

Certified Tester
Quality in DevOps (CT-QDO)
Simulazione d'Esame – Risposte

v1.0

Compatibile con il Syllabus v1.0

International Software Testing Qualifications Board



Avviso di Copyright

Avviso di copyright © International Software Testing Qualifications Board (di seguito denominato ISTQB®).

ISTQB® è un marchio registrato di International Software Testing Qualifications Board.

Copyright © 2026, gli autori Szilárd Széll (product owner), Kari Kakkonen, Rik Marselis, Katja Obring e Yaron Tsubery, e gli autori della simulazione d'esame Tal Pe'er, Daniel Pol'an, Tomas Tumasonis, Aneta Derková.

Tutti i diritti riservati. Con la presente gli autori trasferiscono il copyright a ISTQB®. Gli autori (come attuali titolari del copyright) e ISTQB® (come futuro titolare del copyright) hanno concordato le seguenti condizioni di utilizzo:

È possibile copiare estratti di questo documento per uso non commerciale, a condizione che venga riconosciuta la fonte. Qualsiasi Training Provider accreditato può utilizzare questo syllabus come base per un corso di formazione, a condizione che gli autori e ISTQB® siano riconosciuti come fonte e titolari del copyright del syllabus e a condizione che qualsiasi pubblicità di tale corso di formazione possa menzionare il syllabus solo dopo aver ricevuto l'accreditamento ufficiale dei materiali di formazione da un Member Board riconosciuto da ISTQB®.

Qualsiasi individuo o gruppo di individui può utilizzare questo syllabus come base per articoli e libri, a condizione che gli autori e ISTQB® siano riconosciuti come fonte e titolari del copyright del syllabus.

È proibito qualsiasi altro utilizzo di questo syllabus senza aver prima ottenuto l'approvazione scritta di ISTQB®.

Qualsiasi Member Board riconosciuto da ISTQB® può tradurre questo syllabus a condizione che riproduca il suddetto Avviso di Copyright nella versione tradotta del syllabus.

Responsabilità del Documento

L'Examination Working Group di ISTQB® è responsabile di questo documento.

Questo documento è mantenuto da un team di ISTQB® composto dal Syllabus Working Group e dall'Exam Working Group.

Ringraziamenti

Questo documento è stato prodotto da un team di ISTQB®: Tamás Horváth (Vice-Chair Specialist Technologies and Approaches Working Group), Szilárd Széll (product owner), Kari Kakkonen, Rik Marselis, Katja Obring, Tal Pe'er, Daniel Poľan, Yaron Tsubery, Tomas Tumasonis, Aneta Derková.

Le seguenti persone hanno partecipato alla review, ai commenti e alla votazione di questa simulazione d'esame: Laura Albert, Chris Van Bael, Jürgen Beniermann, Renzo Cerquozzi, Walter Eder, Raymond Gillespie, Andreas Guenther, Mattijs Kemmink, John Kurowski, Gary Mogyorodi, Frank Neiryndck, Ralf Pichler, Giorgio Pisani, Horst Pohlmann, Nishan Portoyan, Adam Roman, Jan Sabak, Márton Siska, Mattijs Kemmink, Klaus Skafte, Michael Stahl, Lucjan Stapp, Darvay Tamás Béla, Reifa Tangon, Linda Vreeswijk

Il team ringrazia il team di review dell'Exam Working Group, il Syllabus Working Group e i Member Board per i loro suggerimenti e contributi.

Traduzione italiana a cura di ITA-STQB (www.ita-stqb.org) – info@ita-stqb.org

Cronologia delle Revisioni

Versione	Data	Osservazioni
v1.0	13/02/2026	Rilascio
v0.4	15/01/2026	Editing tecnico dopo il rilascio Beta
v0.3	25/08/2025	Versione di rilascio Beta con 40 + 8 domande.
v0.2	15/07/2025	Versione di editing tecnico con 40 + 8 domande.
v0.1	20/03/2025	Versione Alpha della simulazione d'esame con 40 + 4 domande.

Cronologia delle versioni della traduzione italiana

Versione	Data	Osservazioni
Ver. A	18/09/2026	Prima versione pubblicata della traduzione italiana

Indice dei Contenuti

Avviso di Copyright	2
Responsabilità del Documento	3
Ringraziamenti	4
Cronologia delle Revisioni	5
Chiave delle Risposte	8
Risposte	9
Domanda #1.....	9
Domanda #2.....	9
Domanda #3.....	10
Domanda #4.....	10
Domanda #5.....	11
Domanda #6.....	12
Domanda #7.....	12
Domanda #8.....	12
Domanda #9.....	13
Domanda #10.....	13
Domanda #11.....	14
Domanda #12.....	14
Domanda #13.....	15
Domanda #14.....	15
Domanda #15.....	16
Domanda #16.....	16
Domanda #17.....	17
Domanda #18.....	17
Domanda #19.....	18
Domanda #20.....	18
Domanda #21.....	19
Domanda #22.....	19
Domanda #23.....	20
Domanda #24.....	20
Domanda #25.....	21
Domanda #26.....	21
Domanda #27.....	21
Domanda #28.....	22
Domanda #29.....	22
Domanda #30.....	23
Domanda #31.....	23
Domanda #32.....	24
Domanda #33.....	24
Domanda #34.....	25
Domanda #35.....	25
Domanda #36.....	26
Domanda #37.....	26
Domanda #38.....	26
Domanda #39.....	27

Domanda #40.....	27
Appendice A – Chiave delle Risposte Aggiuntive	28
Appendice B – Risposte Aggiuntive	29
Domanda #A1	29
Domanda #A2	29
Domanda #A3	30
Domanda #A4	30
Domanda #A5	31
Domanda #A6	31
Domanda #A7	32
Domanda #A8	32

Chiave delle Risposte

Numero della Domanda (#)	Risposta Corretta	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
1	b	QDO-1.1.1	K1	1
2	b	QDO-1.1.2	K1	1
3	d	QDO-1.1.3	K2	1
4	d	QDO-1.1.4	K2	1
5	b, d	QDO-1.1.5	K2	1
6	b	QDO-1.1.6	K2	1
7	b	QDO-1.2.1	K2	1
8	c	QDO-1.2.2	K2	1
9	a, e	QDO-2.1.1	K3	2
10	c	QDO-2.1.2	K2	1
11	c	QDO-2.1.3	K2	1
12	b	QDO-2.1.4	K2	1
13	a	QDO-2.2.1	K2	1
14	c	QDO-2.2.2	K2	1
15	d	QDO-2.2.3	K2	1
16	a	QDO-2.2.4	K2	1
17	b	QDO-2.2.6	K3	2
18	c	QDO-3.1.1	K2	1
19	b	QDO-3.1.2	K2	1
20	a	QDO-3.1.3	K3	2

Numero della Domanda (#)	Risposta Corretta	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
21	d	QDO-3.1.4	K2	1
22	b	QDO-3.1.5	K2	1
23	c	QDO-3.1.6	K2	1
24	b	QDO-3.2.1	K2	1
25	b	QDO-3.2.2	K3	2
26	c	QDO-3.2.3	K2	1
27	a	QDO-3.3.1	K2	1
28	c	QDO-3.3.2	K2	1
29	d	QDO-3.3.3	K2	1
30	b	QDO-3.3.4	K3	2
31	b	QDO-4.1.1	K1	1
32	a	QDO-4.1.2	K2	1
33	b	QDO-4.2.1	K1	1
34	d	QDO-4.2.2	K2	1
35	a, c	QDO-4.2.3	K2	1
36	c	QDO-4.2.4	K1	1
37	a	QDO-4.2.5	K1	1
38	c, e	QDO-4.2.6	K1	1
39	d	QDO-4.2.8	K1	1
40	b	QDO-4.2.9	K1	1

Risposte

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
1	b	a) Errata. Sebbene l'impegno individuale sia parte dello sviluppo software, DevOps enfatizza il lavoro di squadra e la collaborazione. b) Corretta. Questa affermazione riflette accuratamente i principi fondamentali di DevOps, che privilegiano collaborazione, comunicazione e miglioramento continuo tra i team. c) Errata. DevOps incoraggia team autonomi e cross-funzionali, ma non enfatizza che un unico team si occupi dell'intero ciclo di vita dello sviluppo software (SDLC). d) Errata. Sebbene le esigenze dei clienti siano un aspetto cruciale di DevOps, dar loro priorità non è il suo scopo primario.	QDO-1.1.1	K1	1
2	b	a) Errata. DevOps mira a integrare i team, non a crearne di separati per la Quality Assurance (QA). b) Corretta. Il syllabus afferma che DevOps abbate il wall of confusion integrando i team affinché lavorino insieme e aumentando la comunicazione. c) Errata. Sebbene l'automazione sia importante in DevOps, il wall of confusion viene abbattuto attraverso l'integrazione e la comunicazione, non con la sola automazione. d) Errata. Sebbene definire ruoli chiari sia importante, il modo principale in cui DevOps rende il testing un fattore chiave per abbattere il wall of confusion è integrare i team e migliorare la comunicazione.	QDO-1.1.2	K1	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
3	d	a) Errata. Il tempo di recupero da un deploy fallito misura il tempo necessario a ripristinare la situazione dopo un deployment fallito, e indica la tolleranza ai guasti e la reattività del software. b) Errata. La frequenza di deploy è la frequenza dei cambiamenti software rilasciati in produzione, e riflette l'agilità e la reattività del processo di consegna. c) Errata. La change fail percentage indica l'affidabilità del processo di consegna misurando la percentuale di deployment che causano failure in produzione. d) Corretta. Il change lead time misura la durata che intercorre tra un commit di codice e un deployment in produzione riuscito, e riflette l'efficienza della pipeline di consegna.	QDO-1.1.3	K2	1
4	d	La cultura DevOps riguarda team che hanno l'autonomia di consegnare valore ai clienti mantenendo però la proprietà della qualità, per cui il testing non viene svolto al di fuori del team DevOps. Pertanto, 1C. L'automazione consiste nell'automatizzare il più possibile nella pipeline di continuous integration/continuous delivery (CI/CD). Pertanto, 2A. Lean significa snellire i processi ed eliminare gli sprechi (per esempio rimuovere i colli di bottiglia nelle attività di test). Pertanto, 3D. La misurazione fornisce dati per le decisioni relative a come il software viene utilizzato e a come funziona il ciclo di vita dello sviluppo software. Pertanto, 4B.	QDO-1.1.4	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
5	b, d	a) Errata. Questa opzione descrive un modello di sviluppo sequenziale, che non riflette la natura continua e integrata della QA in DevOps. b) Corretta. Questa opzione descrive accuratamente come la QA supporta i tre modi di DevOps: Flusso : un approccio test-first contribuisce a un processo fluido ed efficiente. Ciclo di feedback : il testing automatizzato all'interno della pipeline di CI/CD supporta cicli di feedback rapidi. Apprendimento continuo : il quality hunting incoraggia i team a esplorare nuovi modi di testare e migliorare il software. c) Errata. Sebbene il testing manuale sia rilevante in DevOps, il testing manuale non può coprire tutto il codice e, per questo motivo, l'affermazione è errata. d) Corretta. Questa opzione descrive accuratamente come la QA supporta i tre modi di DevOps: Flusso : concentrandosi sul testing automatizzato, la QA supporta una consegna di valore prevedibile. Ciclo di feedback : utilizzando l'analisi statica, la QA fornisce feedback precoce sulla qualità del codice. Apprendimento continuo : il regression testing continuo fornisce garanzia sulla qualità del sistema. e) Errata. Questa opzione suggerisce erroneamente che ridurre al minimo il testing faccia parte del flusso e che affidarsi unicamente ai test svolti da uno sviluppatore sia sufficiente.	QDO-1.1.5	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
6	b	i è vera. Il syllabus afferma che esiste il rischio che DevOps sostituisca lo sviluppo software Agile invece di estenderlo. ii è falsa. DevOps rafforza la collaborazione all'interno dei team lungo tutto il ciclo di vita dello sviluppo software (SDLC). iii è vera, come da elenco dei benefici riportato nel syllabus. iv è vera. Implementare gli strumenti della pipeline di CI/CD non basta a produrre un cambiamento. È necessario un cambiamento culturale. v è falsa. L'obiettivo primario di DevOps è unificare sviluppo (Dev) e operations (Ops), non mantenere una separazione netta.	QDO-1.1.6	K2	1
7	b	a) Errata. I team cross-funzionali possiedono tutte le conoscenze e le competenze necessarie a svolgere qualsiasi compito comune per i team DevOps. Alcuni compiti specialistici che esulano dalle competenze del team possono essere assegnati a team specializzati all'interno dell'organizzazione. b) Corretta. Conoscenze e responsabilità sono condivise per assicurare che nessuna parte delle conoscenze e delle competenze resti in capo a un solo membro del team. c) Errata. La documentazione interna può supportare il tracciamento delle attività, ma non è sufficiente a ridurre al minimo l'impatto dell'indisponibilità di un membro del team. d) Errata. I membri di un team DevOps cross-funzionale possono ricoprire più ruoli contemporaneamente e possono assumersi qualsiasi compito per il quale abbiano le conoscenze necessarie.	QDO-1.2.1	K2	1
8	c	a) Errata. Migliorare l'estetica del sistema non è un obiettivo del Site Reliability Engineering (SRE). b) Errata. L'SRE enfatizza l'automazione e il continuous monitoring, anziché dare priorità al testing esplorativo. c) Corretta. L'SRE contribuisce a DevOps implementando cicli di feedback automatizzati per ottimizzare l'efficienza. d) Errata. L'SRE complementa DevOps colmando il divario tra sviluppo e operations. Non sostituisce le pratiche DevOps.	QDO-1.2.2	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
9	a, e	a) Corretta. Una politica della qualità e una politica di test sono i principali prodotti di lavoro per promuovere un'implementazione consistente di misure preventive come testing statico, automazione e pair programming, al fine di ottenere la built-in quality. b) Errata. Affidarsi al testing esplorativo in produzione ritarda il rilevamento dei difetti e contraddice l'attenzione di DevOps al testing precoce. c) Errata. Pur essendo utile per monitorare le prestazioni della pipeline di CI/CD, questa opzione non affronta direttamente l'integrazione della Quality Assurance (QA) né il miglioramento della built-in quality. Fornisce soltanto alla build artefatti e dati per produrre quality report. d) Errata. Il testing in produzione si concentra su test di qualità reattivi, mentre DevOps enfatizza attività di test proattive nelle fasi precoci del ciclo di vita dello sviluppo software (SDLC). e) Corretta. Monitoraggio e reporting integrati danno visibilità sulle metriche chiave di qualità e di business, assicurando quality report consistenti e supportando il miglioramento continuo in linea con i principi DevOps.	QDO-2.1.1	K3	2
10	c	a) Errata. Non esiste alcuna correlazione tra un ciclo di vita dello sviluppo software (SDLC) e l'uso del testing esplorativo o dell'automazione. b) Errata. DevOps enfatizza il continuous testing durante lo sviluppo, non soltanto dopo il deployment. Anche lo sviluppo software Agile integra il testing lungo tutti gli sprint. I modelli di sviluppo sequenziali non privilegiano il regression testing manuale nelle fasi iniziali. c) Corretta. In DevOps il testing è integrato in ogni fase dello sviluppo e del deployment, enfatizzando feedback continuo e qualità. Analogamente, lo sviluppo software Agile incorpora il testing lungo tutto il ciclo di sviluppo software e valorizza il feedback precoce e continuo. I modelli di sviluppo sequenziali, invece, organizzano il testing in livelli di test distinti. Il testing è gestito da team specializzati. d) Errata. DevOps sfrutta un'ampia automazione a supporto delle pipeline di CI/CD, pur consentendo il testing manuale in scenari specifici. Lo sviluppo software Agile non si affida unicamente ai test automatizzati.	QDO-2.1.2	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
11	c	a) Errata. Questa descrive soltanto l'aspetto di monitoraggio in produzione del continuous testing. Sebbene l'observability faccia parte dell'approccio, il continuous testing si estende a tutto l'SDLC, non solo alla produzione. b) Errata. Questa è in parte vera. Il syllabus sottolinea che il continuous testing comprende il regression testing, ma coinvolge anche testing precoce, collaborazione, miglioramento adattivo e monitoraggio in produzione. Non si limita al regression testing. c) Corretta. Questa riflette direttamente il syllabus: «Il continuous testing integra le attività di test lungo tutto l'SDLC ... verificando e validando il software in ogni fase». d) Errata. La risposta è incompleta, poiché integrare i test automatizzati nella pipeline di CI/CD è solo uno degli aspetti del continuous testing. Il continuous testing si estende a pianificazione, codifica, deployment e produzione.	QDO-2.1.3	K2	1
12	b	a) Errata. Una pull request (PR) non serve a limitare i contributi in base all'anzianità; le PR garantiscono qualità consistente attraverso review strutturate e test automatizzati per tutti gli sviluppatori. b) Corretta. Una PR impone controlli di qualità pre-merge consentendo review collaborative, attivando test automatizzati e incorporando il feedback prima di integrare i cambiamenti di codice. c) Errata. Una PR mira a impedire che codice di bassa qualità venga integrato imponendo dei test, non riducendo le review o affidandosi al testing in produzione. d) Errata. Una PR non si affida esclusivamente alle review manuali di un tester. Comporta collaborazione di team e automazione, per garantire efficienza e scalabilità.	QDO-2.1.4	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
13	a	a) Corretta. La continuous discovery si concentra sul comunicare con gli stakeholder per giungere a una comprensione condivisa delle nuove funzionalità. I prototipi di user experience (UX) aiutano a chiarire le esigenze degli stakeholder. b) Errata. Finalizzare i criteri di accettazione prima dell'esplorazione delle idee contraddice i principi della continuous discovery. Questa pratica è iterativa e dinamica, e consente aggiustamenti sulla base di esperimenti e feedback raccolti durante il processo di esplorazione. c) Errata. La continuous discovery non si concentra solo sulla fattibilità tecnica. Valuta anche il valore per il cliente, l'usabilità e l'allineamento con il business, assicurando che le soluzioni soddisfino obiettivi più ampi rispetto alla sola implementazione tecnica. d) Errata. La continuous discovery comporta la validazione di un'ampia gamma di assunzioni, compresi requisiti non-funzionali, valore per il cliente e impatto sul business, e non dei soli requisiti funzionali durante la fase di progettazione. Questo assicura un allineamento più completo con gli obiettivi.	QDO-2.2.1	K2	1
14	c	a) Errata. Come da Sezione 2.2.2 del syllabus: «Includere tutto nella gestione della configurazione (per esempio script di test automatizzati, script di deployment automatizzati, dati di test e configurazioni degli ambienti)». b) Errata. Come da Sezione 2.2.2 del syllabus: «Gli script di automazione dei passi di build o di test nella pipeline di CI/CD dovrebbero agevolare l'indagine in caso di failure della build». c) Corretta. I test dovrebbero attivarsi automaticamente a ogni commit sulla codebase condivisa. Richiedere un'approvazione manuale prima di eseguire i test crea un collo di bottiglia inutile e contraddice le pratiche di CI, orientate a feedback rapido e automazione. d) Errata. Come da Sezione 2.2.2 del syllabus: «I team DevOps effettuano il check-in di cambiamenti di codice piccoli e incrementali, e attendono che i test siano superati oppure seguono un approccio stop-and-fix se un test fallisce».	QDO-2.2.2	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
15	d	a) Errata. Come da Sezione 2.2.3 del syllabus: la qualità si ottiene con «l'automazione del deployment mediante strumenti di automazione, per consistenza, conformità e per fornire un audit trail». b) Errata. Come da Sezione 2.2.3 del syllabus: la qualità si ottiene con «un ambiente di test standardizzato e la gestione dei dati di test». c) Errata. Come da Sezione 2.2.3 del syllabus: la qualità si ottiene con infrastructure as code (IaC) «per gestire la configurazione dell'infrastruttura mediante codice, ai fini di consistenza e ripetibilità». d) Corretta. Consentire che i processi di deployment varino introduce inconsistenze e rischi, e contraddice l'enfasi della continuous delivery (CD) su automazione, ripetibilità e standardizzazione tra gli ambienti.	QDO-2.2.3	K2	1
16	a	Dark launch: una nuova funzionalità viene eseguita in produzione accanto alle funzionalità esistenti, il che consente monitoraggio, testing e analisi dell'impatto sul sistema senza esporla agli utenti. Pertanto, 1C. Rilascio blue-green: si utilizzano due ambienti identici — «quello attualmente in esercizio» e «la nuova versione» — per ridurre al minimo tempi di fermo e rischio, consentendo test aggiuntivi sulla «nuova versione» mentre «quello attuale» resta in esercizio. Pertanto, 2B. Canary release: un piccolo sottoinsieme di utenti viene esposto a un cambiamento software per validarne la qualità prima di estenderlo all'intera base di utenti. Pertanto, 3A. Testing A/B: varianti di una funzionalità vengono presentate a segmenti di utenti diversi per confrontare le metriche di prestazione e determinare quale versione, la più efficace, mantenere. Pertanto, 4D.	QDO-2.2.4	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
17	b	a) Errata: sebbene questo possa aiutare nel breve periodo, introduce passi manuali che riducono velocità, consistenza e scalabilità della pipeline di CI/CD. Va contro il principio della piena automazione nelle pratiche DevOps moderne. b) Corretta. Questa opzione affronta direttamente la root cause — ambienti e dati di test inconsistenti — raccomandando un potenziamento dell'automazione esistente. È in linea con le best practice di CI/CD, in cui ambienti automatizzati e riproducibili e il provisioning dei dati di test sono essenziali per affidabilità e Quality Assurance (QA). c) Errata. È una buona salvaguardia, ma non affronta il problema di fondo, cioè l'instabilità dell'ambiente. Saltare i deployment quando i test falliscono è reattivo. Non previene i failure causati da ambienti mal configurati o inconsistenti. d) Errata. Il deployment è legato alla build. Eseguire il deployment più spesso ridistribuirà gli stessi artefatti di build senza ricostruire effettivamente il codice sorgente, il che in questa situazione non è d'aiuto. Nemmeno eseguire build e deployment più spesso risolverà il problema, poiché il difetto risiede nella configurazione e nei dati di test dell'ambiente rilasciato.	QDO-2.2.6	K3	2
18	c	Sistema di gestione delle informazioni: fornisce la gestione del ciclo di vita degli elementi di backlog, quali requisiti e defect report. Pertanto, 1B. Version control system: supporta una gestione controllata dei cambiamenti per codice sorgente, codice di test automation e definizioni delle pipeline di CI/CD. Pertanto, 2A. Sistema di gestione degli artefatti: agevola la riutilizzabilità degli artefatti di build, delle dipendenze e dei dati di test. Pertanto, 3D. Sistema di observability: abilita tracciabilità e gestione degli accessi per monitoraggio, tracce e log. Pertanto, 4C.	QDO-3.1.1	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
19	b	a) Errata. La generazione di test case non fa parte della tracciabilità e non sostituisce la manutenzione dei collegamenti tracciabili tra la base di test e il testware. b) Corretta. L'automazione aiuta a mantenere accurate e aggiornate le relazioni tra requisiti, codice e testware, cosa essenziale per la tracciabilità. c) Errata. La tracciabilità è una responsabilità condivisa tra i ruoli DevOps, non dei soli sviluppatori. Automatizzare la tracciabilità non modifica in alcun modo tale responsabilità. d) Errata. L'automazione rafforza la collaborazione rendendo le relazioni tra i prodotti di lavoro trasparenti e verificabili in sede di audit.	QDO-3.1.2	K2	1
20	a	a) Corretta: la dashboard deve integrare i risultati dei test automatizzati e manuali per fornire una visione completa della qualità. Automatizzare la raccolta dei dati e fornire aggiornamenti in tempo reale migliora la visibilità e il processo decisionale. b) Errata: come da Sezione 3.1.3 del syllabus: «Creare una Single Source of Truth combinando le informazioni sulle metriche provenienti da fonti diverse in un unico punto accessibile». Archiviare separatamente i risultati dei test manuali frammenta il reporting della qualità e riduce la trasparenza. c) Errata: come da Sezione 3.1 del syllabus: «Il reporting della qualità integrato con la pipeline di CI/CD fornisce informazioni in tempo reale attraverso dashboard, combinando i risultati dei test automatizzati e manuali per un'analisi e un alerting completi». I soli test automatizzati non colgono interamente la qualità del software. Utilizzare soltanto strumenti di monitoraggio della produzione è insufficiente, poiché per un reporting completo occorre integrare più strumenti. d) Errata: i report statici non offrono aggiornamenti in tempo reale, il che li rende inefficaci per seguire le tendenze nella pipeline di CI/CD. Le dashboard consentono un monitoraggio continuo e un migliore allineamento tra i team.	QDO-3.1.3	K3	2

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
21	d	a) Errata: come da Sezione 3.1.4 del syllabus: «Subsetting dei dati di test: selezionare un sottoinsieme più piccolo e rappresentativo dei dati di produzione per il testing, mantenendo i riferimenti tra gli elementi di dati, così da ridurre il fabbisogno di spazio di archiviazione e accelerare il provisioning dei dati di test». b) Errata: come da Sezione 3.1.4 del syllabus: «Generazione dei dati: creare o aggiornare i dati di test per assicurare una copertura completa e una varietà di input (per esempio dati validi e non validi, edge case e valori limite)». c) Errata: come da Sezione 3.1.4 del syllabus: «Organizzazione dei dati: organizzare e archiviare i dati di test in modo strutturato in un VCS e renderli disponibili su richiesta all'interno della pipeline di CI/CD o per il testing manuale». d) Corretta: sebbene il masking dei dati di test riduca il rischio, non elimina completamente il rischio di violazioni della sicurezza. Un masking non corretto può comunque esporre dati sensibili.	QDO-3.1.4	K2	1
22	b	a) Errata. L'analisi statistica aiuta a individuare tendenze e pattern nella qualità del software e nei rischi residui. Inoltre, gli strumenti basati su AI possono prevedere i risultati futuri dei test. b) Corretta. Sebbene l'analisi statistica dei risultati dei test rafforzi l'automazione, l'interpretazione umana resta necessaria per validare le conclusioni, valutare il comportamento del software e rilevare problemi quali test difettosi e inefficienze di processo. c) Errata. Metriche quali densità dei difetti, copertura del codice e mean time between failures (MTBF) aiutano gli stakeholder a maturare fiducia nel valore di business del software, assicurando una migliore valutazione del rischio e un migliore processo decisionale. d) Errata. L'analisi statistica fornisce informazioni che danno un quadro più ampio della qualità del software, ma anche dei processi applicati e delle persone coinvolte.	QDO-3.1.5	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
23	c	a) Errata. Mantenere le risorse allocate in ogni momento comporta consumi superflui e costi più elevati, mentre gli ambienti controllati utilizzano le risorse solo quando servono. b) Errata. Assegnare una quantità fissa di risorse a tutti gli ambienti è inefficiente, poiché test diversi hanno requisiti diversi, con conseguente spreco di risorse o problemi di prestazione. c) Corretta. Gli ambienti di test controllati allocano le risorse dinamicamente, assicurando che ciascun test ottenga esattamente ciò di cui ha bisogno e ottimizzando costi e prestazioni. d) Errata. Allocare il massimo delle risorse disponibili a ogni test è dispendioso, poiché gli ambienti controllati ottimizzano l'utilizzo delle risorse in base ai requisiti di test specifici.	QDO-3.1.6	K2	1
24	b	a) Errata. Sebbene i test specifici di una funzionalità diventino regression test, eseguire tutti i regression test in ogni pipeline di CI/CD è dispendioso. L'ambito dei regression test andrebbe ottimizzato in base ai cambiamenti di codice e ai rischi. b) Corretta. I regression test sono una parte cruciale di una pipeline di CI/CD e il loro ambito andrebbe ottimizzato in base ai file modificati e ai rischi. Idealmente, tutti i regression test dovrebbero essere eseguiti prima di un release, ma spesso questo sarebbe negativo per la sostenibilità economica e operativa, poiché consuma risorse inutilmente: si esegue quindi una selezione dei regression test basata sul rischio. c) Errata. I regression test dovrebbero seguire i test di una funzionalità specifica, per confermare che nel software non siano stati introdotti nuovi difetti. d) Errata. I regression test sono una parte cruciale di una pipeline di CI/CD. Testare soltanto il codice nuovo non ne rivelerà l'impatto sul codice esistente.	QDO-3.2.1	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
25	b	a) Errata. Il performance testing è testing non-funzionale: non valida i contratti delle API, ma vi si appoggia dandoli per funzionanti. b) Corretta. Il testing di contratti determina che sia il provider sia il consumer aderiscano a una struttura, un formato e un insieme di aspettative definiti per l'API, relativamente a payload di richiesta e risposta, header, status code e tipi di dato. Può essere svolto senza rilasciare i componenti. c) Errata. Il testing di accettazione, che valida requisiti di alto livello in una fase avanzata del processo, non previene i difetti nell'API. d) Errata. Il testing di integrazione è utile, ma avviene più avanti nel ciclo di vita dello sviluppo software. Non fornisce la validazione precoce e isolata offerta dal testing di contratti, essenziale per prevenire i difetti prima dell'integrazione completa.	QDO-3.2.2	K3	2
26	c	iii e v sono specifiche dello sviluppo software Agile, in cui i test sono sviluppati lungo tutte le fasi dell'SDLC, il che consente loro di essere vicini allo sviluppo e di concentrarsi sugli unit test e sui test di integrazione dei componenti di livello più basso. i, ii e iv sono specifiche dei modelli di sviluppo sequenziali, in cui il testing automatizzato è svolto in una fase di test successiva al completamento dello sviluppo; offre un feedback limitato e richiede una pianificazione anticipata.	QDO-3.2.3	K2	1
27	a	a) Corretta. Riprodurre manualmente script di test automatizzati è l'attività meno preziosa, poiché il testing esplorativo può ottenere una validazione maggiore con minore impegno. b) Errata. Il testing esplorativo può estendere significativamente la copertura e aumentare la fiducia complessiva nel software. c) Errata. Le review del codice fanno parte del testing statico e sono un importante tipo di testing manuale per prevenire i difetti nel prodotto. d) Errata. Il testing di accessibilità può essere automatizzato solo in misura limitata, per cui nella maggior parte dei casi è necessario il testing manuale.	QDO-3.3.1	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
28	c	a) Errata. Sebbene i test esplorativi siano eseguiti manualmente, possono essere attivati dalla pipeline di CI/CD. b) Errata. I test esplorativi possono essere eseguiti in qualsiasi ambiente di test. c) Corretta. Gli strumenti possono supportare il testing esplorativo nella progettazione dei test charter, nella registrazione dei risultati dei test, nella segnalazione dei difetti e nell'archiviazione degli artefatti di test. d) Errata. I test esplorativi sono progettati, eseguiti e valutati in un'unica sessione di test, fatta eccezione per la pianificazione e la preparazione anticipate del test charter.	QDO-3.3.2	K2	1
29	d	a) Errata. Questo è un beneficio offerto dalla test automation e dalla tracciabilità, non dal crowd testing. b) Errata. Questo è un beneficio offerto dal testing di sicurezza, non dal crowd testing, perché il suo focus è sul testing di basso livello. c) Errata. Questo fa parte del release management e del monitoraggio della qualità, non del crowd testing. d) Corretta. Il crowd testing è orientato prima di tutto all'ampiezza (per esempio è utilizzato nel beta testing dopo una canary release), per raccogliere feedback dalla produzione e nel testing su dispositivi reali.	QDO-3.3.3	K2	1
30	b	a) Errata. Sebbene siano gli stakeholder a trarre i maggiori benefici da un evento di quality hunting, vi partecipa una più ampia varietà di ruoli, che va oltre i requisiti di business. b) Corretta. Il quality hunting è un evento in cui team, di dimensioni pari a circa metà di un team Agile, competono per fornire la valutazione più preziosa del livello di qualità del proprio oggetto di test. Questi team sono sfidati a creare approcci nuovi e innovativi per valutare la qualità. Gli stakeholder traggono beneficio dal ricevere molte idee e nuovi punti di vista sugli aspetti di qualità rilevanti del prodotto software. c) Errata. Non sono solo i tester, ma anche altri stakeholder, a partecipare e a estendere le informazioni ottenute dal testing mediante approcci innovativi. d) Errata. Il quality hunting, a differenza del bug hunting, consiste nel trovare difetti e nel valutare la qualità complessiva dell'oggetto di test.	QDO-3.3.4	K3	2

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
31	b	a) Errata. La generazione dei dati di test è una capability degli strumenti di gestione dei dati di test e di test management. I dati di test possono essere archiviati negli strumenti di gestione degli artefatti di build, ma questi non possono generarli. b) Corretta. La scansione alla ricerca di vulnerabilità è una capability chiave degli strumenti di gestione degli artefatti di build. c) Errata. Le pipeline di build automatizzate sono una capability degli strumenti di CI o CD. Gli output delle pipeline di CI/CD possono essere caricati nello strumento di gestione degli artefatti di build. d) Errata. La condivisione dei documenti è una capability degli strumenti di collaborazione e comunicazione. Gli strumenti di gestione degli artefatti di build trattano principalmente eseguibili compilati, librerie e altri deliverable, più che documenti di progetto.	QDO-4.1.1	K1	1
32	a	Gli strumenti di analisi statica del codice e di qualità del codice valutano il codice sorgente senza eseguirlo, contribuendo a far rispettare gli standard di codifica, a rilevare potenziali difetti e a individuare code smell o vulnerabilità di sicurezza. Pertanto, 1C. Gli strumenti di continuous testing abilitano test automatizzati in ogni fase dell'SDLC e a ogni livello di test, intercettando precocemente i difetti e riducendo al minimo le interruzioni. Pertanto, 2A. Gli strumenti di gestione della configurazione assicurano ambienti consistenti tra sviluppo, test e produzione automatizzando il provisioning e la gestione delle configurazioni. Pertanto, 3B. Gli strumenti di continuous monitoring forniscono informazioni in tempo reale sullo stato di salute del sistema e sul comportamento degli utenti, così che i difetti siano individuati e risolti rapidamente. Pertanto, 4D.	QDO-4.1.2	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
33	b	a) Errata. Sebbene il source code management organizzi i commit di codice e il version control, non supporta direttamente la gestione dello sviluppo delle funzionalità né la riduzione dei rischi di integrazione del codice. b) Corretta. Branch e merge supportano una gestione efficiente dello sviluppo delle funzionalità e dell'integrazione del codice mediante l'uso di feature branch e di review del codice per i cambiamenti di codice specifici di una funzionalità, il che contribuisce a ridurre i problemi di integrazione del codice e altri potenziali problemi successivi. c) Errata. Monitoraggio e feedback si concentrano sull'osservazione dello stato di salute e delle prestazioni dell'applicazione in produzione, non sulla gestione dello sviluppo delle funzionalità o dei processi di integrazione del codice. d) Errata. Sebbene il release management gestisca il coordinamento dei rilasci e le strategie di deployment, non riguarda specificamente la riduzione dei difetti di integrazione del codice durante lo sviluppo delle funzionalità.	QDO-4.2.1	K1	1
34	d	Le rolling release sostituiscono gradualmente le istanze del sistema con le versioni aggiornate, sostenendo la continuità di disponibilità del servizio. Pertanto, 1B. L'hotfix deployment è un release urgente per risolvere difetti critici in produzione, che privilegia la rapidità rispetto a un testing approfondito. Pertanto, 2C. Il dark launching rilascia le funzionalità nell'ambiente di produzione, ma esse non sono accessibili agli utenti. Questo consente ai team di testare e monitorare le prestazioni senza incidere sulla UX. Pertanto, 3A. Il phased rollout rilascia gli aggiornamenti in modo incrementale per regione, gruppo di utenti o altri criteri di segmentazione. Pertanto, 4D.	QDO-4.2.2	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
35	a, c	a) Corretta. L'idempotenza assicura che gli script IaC producano gli stessi risultati anche se applicati più volte, prevenendo deviazioni di configurazione. b) Errata. L'IaC è gestita mediante codice, non tramite interfacce grafiche. L'IaC è realizzata con un modello descrittivo, sfruttando le stesse pratiche di version control utilizzate dai team DevOps. c) Corretta. L'IaC utilizza approcci dichiarativi e imperativi. L'approccio dichiarativo si concentra sullo stato desiderato, mentre l'approccio imperativo specifica i passi per raggiungerlo. d) Errata. L'IaC riduce la necessità di intervento manuale automatizzando i cambiamenti dell'infrastruttura. L'IaC elimina la necessità di compiti manuali ripetitivi, assicurando consistenza e riducendo l'errore umano. e) Errata. L'IaC non elimina il version control. Si integra senza soluzione di continuità con i sistemi di gestione della configurazione, consentendo ai team di tracciare i cambiamenti, collaborare in modo efficiente e tornare a configurazioni precedenti in caso di necessità.	QDO-4.2.3	K2	1
36	c	a) Errata. I feature toggle consentono di modificare il comportamento del sistema senza cambiare il codice. b) Errata. I feature toggle offrono un controllo temporaneo, più che un'abilitazione permanente delle funzionalità. c) Corretta. I feature toggle sono un approccio che consente ai team di modificare il comportamento del sistema senza cambiare il codice. d) Errata. I feature toggle non sostituiscono le strategie di branch.	QDO-4.2.4	K1	1
37	a	Trunk-based development significa che ogni cambiamento di codice viene inviato direttamente al ramo principale dopo il superamento dei test durante il commit stage. Pertanto, 1C. I feature branch sono creati per cambiamenti di codice specifici ed eliminati dopo il merge. Pertanto, 2D. Una pull request (PR) integra cambiamenti di codice da un branch a un altro e consente ai team DevOps o ai colleghi di fornire feedback sui cambiamenti di codice. Pertanto, 3A. Un conflitto di merge si verifica quando, durante il merge, la stessa parte di codice deve essere modificata in entrambi i branch. Pertanto, 4B.	QDO-4.2.5	K1	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
38	c, e	a) Errata. Il chaos engineering si concentra sulla tolleranza ai guasti, non sull'efficienza del cloud. b) Errata. Il chaos engineering è reso possibile da una pipeline di CI/CD affidabile, che consente di rilasciare i cambiamenti in produzione e di effettuare il rollback senza difficoltà. c) Corretta. Il chaos engineering è cruciale per determinare la tolleranza ai guasti e l'affidabilità dei sistemi software, iniettando intenzionalmente failure e osservando come il sistema reagisce. d) Errata. Il chaos engineering comporta l'iniezione intenzionale di failure e l'osservazione di come il sistema reagisce. e) Corretta. I team DevOps utilizzano il chaos engineering per far emergere difetti nascosti, che potrebbero non manifestarsi durante il normale funzionamento o durante il testing in un ambiente di test.	QDO-4.2.6	K1	1
39	d	a) Errata. Una Software Bill of Materials (SBoM) comprende i dettagli delle condizioni di licenza dei componenti, non le licenze dell'intero sistema software. b) Errata. Le statistiche di utilizzo di CPU e memoria attengono alla telemetria per il monitoraggio dell'ambiente di produzione, non alla SBoM. c) Errata. L'automazione del deployment e i meccanismi di rollback fanno parte delle attività di continuous delivery e continuous deployment, non della SBoM. d) Corretta. La SBoM richiede le dipendenze del componente da altri componenti.	QDO-4.2.8	K1	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
40	b	a) Errata. La containerizzazione non elimina la necessità di misure di sicurezza; quando si utilizzano componenti di terze parti resta necessaria una diligente gestione delle dipendenze. b) Corretta. La containerizzazione consente ambienti consistenti tra sviluppo, test e produzione, risolvendo così problemi del tipo «sulla mia macchina funziona». c) Errata. La containerizzazione non sostituisce interamente le tecnologie di virtualizzazione tradizionali; è un'alternativa leggera utilizzata in condizioni specifiche. d) Errata. Sebbene i container siano efficienti e possano essere rimossi dopo l'uso, non garantiscono un utilizzo di risorse nullo quando sono inattivi.	QDO-4.2.9	K1	1

Appendice A – Chiave delle Risposte Aggiuntive

Numero della Domanda (#)	Risposta Corretta	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
A1		QDO-1.1.3	K2	1
A2	a, e	QDO-1.2.3	K2	1
A3	b	QDO-2.2.5	K2	1
A4		QDO-2.2.6	K3	2
A5		QDO-3.2.3	K2	1
A6	c	QDO-3.3.3	K2	1
A7	b	QDO-4.1.1	K1	1
A8	b	QDO-4.2.7	K1	1

Appendice B – Risposte Aggiuntive

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
A1		1. Il tempo intercorso tra un deployment fallito e il ritorno a un ambiente funzionante è detto tempo di recupero da un deploy fallito. 2. La percentuale di deployment in produzione non riusciti è detta change fail percentage. Entrambe le misurazioni sono legate alla qualità dei cambiamenti consegnati, che è detta stabilità. 3. La frequenza dei deployment in produzione è detta frequenza di deploy. 4. La durata che intercorre tra un commit di codice e un deployment in produzione riuscito è detta change lead time. Entrambe le misurazioni sono legate alla velocità con cui vengono introdotti i cambiamenti, che è detta throughput. Pertanto, 1 e 3 corrispondono a B) e 2 e 4 corrispondono ad A).	QDO-1.1.3	K2	1
A2	a, e	a) Corretta. DevOps come team di strumenti descrive un team separato che lavora sulla dotazione di strumenti necessaria a DevOps. b) Errata. Dev non ha bisogno di Ops non è presente in questo scenario. Gli sviluppatori continuano ad affidarsi alle Ops, lasciando loro interamente i compiti di deployment e manutenzione. c) Errata. Le responsabilità operative pienamente condivise promuovono la cooperazione tra i team Dev e Ops, e non sono un anti-pattern. d) Errata. Il team SRE è un modello di collaborazione valido, non un anti-pattern, e non è presente in questo caso. e) Corretta. I silos Dev e Ops sono presenti, poiché sviluppo e operations operano in isolamento senza collaborazione diretta.	QDO-1.2.3	K2	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
A3	b	a) Errata. Nel release on demand le funzionalità rilasciate in produzione non sono automaticamente visibili a tutti gli utenti. Meccanismi di rilascio controllato, quali gli strumenti di gestione delle funzionalità, consentono una visibilità selettiva in base alle esigenze di business. b) Corretta. Disaccoppiare i componenti architetturali consente di rilasciare componenti diversi in modo indipendente e con frequenze diverse, riducendo il rischio e sostenendo la flessibilità del release on demand. c) Errata. Sebbene prodotti di lavoro non software, quali manuali utente e materiali di marketing, facciano parte del release, l'approvazione manuale di tutti questi prodotti di lavoro non è obbligatoria prima di rilasciare le funzionalità. L'accento è posto sull'agilità e sulla consegna incrementale di valore. d) Errata. Il release on demand mira ad allineare i rilasci alle esigenze di business e dei clienti, per esempio ai tempi del mercato, anziché attenersi a scadenze fisse.	QDO-2.2.5	K2	1
A4		Come da Sezione 2.2.6 del syllabus: C - La definizione del flusso di informazioni e materiali fa parte della modellazione del Value Stream Mapping, passo #1 dell'implementazione della pipeline di CI/CD. Si veda la definizione di «value stream». D - Il provisioning riproducibile degli ambienti fa parte dell'automazione di build e deployment, passo #2 dell'implementazione della pipeline di CI/CD. B - Il testing di contratti fa parte del commit stage, passo #3 dell'implementazione della pipeline di CI/CD. E - Il performance testing per validare il comportamento del sistema può far parte dell'automazione dell'acceptance stage, passo #4 dell'implementazione della pipeline di CI/CD. A - I feature toggle fanno parte dell'automazione del release, passo #5 dell'implementazione della pipeline di CI/CD.	QDO-2.2.6	K3	2

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
A5		1, 2 e 4 sono specifici dei modelli di sviluppo sequenziali, in cui il testing automatizzato è svolto in una fase di test successiva al completamento dello sviluppo; offre un feedback limitato e richiede una pianificazione anticipata. 3 e 5 sono specifici dello sviluppo software Agile, in cui i test sono sviluppati lungo tutte le fasi dell'SDLC, il che consente loro di essere più vicini allo sviluppo e di concentrarsi sugli unit test e sui test di integrazione dei componenti di livello più basso.	QDO-3.2.3	K2	1
A6	c	a) Errata. Il crowd testing è testing manuale svolto da un gruppo di persone (crowd tester), di norma esterne all'organizzazione di sviluppo. b) Errata. Il crowd testing si concentra su un ambito di testing più ampio. Questa definizione si riferisce a un evento di quality hunting. c) Corretta. Il crowd testing si riferisce a una sessione in cui un gruppo di persone (crowd tester), di norma esterne all'organizzazione di sviluppo e possibilmente distribuite geograficamente, utilizza un test charter per guidare il proprio testing senza predisporre in anticipo test case dettagliati. d) Errata. Il crowd testing è un'attività di test manuale. Non comporta necessariamente test automatizzati.	QDO-3.3.3	K2	1
A7	b	a) Errata. Sebbene archiviare artefatti di build e dipendenze possa migliorare la consistenza, non affronta la root cause delle configurazioni inconsistenti tra gli ambienti. b) Corretta. Automatizzare la creazione degli ambienti, standardizzare le configurazioni e utilizzare strumenti IaC affronta il problema degli ambienti inconsistenti e sostiene un allestimento consistente tra gli stadi di sviluppo, test e produzione. c) Errata. Predisporre pipeline di CI/CD automatizzate si concentra sull'integrazione del codice e sul rilevamento dei difetti, anziché risolvere il difetto di configurazione degli ambienti. d) Errata. Svolgere load testing è utile per valutare le prestazioni del sistema, ma non risolve le inconsistenze di configurazione tra gli ambienti.	QDO-4.1.1	K1	1

Numero della Domanda (#)	Risposta Corretta	Spiegazione / Motivazione	Obiettivo di Apprendimento (LO)	Livello K	Numero di Punti
A8	b	a) Errata. Applicare il machine learning per rilevare le anomalie fa parte del monitoraggio, non della telemetria in sé. b) Corretta. La telemetria si concentra specificamente sulla raccolta di misurazioni delle caratteristiche di qualità funzionali e non-funzionali, per fornire informazioni sulle prestazioni del software e sull'interazione degli utenti. c) Errata. L'avvio di azioni di controllo è legato al continuous monitoring nel suo complesso, non specificamente alla telemetria. d) Errata. Questa descrive l'observability, che utilizza logging, tracing e metriche del comportamento del sistema per fornire una visione d'insieme della qualità del software.	QDO-4.2.7	K1	1