, while both elements may appear superficially relevant, their classification contradicts the competence model described in Section 3.1.1. c) Is not correct. Collaboration skills are not part of professional competence but fall under social competence. Moreover, creativity in writing is not listed among the key elements of methodological competence, which is instead defined as including skills in analysis, planning, and decision-making. The competencies here are mismatched both in type and focus, making this a poor choice under the given scenario. d) Is correct. Drafting a high-level test strategy outline is a specialized testing task and therefore assesses professional competence. Structuring and reasoning about the strategy assesses methodological competence (analytical/conceptual/judgmental skills). This is consistent with the syllabus example that professional and methodological competence can be assessed by outlining a test strategy and discussing feedback with colleagues.	TM-3.1.3	K2	1

46	c	a) Is not correct. Self-study is ideal for building professional and methodological competence but is not recommended for developing social or personal skills, which require interaction, feedback, and reflection. b) Is not correct. Peer learning is a mutual knowledge-sharing approach between colleagues, not structured coaching. Coaching and mentoring, by contrast, involve structured, individual support from a more experienced person. c) Is correct. Training sessions are particularly well-suited for developing professional and methodological competence, as they teach predefined knowledge and structured practices. (Syllabus TM-3.1.4) d) Is not correct. Coaching provides individual, personalized guidance to help develop competencies, not predefined knowledge to a group. That would characterize training, not coaching.	TM-3.1.4	K2	1

47	b	a) Is not correct. While technical coaching is valuable, in this early stage of team development (before agreed working rules are established) the bigger challenge is not yet technical competence but establishing collaboration, communication, and clarity of roles. Coaching becomes more effective once team foundations are stable. b) Is correct. When responsibilities are unclear and coordination routines are missing, the test manager should first help the team agree on clear working rules and responsibilities. In the early stages of team development, structure, role clarity, and agreed ways of working are essential to enable effective collaboration and reduce confusion. This aligns with the syllabus emphasis that leadership should support the establishment of working rules and responsibilities as the team develops (Section 3.1.5). c) Is not correct. Encouraging independence is important in later stages (e.g., norming/performing), but in this early stage (before values/rules are agreed) pushing for autonomy too soon may create confusion and inefficiency, as rules and responsibilities are not yet established. d) Is not correct. Delegating tasks immediately may seem efficient, but without agreed working rules or coordination, it risks miscommunication and frustration. Early focus should be on structure and collaboration before task distribution; delegation to give personal responsibility is characteristic of the Performing stage.	TM-3.1.5	K2	1

Section: Stakeholder Relationships					
48	b	a) Is not correct. B (executing test cases) is an appraisal activity (defect detection), while A (developer training) is a prevention activity; option a swaps prevention and appraisal and is therefore incorrect b) Is correct. This option correctly maps each cost category to its respective example, based on the definitions in Syllabus, section 3.2.1, which describes the cost of quality (CoQ) model as comprising prevention, appraisal, internal failure, and external failure costs. A (Developer training) is a prevention cost: it aims to reduce defect injection. B (System test execution) is an appraisal cost: it evaluates quality. D (Fixing defects in integration testing) is an internal failure cost: the defect is found before release. C (Production defects) is an external failure cost: the defect escaped to users. c) Is not correct. C (after release) - defects reported by end users after release are not internal failures. Once a defect is found after the product is released, it becomes an external failure cost, because the customer is impacted. D (Fixing defects in integration testing) - External failure costs apply only after delivery to users. Time spent analyzing and fixing defects found during integration testing is an internal failure cost, not an external one, because the defect is detected before release, customer is not yet impacted, and the cost is limited to internal rework and retesting d) Is not correct. 1D is incorrect: Fixing defects is not prevention. 2C is incorrect: User-reported defects are not appraisal. 3A is incorrect: Training developers is not an internal failure. 4B is incorrect: System testing is not an external failure. This option misclassifies all four cost types, directly contradicting the Syllabus, section 3.2.1 definitions.	TM-3.2.1	K2	1

49	b	a) Is not correct. $5,300 incorrectly adds mutually exclusive costs: for a defect found before release the cost is Appraisal + Internal ($500), while for a defect found after release the cost is External ($4,800) instead. It also does not represent “the total cost of testing,” and prevention costs are not included. b) Is correct. Correct use of the formula: Savings = External failure cost – (Appraisal + Internal failure costs) → $4,800 – ($200 + $300) = $4,300 “Average Savings per Defect = Average of External Failure Costs – (Average Appraisal Costs + Average of Internal Failure Costs)” c) Is not correct. This is a common misconception: that reducing test effort (appraisal) will lower costs. In reality, less appraisal often leads to more external failures, which are far costlier. According to the syllabus (Section 3.2.2), an effective quality approach aims to balance CoQ elements and invest enough in appraisal to avoid expensive failures. d) Is not correct. This is a misrepresentation. Identifying fewer defects due to less testing leads to higher business risk, not improved cost-benefit. “If too little is tested, missed defects can pose a high risk to generate higher costs than the omitted tests would have costed.”	TM-3.2.2	K3	2

50	a, c	a) Is correct. Prevention costs (€200/defect) are significantly lower than internal (€350) or external (€3,000) failure costs. Investing more in prevention reduces the number of defects entering later stages, where fixing is far more expensive. This follows Boehm's cost-of-change curve, which shows that the cost of fixing defects rises significantly the later a defect is discovered. This principle supports the Cost of Quality model described in Syllabus, Sections 3.2.1 and 3.2.2, where prevention is shown as the most cost-effective quality investment. b) Is not correct. A defect found before release costs Appraisal + Internal = €400 + €350 = €750, which is still far less than the €3,000 external failure cost; reducing appraisal would therefore tend to increase expensive external failures rather than improve cost-effectiveness. c) Is correct. Appraisal is justified because detecting 70 defects before release avoided external failure costs; the savings follow the syllabus formula: Savings = External − (Appraisal + Internal) = €3,000 − (€400 + €350) = €2,250 per defect → 70 × €2,250 = €157,500. (Syllabus, Section 3.2.2). d) Is not correct. While internal failures (€350) are cheaper than external ones (€3,000), prioritizing internal failure reduction over prevention/appraisal ignores the principle that prevention and appraisal reduce *all* later defect costs. The greatest savings come from avoiding external defects altogether, not from focusing on internal correction (Syllabus, Section 3.2.2). e) Is not correct. Testing delivers measurable value: 70 defects were detected before release, avoiding high external costs. Claiming "no benefit" contradicts both the data and the CoQ model. (Syllabus, Section 3.2.2).	TM-3.2.2	K3	2

Appendix

Question 7A [Source: CTAL-TM 3.0; SAMPLE EXAM PAPER ; SET A]

7A	a	**a) I, ii and iv --- Correct.** These are the most effective short-term priorities to support continuous delivery: risk-based monitoring/control (i), faster regression feedback through automation (ii), and clear release decision rules via quality gates (iv). **b) i and ii — Not correct.** Both are important, but this combination omits (iv), which is critical for consistent release decisions and stakeholder confidence in a daily release context. **c) i and iv — Not correct.** This set lacks (ii). Without strengthening regression automation, daily releases remain fragile and time-boxes are harder to maintain. **d) ii and iv — Not correct.** This set lacks (i). Without continuous monitoring, test control cannot effectively steer priorities based on the latest risk and defect information.	TM-1.2.7	K4	3

Quelle: Question #6A [Source: CTAL-TM 3.0 ; SAMPLE EXAM PAPER ; SET A]:

6a	c	**1 – A (Sequential):** Ist korrekt, weil Testen erst nach abgeschlossener und freigegebener Anforderungsphase beginnt und das Projekt in klar getrennten Phasen mit einem großen Release am Ende verläuft. Das entspricht einem sequenziellen Vorgehen. **2 – B (Iterative):** Ist korrekt, weil in festen 3-Wochen-Iterationen gearbeitet wird, innerhalb jeder Iteration getestet wird und am Ende jeweils ein Inkrement geliefert wird. Das entspricht einem iterativen Vorgehen mit Testing innerhalb der Iterationen. **3 – C (Hybrid):** Ist korrekt, weil zunächst eine klassische, vorgelagerte Anforderungsbasis geschaffen wird, danach aber in Iterationen weitergearbeitet wird. Damit werden sequenzielle und agile/iterative Elemente kombiniert, was dem Lehrplanverständnis eines Hybridmodells entspricht. **4 – D (DevOps):** Ist korrekt, weil Änderungen häufig integriert und sehr oft ausgeliefert werden und dies durch automatisierte Pipelines sowie operatives Monitoring unterstützt wird. Das sind typische DevOps-Merkmale mit kontinuierlicher Lieferung, schnellem Feedback und starker Automatisierung.	TM-1.2.4	K2	1

Question 36A [Source: CTAL-TM 3.0 ; SAMPLE Exam Paper ; SET A]

36A		**Correct sequence: OPEN → IN PROGRESS → RESOLVED → CLOSED** **Why this order is correct:** • **OPEN**: The defect is logged/recorded and is now visible for triage and assignment. This is the typical initial state after creation of a defect report. • **IN PROGRESS**: The defect is being analyzed and worked on by developers or relevant stakeholders. This represents the active handling/implementation phase. • **RESOLVED**: A fix (or another resolution) has been provided and the defect is considered addressed from the development perspective. It may still require verification. • **CLOSED**: The resolution has been verified/accepted (e.g., by testing or review), and no further action is required. This is a terminal state. **Why other sequences would be incorrect (general rationale):** • Any sequence that **starts with IN PROGRESS** is incorrect because work cannot begin before the defect has been **logged (OPEN)** and made available for assignment. • Any sequence that **places CLOSED before RESOLVED** is incorrect because closure typically requires a prior resolution (fix/decision) to close against. • Any sequence that **ends at RESOLVED** is incomplete in this context because **RESOLVED** is not necessarily terminal; it commonly requires verification before the defect can be **CLOSED**. • A sequence that includes a terminal state and then continues (e.g., CLOSED followed by another state) would be inconsistent because a terminal state indicates the workflow has ended.	TM-2.3.2	K2	1

Question #27A [Source: CTAL-TM 3.0; SAMPLE EXAM PAPER; SET A]

27a	1A, 2B, 3C	1. Planned test automation degree is done in test planning. 2. The number of resolved defects vs. total number of defects found is a test monitoring and test control metric. 3. The percentage of actual automated vs. planned automated test cases is a test completion metric. A) Test planning B) Test monitoring and test control C) Test completion	TM-2.1.1	K2	1