

Simulazione d'Esame - Risposte

Simulazione d'Esame set B

Versione 1.6

ISTQB® Certified Tester Syllabus Foundation Level

Compatibile con il Syllabus versione 4.0

International Software Testing Qualifications Board



Avviso di copyright

Avviso sul Copyright © International Software Testing Qualifications Board (di seguito chiamato ISTQB®).

ISTQB® è un marchio registrato di International Software Testing Qualifications Board. Tutti i diritti riservati.

Gli autori dichiarano con la presente di trasferire il copyright a International Software Testing Qualifications Board (ISTQB®). Gli autori (come attuali titolari del copyright) e ISTQB® (come futuro titolare del copyright) hanno concordato le seguenti condizioni di utilizzo:

Possono essere copiati estratti di questo documento per un uso non commerciale se la fonte è riconosciuta.

Qualsiasi azienda Training Provider accreditata alla Formazione può utilizzare questa Simulazione d'Esame nei loro corsi di Formazione se gli autori e ISTQB® sono riconosciuti come fonti e proprietari del copyright della Simulazione d'Esame, e a condizione che qualsiasi pubblicità di tale corso di formazione sia fatta dopo che l'accREDITAMENTO ufficiale dei materiali di formazione è stato ricevuto da un Member Board riconosciuto da ISTQB®.

Qualsiasi individuo o gruppo di individui può utilizzare questa Simulazione d'Esame, se gli autori e ISTQB® sono riconosciuti come fonti e possessori del copyright di questa Simulazione d'Esame.

E' proibito qualsiasi altro utilizzo di questa Simulazione d'Esame senza prima avere ottenuto l'approvazione scritta di ISTQB®.

Qualsiasi Member Board riconosciuto da ISTQB® può tradurre questa Simulazione d'Esame a condizione di riprodurre il sopra menzionato Avviso di Copyright nella versione tradotta della Simulazione d'Esame.

Responsabilità del documento

Exam Working Group ISTQB® è responsabile di questo documento.

Questo documento è mantenuto da un team dedicato ISTQB® che consiste nel Syllabus Working Group e nell'Exam Working Group.

Riconoscimenti

Questo documento è stato prodotto da un core team di ISTQB®: Stuart Reid e Adam Roman

Il core team ringrazia Exam Working Group, Syllabus Working Group e i Member Board per i loro suggerimenti e contributi.

Storia delle revisioni

Simulazione d'esame – Modello Layout Risposte utilizzato: Versione 2.11 Data: 16 ottobre 2023

Versione	Data	Osservazioni
1.6	8 luglio 2024	Versione allineata alla versione inglese V1.6 (8 luglio 2024). Correzione della domanda: #28, #32
1.4	12 Aprile 2024	Versione allineata alla versione inglese V1.4 (12 aprile 2024). Correzione della domanda: #2, #15
1.0	16 ottobre 2023	Prima versione

Indice dei contenuti

Avviso di copyright	2
Responsabilità del documento.....	2
Riconoscimenti	2
Storia delle revisioni	3
Indice dei contenuti	4
Introduzione	5
Scopo del documento	5
Istruzioni	5
Chiave di Risposta	6
Risposte	7
1	8
2	9
3	10
4	12
5	13
6	14
7	15
8	16
9	17
10	18
11	18
12	19
13	20
14	21
15	22
16	23
17	24
18	25
19	26
20	27
21	28
22	29
23	30
24	31
25	32
26	33
27	34
28	34
29	35
30	35
31	36
32	36
33	38
34	38
35	39
36	39
37	40
38	40
39	41
40	42

Introduzione

Scopo del documento

Le domande di questa simulazione, le risposte e le relative giustificazioni associate alle domande di questa simulazione d'esame sono state create da un team di esperti in materia e da scrittori esperti di domande, con l'obiettivo di:

- Supportare i Member Board e l'Exam Board ISTQB® nelle loro attività di scrittura delle domande.
- Fornire ai Training Provider e ai candidati all'esame esempi di domande d'esame.

Queste domande non possono essere utilizzate senza essere modificate in nessun esame ufficiale.

Si noti che gli esami reali possono includere un'ampia varietà di domande, e questa simulazione d'esame **non** intende includere esempi di tutti i possibili tipi, stili o lunghezze di domande; inoltre, questa simulazione d'esame potrebbe essere più difficile oppure meno difficile di qualsiasi esame ufficiale.

Istruzioni

In questo documento potete trovare:

- Tabella delle Risposte Chiave, includendo per ogni risposta corretta:
 - Livello di conoscenza (Livello K), obiettivo di apprendimento (LO) e valore del punteggio
- Insieme di risposte, includendo per tutte le domande:
 - Risposta corretta
 - Giustificazione per ogni opzione di risposta
 - Livello di conoscenza (Livello K), obiettivo di apprendimento (LO) e valore del punteggio
- Insieme di risposte aggiuntive, includendo per tutte le domande [non si applica a tutte le simulazioni d'esame]:
 - Risposta corretta
 - Giustificazione per ogni opzione di risposta
 - Livello di conoscenza (Livello K), obiettivo di apprendimento (LO) e valore del punteggio
- *Le domande sono contenute in un documento separato*

Chiave di Risposta

Numero domanda (#)	Risposta corretta	LO	Livello K	Punti
1	d	FL-1.2.1	K2	1
2	b	FL-1.2.2	K1	1
3	d	FL-1.3.1	K2	1
4	a	FL-1.4.1	K2	1
5	c	FL-1.4.2	K2	1
6	b	FL-1.4.4	K2	1
7	b	FL-1.5.1	K2	1
8	d	FL-1.5.2	K1	1
9	b	FL-2.1.1	K2	1
10	b	FL-2.1.2	K1	1
11	a	FL-2.1.3	K1	1
12	b	FL-2.1.4	K2	1
13	a	FL-2.2.1	K2	1
14	d	FL-2.2.3	K2	1
15	b	FL-3.1.2	K2	1
16	c	FL-3.2.1	K1	1
17	d	FL-3.2.2	K2	1
18	c	FL-3.2.3	K1	1
19	d	FL-4.1.1	K2	1
20	a	FL-4.2.1	K3	1

Numero domanda (#)	Risposta corretta	LO	Livello K	Punti
21	d	FL-4.2.2	K3	1
22	b	FL-4.2.3	K3	1
23	c	FL-4.2.4	K3	1
24	b	FL-4.3.1	K2	1
25	c	FL-4.3.2	K2	1
26	a, e	FL-4.4.2	K2	1
27	d	FL-4.4.3	K2	1
28	b	FL-4.5.2	K2	1
29	d	FL-4.5.3	K3	1
30	a	FL-5.1.3	K2	1
31	b	FL-5.1.4	K3	1
32	b	FL-5.1.5	K3	1
33	d	FL-5.1.7	K2	1
34	c	FL-5.2.4	K2	1
35	a	FL-5.3.1	K1	1
36	a	FL-5.3.3	K2	1
37	a	FL-5.4.1	K2	1
38	b	FL-5.5.1	K3	1
39	c	FL-6.1.1	K2	1
40	a	FL-6.2.1	K1	1

Risposte

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
1	d	<p>a) Non è corretta. Spesso è possibile utilizzare il testing dinamico per far fallire un oggetto di test in modi che non potrebbero mai essere ottenuti dagli utenti, ad esempio utilizzando il fault injection. Tuttavia, se il failure non può mai verificarsi con gli utenti finali reali, la sua identificazione non è particolarmente preziosa, in quanto il testing è finalizzato a migliorare il prodotto di lavoro per gli utenti finali. Passare il tempo ad eseguire il testing per failure che non possono verificarsi con gli utenti reali non è un utilizzo efficiente del tempo del tester</p> <p>b) Non è corretta. Il testing statico, sotto forma di analisi statica, viene utilizzato dagli sviluppatori per identificare i difetti nel loro codice del programma prima che possano essere identificati attraverso il testing dinamico. Si noti, tuttavia, che il testing statico (e l'analisi statica) viene utilizzato per rilevare difetti, non failure, che invece vengono rilevati dal testing dinamico. Pertanto, è l'uso del termine "failure" che rende questa opzione non corretta</p> <p>c) Non è corretta. L'analisi statica rileva direttamente le anomalie nel codice, che possono essere difetti, e questo è in genere fornito per lo sviluppatore, non per il cliente. Fornire le prove per il rilascio attraverso l'utilizzo dell'analisi statica degli elementi del sistema che non forniscono alcun output non ha senso</p> <p>d) È corretta. Le review sono una forma di testing statico che può essere applicata fin dall'inizio del ciclo di vita dello sviluppo software e sono utilizzate per rilevare i difetti che possono essere corretti prima che le attività successive di sviluppo perdano tempo ed effort per requisiti difettosi. Se i difetti non vengono rilevati e corretti il prima possibile, quando il difetto viene rilevato i prodotti di lavoro derivati, come la progettazione e il codice, dovranno essere modificati, perché si basavano su requisiti difettosi</p>	FL-1.2.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
2	b	<p>a) Non è corretta. Quality Assurance (QA) si focalizza sul miglioramento del processo e sull'implementazione del processo, utilizzando un approccio preventivo per evitare errori e difetti, mentre il testing è una forma di Quality Control (QC) che viene utilizzata per rilevare i difetti</p> <p>b) È corretta. QC mira ad ottenere livelli di qualità appropriati focalizzandosi sull'identificazione e correzione dei difetti di prodotto. Il testing è una parte importante del QC e aiuta a scoprire questi difetti</p> <p>c) Non è corretta. Sebbene il testing sia una parte significativa del QC e aiuti a scoprire i difetti, altre tecniche (non di test) utilizzate nel QC includono metodi formali come la verifica del modello, la prova di correttezza, la simulazione e la prototipazione</p> <p>d) Non è corretta. QA si concentra sul miglioramento del processo e sull'implementazione del processo, utilizzando un approccio preventivo per evitare errori e difetti, mentre il testing è una forma di QC che viene utilizzata per rilevare i difetti</p>	FL-1.2.2	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
3	d	<p>Il principio "il testing esaustivo è impossibile" è relativo al fatto che non è fattibile testare ogni possibile variazione degli input di test in tutte le diverse circostanze, ad eccezione dei casi triviali. Al contrario, il testing utilizza le tecniche di test, la prioritizzazione dei test case e il testing basato sul rischio per identificare un campione dall'insieme delle possibilità e focalizzare l'effort del testing.</p> <p>a) Non è corretta. Il principio afferma che non è possibile testare tutto, ad eccezione dei casi triviali. Testare tutto richiederebbe il testing di ogni possibile variazione degli <u>input</u> di test in tutte le diverse circostanze, che è generalmente inattuabile poiché esisterà un numero praticamente infinito. Testare ogni possibile <u>output</u> specificato non indirizzerà il problema, poiché la relazione tra gli input e gli output specificati può essere differente per ogni oggetto di test. Alcune volte, possono esistere un numero praticamente infinito di possibili output specificati (p.e. quando esistono differenti variabili che rappresentano numeri reali), mentre altre volte possono esistere solo due output specificati, come una singola variabile che può essere vera o falsa</p> <p>b) Non è corretta. Il principio afferma che non è possibile testare ogni possibile variazione degli input di test in tutte le diverse circostanze. Questo perché per i sistemi non triviali esiste un numero praticamente infinito. Pertanto, nella pratica, documentare tutte le possibili variazioni degli input di test sarebbe impraticabile, perché richiederebbe un tempo infinito</p>	FL-1.3.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		<p>c) Non è corretta. Iniziare il testing il prima possibile con le review e con altri approcci di testing statico non indirizza il problema del numero eccessivo di possibili test case. Il principio "il testing anticipato permette di risparmiare tempo e denaro" riguarda la correzione anticipata dei difetti per prevenire la presenza di difetti successivi nei prodotti di lavoro derivati, riducendo così i costi e la probabilità di failure</p> <p>d) È corretta. L'uso del partizionamento di equivalenza e dell'analisi ai valori limite per generare test case è un modo per indirizzare il principio, poiché queste tecniche di test forniscono un modo sistematico per derivare un sottoinsieme finito di tutti i possibili test case</p>			

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
4	a	<p>a) È corretta. La progettazione dei test prevede l'utilizzo delle <u>condizioni di test</u> per creare i <u>test case</u> e altro testware necessario, come i <u>requisiti dei dati di test</u> e i test charter per il testing esplorativo. Vengono inoltre specificati i <u>requisiti dell'ambiente di test</u>, che includono l'infrastruttura e gli strumenti necessari</p> <p>b) Non è corretta. L'esecuzione dei test prevede l'esecuzione dei test case (come parte delle procedure di test), ma non copre direttamente il restante testware menzionato nella domanda, come i requisiti dei dati di test, i requisiti dell'ambiente di test e le condizioni di test</p> <p>c) Non è corretta. L'analisi dei test viene utilizzata per identificare le funzionalità che devono essere testate. La base di test viene analizzata e definita come condizione di test, che viene poi prioritizzata insieme ai rischi correlati. Sebbene questa attività preveda di lavorare con le condizioni di test, non copre il restante testware menzionato nella domanda, come i requisiti dei dati di test, i requisiti dell'ambiente di test e i test case</p> <p>d) Non è corretta. L'implementazione dei test include la generazione delle procedure di test, come i test script manuali e automatizzati, che vengono creati a partire dai test case e possono essere organizzati in test suite. Le procedure di test vengono prioritizzate e organizzate in una schedulazione di esecuzione dei test. Vengono creati i dati di test, viene costruito l'ambiente di test e viene verificato il suo set up. Sebbene questa attività preveda esplicitamente di lavorare con i test case, e possa utilizzare i requisiti dei dati di test e i requisiti dell'ambiente di test per creare i dati di test e l'ambiente di test, non copre le condizioni di test</p>	FL-1.4.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
5	c	<p>a) Non è corretta. È improbabile che il team di marketing dell'organizzazione esegua molto testing (anche se in alcune organizzazioni può essere coinvolto nel testing di accettazione), quindi è improbabile che il loro livello medio di esperienza (per lo più nel marketing) abbia un impatto su come viene eseguito il testing per un determinato oggetto di test</p> <p>b) Non è corretta. È improbabile che il livello di conoscenza degli utenti sul fatto che si stia costruendo un nuovo sistema per loro possa influire su come venga eseguito il testing. Qualsiasi coinvolgimento degli utenti che potrebbe influenzare come il testing venga eseguito è più probabile che sia il risultato di decisioni prese dai tester, dal cliente e dal project manager</p> <p>c) È corretta. Il numero di anni di esperienza dei membri del team di performance testing aiuterà a determinare le capacità e le conoscenze (p.e. dei differenti strumenti e tipi di difetti) che i membri del team applicheranno durante il testing</p> <p>d) Non è corretta. La struttura organizzativa dei differenti utenti finali (che possono essere molti) cambierà da un utente all'altro. Quindi, può anche non essere nota quando l'applicazione viene testata, e la struttura organizzativa dell'utente finale può quindi avere poco effetto su come viene eseguito il testing</p>	FL-1.4.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
6	b	<p>a) Non è corretta. La tracciabilità tra i rischi mitigati e i test case superati fornisce poche informazioni, perché per essere mitigati (attraverso il testing) i rischi dovrebbero avere un corrispondente test case superato. Per poter valutare il rischio residuo, deve essere disponibile la tracciabilità tra tutti i rischi e i risultati dei test, in modo da poter identificare come rischi residui quei rischi che non hanno un corrispondente test superato</p> <p>b) È corretta. La tracciabilità tra i requisiti utente e i risultati dell'esecuzione dei test fornisce un'indicazione di quali requisiti utente sono stati testati e quindi fornisce un mezzo per misurare l'avanzamento del progetto (nel contesto del testing) rispetto agli obiettivi di business</p> <p>c) Non è corretta. Non è chiaro se i test case falliti forniscano un'indicazione delle competenze del tester più che dei test case superati. Questo dipenderebbe in parte dall'obiettivo del test (p.e. creare confidenza o causare failure). Inoltre, tale misura dei tester basata sui test case superati e falliti può essere controproducente, in quanto potrebbe indurre i tester a ottimizzare il loro testing in base a tale metrica piuttosto che all'obiettivo del test</p> <p>d) Non è corretta. La tracciabilità tra i rischi identificati e le condizioni di test specificate fornisce un mezzo per determinare quali ulteriori condizioni di test devono essere specificate. Determinare quali rischi valga la pena di testare fa parte del risk management e, in particolare, della mitigazione del rischio</p>	FL-1.4.4	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
7	b	<p>a) Non è corretta. Forti capacità comunicative, ascolto attivo (active listening) e capacità di lavoro di squadra consentono a un tester di interagire efficacemente con tutti gli stakeholder, ma una profonda conoscenza di una varietà di giochi per computer che gli consenta di andare d'accordo con uno sviluppatore non è un esempio di competenza generica utile ai tester</p> <p>b) È corretta. La conoscenza di dominio che può essere utilizzata per comprendere e comunicare con gli utenti finali e i rappresentanti di business è una delle competenze generiche richieste ai tester. Un tester con esperienza di pilota sarebbe in grado di apprezzare meglio i criteri di accettazione per il sistema di controllo dell'elicottero</p> <p>c) Non è corretta. Sebbene le competenze di programmazione possano essere considerate conoscenze tecniche che possono aumentare l'efficienza nell'utilizzo di alcuni strumenti di test, è improbabile che queste competenze migliorino la comunicazione con i business analyst</p> <p>d) Non è corretta. Sebbene la scrupolosità, l'attenzione ai dettagli, la curiosità e l'approccio metodico per identificare difetti difficili da rilevare siano tutte competenze generiche utili per i tester, è improbabile che i tester generino test case prima di iniziare il testing esplorativo. Infatti, uno dei punti fondamentali del testing esplorativo è che i test case sono generati durante il testing, non vengono progettati in anticipo</p>	FL-1.5.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
8	d	<p>a) Non è corretta. L'approccio whole-team consente a qualsiasi membro del team, con le competenze e le conoscenze necessarie, di svolgere qualsiasi compito, ma questo non significa che i membri del team possano assumere qualsiasi ruolo in qualsiasi momento. In genere, assumono solo i ruoli in cui sono competenti e non è detto che ogni membro del team possa svolgere ogni ruolo</p> <p>b) Non è corretta. L'approccio whole-team si applica su come lavora un singolo team (in genere nello sviluppo software Agile); non copre come più team si supponga lavorino su progetti più grandi e non suggerisce che sia necessario un solo "whole" team per un progetto completo</p> <p>c) Non è corretta. L'approccio whole-team non prevede che ogni membro del team sia coinvolto in ogni decisione importante. Ad esempio, non è necessario che il rappresentante di business (cioè il Product Owner) sia coinvolto in ogni decisione tecnica che non influisce sul risultato di business e l'implementazione di tale approccio rallenterebbe inutilmente i progressi del team</p> <p>d) È corretta. Sfruttando al meglio le diverse competenze di ogni membro del team, l'approccio whole-team favorisce dinamiche superiori del team, promuove una comunicazione e una collaborazione solide, e genera una sinergia del team che va a vantaggio dell'intero progetto</p>	FL-1.5.2	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
9	b	<p>a) Non è corretta. Nello sviluppo software Agile, i rilasci vengono prodotti in ogni iterazione e il rilascio frequente di incrementi richiede un regression testing completo. Anche se alcuni (o tutti) questi regression test possono essere automatizzati, il regression testing (automatizzato o non) non può essere sostituito dalla test automation del sistema</p> <p>b) È corretta. Se viene utilizzato un modello di sviluppo sequenziale, all'inizio del ciclo di vita il codice non è disponibile per l'esecuzione e quindi, durante questo periodo viene eseguito il testing statico (p.e. review). Nelle fasi successive nel ciclo di vita, quando il codice è disponibile per l'esecuzione, è possibile eseguire il testing dinamico. Si noti, tuttavia, che la preparazione per il testing dinamico avverrà spesso all'inizio di qualsiasi ciclo di vita dello sviluppo software</p> <p>c) Non è corretta. Se si utilizza un modello di sviluppo iterativo, come lo sviluppo software Agile, i test di componente possono essere ben utilizzati per il regression testing di ogni iterazione. In questo caso, esiste una forte argomentazione per l'automatizzazione di questi test di componente, che dovranno essere eseguiti frequentemente, mentre è improbabile che esista una forte argomentazione per gli sviluppatori che eseguono questi test di componente manualmente</p> <p>d) Non è corretta. Nella maggior parte dei modelli di sviluppo incrementale, i rilasci vengono prodotti in ogni incremento richiedendo, per ogni incremento rilasciato, sia testing statico sia testing dinamico a tutti i livelli di test</p>	FL-2.1.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
10	b	<p>a) Non è corretta. I tester dovrebbero eseguire la review dei prodotti di lavoro non appena sono disponibili le versioni draft, per consentire un testing anticipato come parte di un approccio shift-left. Se aspettassero fino alla successiva fase di sviluppo, si potrebbe iniziare un'attività di sviluppo (e di test) non necessaria su prodotti di lavoro non revisionati e difettosi</p> <p>b) È corretta. I tester dovrebbero eseguire la review dei prodotti di lavoro non appena sono disponibili le versioni draft, per consentire un testing anticipato come parte di un approccio shift-left</p> <p>c) Non è corretta. I tester generalmente eseguono la review dei prodotti di lavoro che costituiscono la base di test durante l'analisi dei test, non prima</p> <p>d) Non è corretta. I tester dovrebbero eseguire la review dei prodotti di lavoro non appena sono disponibili le versioni draft, per consentire un testing anticipato come parte di un approccio shift-left. Aspettare che vengano rilasciati significa che tutti i difetti che potrebbero essere rilevati dalla review dei tester saranno presenti nel documento rilasciato</p>	FL-2.1.2	K1	1
11	a	<p>a) È corretta. Test-Driven Development (TDD) è un esempio conosciuto di approccio test-first allo sviluppo</p> <p>b) Non è corretta. Coverage-Driven Development non è un esempio corretto di approccio test-first allo sviluppo</p> <p>c) Non è corretta. Quality-Driven Development non è un esempio corretto di approccio test-first allo sviluppo</p> <p>d) Non è corretta. Feature-Driven Development non è un esempio di approccio test-first allo sviluppo, ma è una metodologia di sviluppo software Agile basata sul rilascio di funzionalità (a differenza delle user story in Scrum)</p>	FL-2.1.3	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
12	b	<p>a) Non è corretta. DevOps migliora il testing in diversi modi, ad esempio fornendo un feedback rapido sulla qualità del codice, applicando un testing di regressione automatizzato che minimizza il rischio di regressione e promuovendo un approccio shift-left con codice di alta qualità e test di componente. Questo è in gran parte fornito attraverso la continuous integration, dove gli sviluppatori eseguono i test di componente (unit test) con il loro nuovo codice, che devono essere superati affinché il codice sia inserito nella build. Pertanto, gli sviluppatori devono completare il testing di componente</p> <p>b) È corretta. DevOps migliora il testing in diversi modi, ad esempio fornendo un feedback rapido sulla qualità del codice, applicando un testing di regressione automatizzato che minimizza il rischio di regressione e promuovendo un approccio shift-left con codice di alta qualità e test di componente</p> <p>c) Non è corretta. DevOps migliora il testing in diversi modi, ad esempio fornendo un feedback rapido sulla qualità del codice, applicando un testing di regressione automatizzato che minimizza il rischio di regressione e promuovendo un approccio shift-left con codice di alta qualità e test di componente. I tester non cercano di trattare gli sviluppatori e operations allo stesso modo dedicando più tempo al testing del rilascio, anche se è possibile utilizzare un approccio shift-right al testing (testing in produzione)</p> <p>d) Non è corretta. I processi automatizzati come il continuous integration/continuous delivery (CI/CD) in DevOps facilitano la creazione di ambienti di test stabili e riducono la necessità del testing manuale; tuttavia, esiste il rischio di trascurare l'importanza del testing manuale, soprattutto dal punto di vista dell'utente</p>	FL-2.1.4	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
13	a	<p>a) È corretta. Il testing di sistema esamina il comportamento e le capacità del sistema completo e copre il testing non-funzionale delle caratteristiche di qualità, che include il testing di sicurezza. Questo tipo di test viene spesso eseguito da un team di test indipendente sulla base delle specifiche di sistema</p> <p>b) Non è corretta. Il testing di integrazione dei sistemi esamina le interfacce con altri sistemi e con servizi esterni</p> <p>c) Non è corretta. Beta testing è un tipo di testing di accettazione eseguito in un luogo esterno da ruoli esterni all'organizzazione di sviluppo</p> <p>d) Non è corretta. Il testing di integrazione dei componenti comporta il testing delle (interfacce e) interazioni tra i componenti di un sistema, come l'interfaccia utente e il database</p>	FL-2.2.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
14	d	<p>a) Non è corretta. Il numero di regression test aumenta con l'avanzare del progetto, poiché i regression test sono generalmente richiesti quando vengono apportate modifiche al sistema. Allo stesso modo, anche il numero di test confermativi generalmente aumenta con l'avanzare del progetto, poiché sono necessari nuovi test confermativi per ogni correzione apportata al sistema</p> <p>b) Non è corretta. È il contrario. I test confermativi vengono creati ed eseguiti quando l'oggetto di test viene corretto, mentre i regression test vengono eseguiti (idealmente) ogni volta che l'oggetto di test viene migliorato (modificato)</p> <p>c) Non è corretta. Il testing confermativo verifica che un difetto sia stato risolto correttamente e quindi si occupa di testare le modifiche all'oggetto di test. Tuttavia, il regression testing assicura che le modifiche (comprese quelle all'ambiente operativo) non abbiano effetti negativi sul software non modificato e quindi non verifica che l'ambiente operativo rimanga invariato</p> <p>d) È corretta. Il regression testing assicura che le modifiche non abbiano effetti negativi sul software non modificato. Il testing confermativo verifica che un difetto sia stato corretto, e quindi riguarda il codice modificato</p>	FL-2.2.3	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
15	b	<p>a) Non è corretta. Una mancanza di usabilità dell'interfaccia utente può essere rilevata attraverso una review che utilizza una checklist adeguata, ma la mancanza di usabilità può essere identificata anche chiedendo a diversi tipici utenti di testare dinamicamente l'interfaccia utente e di fornire un feedback sulla sua usabilità</p> <p>b) È corretta. Una code review può rilevare il codice che non può essere raggiunto da nessun percorso, tuttavia i test dinamici possono solo esercitare il codice raggiungibile e non possono determinare che il codice non può essere raggiunto senza aver eseguito tutte le possibili combinazioni di input e stati di input, e questo è impraticabile per il codice reale</p> <p>c) Non è corretta. I tempi di risposta insufficienti per la maggior parte degli utenti previsti sono difficili da determinare senza eseguire il codice (cioè con il testing statico), quindi in questa situazione il testing dinamico potrebbe trovare un difetto, ma è improbabile che il testing statico ne trovi</p> <p>d) Non è corretta. Una code review da parte di qualcuno che conosce le funzionalità richieste potrebbe rilevare che le funzionalità richieste non sono state implementate nel codice, e anche il testing dinamico potrebbe essere utilizzato per determinare che queste funzionalità richieste non sono state implementate</p>	FL-3.1.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
16	c	<p>a) Non è corretta. Il feedback è degli stakeholder (p.e. rappresentanti di business, utenti finali) e non degli sviluppatori, quindi questo feedback non è probabilmente per informare i manager su quali sviluppatori siano più o meno produttivi</p> <p>b) Non è corretta. Il feedback anticipato e frequente degli stakeholder non viene utilizzato dai project manager per prioritizzare come interagiscono i diversi stakeholder</p> <p>c) È corretta. Ottenere il feedback anticipato e frequente degli stakeholder durante il processo di sviluppo software può essere molto vantaggioso, poiché facilita la comunicazione anticipata di potenziali problemi di qualità, può prevenire incomprensioni sui requisiti e assicura che qualsiasi modifica dei requisiti degli stakeholder sia compresa e implementata prima</p> <p>d) Non è corretta. Un feedback anticipato e frequente può prevenire lo sviluppo di un prodotto che non soddisfa le esigenze degli stakeholder, con conseguenti costosi rework e scadenze non rispettate; quindi, idealmente, non dovrebbero esistere ritardi. Inoltre, il feedback è degli stakeholder (e non su di loro), che includono gli utenti finali, quindi gli utenti finali che forniscono il loro feedback non aiuteranno la comprensione degli utenti finali</p>	FL-3.2.1	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
17	d	<p>Si consideri ognuna delle descrizioni dei compiti elencati:</p> <ol style="list-style-type: none"> 1. Vengono selezionate le caratteristiche di qualità da valutare e i criteri di uscita – (Pianificazione (C): Definire l'ambito della review, lo scopo, il prodotto di lavoro da sottoporre a review, le caratteristiche di qualità da valutare, le aree di interesse, i criteri di uscita, le informazioni di supporto come gli standard, l'effort e le tempistiche) 2. Viene distribuito il prodotto di lavoro ai partecipanti della review - (Inizio della review (B): Assicurarsi che tutti i partecipanti abbiano accesso al prodotto di lavoro e alle risorse necessarie, e chiarire i loro ruoli e responsabilità) 3. Vengono identificati i potenziali difetti nel prodotto di lavoro - (Review individuale (A): Valutare la qualità del prodotto di lavoro, identificare e memorizzare anomalie, raccomandazioni e domande utilizzando tecniche di review, come le review checklist-based e review scenario-based) 4. Vengono discussi i potenziali difetti (comunicazione e analisi (D): Analizzare e discutere ogni anomalia, determinarne il suo stato, il suo proprietario e le azioni richieste, e prendere decisioni della review, di solito durante un meeting). Questo potrebbe includere determinare la necessità di una review di follow-up) <p>Quindi:</p> <ol style="list-style-type: none"> 5. Non è corretta 6. Non è corretta 7. Non è corretta 8. È corretta. L'abbinamento corretto è: 1C, 2B, 3A, 4D 	FL-3.2.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
18	c	<p>Si consideri i ruoli elencati:</p> <ol style="list-style-type: none"> 1. Scribe (o Recorder) - responsabile di raccogliere i feedback dei reviewer e di documentare le informazioni della review, come le decisioni prese e le nuove anomalie individuate durante il review meeting. (Memorizzare le informazioni della review, come le decisioni e le nuove anomalie rilevate durante il review meeting - B) 2. Review Leader - responsabile di supervisionare il processo di review, come di selezionare i membri del team di review, di schedulare i review meeting e di assicurare che la review sia stata completata con successo. (Assumere la responsabilità generale della review, organizzando ad esempio quando e dove si svolgerà la review - D) 3. Facilitatore (o moderatore) - responsabile di assicurare che il review meeting sia svolto in modi efficace, includendo la gestione del tempo, la mediazione delle discussioni e la creazione di un ambiente sicuro in cui tutti possano esprimere liberamente le proprie opinioni. (Assicurare lo svolgimento efficace dei review meeting e la creazione di un ambiente di review sicuro - A) 4. Manager - responsabile di decidere cosa deve essere sottoposto a review e di allocare le risorse, come il personale e il tempo, per la review. (Decidere cosa deve essere sottoposto a review e assegnare le risorse, come lo staff e il tempo per la review - C) <p>Quindi:</p> <ol style="list-style-type: none"> a) Non è corretta b) Non è corretta c) È corretta. La corrispondenza corretta è: 1B, 2D, 3A, 4C d) Non è corretta 	FL-3.2.3	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
19	d	<p>a) Non è corretta. Il testing della tabella delle decisioni è una tecnica di test black-box e non white-box: i test case non sono basati sulle decisioni nel codice sorgente. Nel testing dei rami, i test case sono derivati dalla conoscenza del flusso di controllo dell'oggetto di test</p> <p>b) Non è corretta. L'anticipazione dei potenziali difetti viene utilizzata nella tecnica di error guessing (una tecnica di test basata sull'esperienza), non nel testing dei rami (una tecnica di test white-box). Nel testing della tabella delle decisioni, i test case sono derivati dalla specifica che descrive la logica di business</p> <p>c) Non è corretta. Se un test case si basa sulla conoscenza del flusso di controllo dell'oggetto di test, si tratta di una tecnica di test white-box. Il testing della tabella delle decisioni si basa tipicamente sull'analisi della logica di business, quindi è una tecnica di test black-box. Nel testing dei rami, i test case non sono derivati dalle specifiche, il che lo renderebbe una tecnica di test black-box. Il testing dei rami è una tecnica di test white-box, in cui i test case sono derivati dalla struttura del codice sorgente</p> <p>d) È corretta. Il testing della tabella delle decisioni è una tecnica di test black-box, quindi si basa sull'analisi del comportamento specificato dell'oggetto di test senza fare riferimento alla sua struttura interna. Pertanto, i test case sono indipendenti dal modo in cui il software viene implementato. Il testing dei rami è una tecnica di test white-box, quindi i test case si basano sull'analisi della struttura interna dell'oggetto di test. Poiché i test case dipendono dal modo in cui il software viene progettato, possono essere creati solo dopo la progettazione o l'implementazione dell'oggetto di test</p>	FL-4.1.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
20	a	<p>a) È corretta. 19 copre la partizione "nessuno sconto", 20 copre la partizione "50% di sconto" e 30 copre la partizione "10% di sconto". Questi tre valori coprono tutte e tre le partizioni di equivalenza valide</p> <p>b) Non è corretta. 11 e 12 coprono la partizione "nessuno sconto", 20 copre la partizione "50% di sconto". La partizione "10% di sconto" non è coperta, quindi complessivamente sono coperte due delle tre partizioni di equivalenza valide</p> <p>c) Non è corretta. 1 copre la partizione "nessuno sconto", 10 e 50 coprono la partizione "10% di sconto". La partizione "50% di sconto" non è coperta, quindi complessivamente sono coperte due delle tre partizioni di equivalenza valide</p> <p>d) Non è corretta. 29 e 31 coprono la partizione "nessuno sconto", 10 e 30 coprono la partizione "10% di sconto". La partizione "50% di sconto" non è coperta, quindi complessivamente sono coperte due delle tre partizioni di equivalenza valide</p>	FL-4.2.1	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
21	d	<p>Il dominio della lunghezza della password ha tre partizioni di equivalenza:</p> <ul style="list-style-type: none"> • password troppo corte {0, 1, ..., 4, 5} • password OK {6, 7, ..., 11, 12} • password troppo lunghe {13, 14, ...} <p>Per ottenere una copertura completa dell'analisi ai valori limite a 3 valori, è necessario testare i seguenti valori: 0, 1, 4, 5, 6, 7, 11, 12, 13, 14.</p> <p>Poiché l'analisi ai valori limite a 2 valori è già stata coperta, questo significa che abbiamo già testato le password di lunghezza: 0, 5, 6, 12 e 13.</p> <p>Quindi, le lunghezze aggiuntive che devono essere coperte per passare da una copertura completa dell'analisi ai valori limite a 2 valori a una copertura completa dell'analisi ai valori limite a 3 valori sono: 1, 4, 7, 11 e 14.</p> <p>Quindi:</p> <ol style="list-style-type: none"> a) Non è corretta b) Non è corretta c) Non è corretta d) È corretta 	FL-4.2.2	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
22	b	<p>Nella tabella delle decisioni esistono cinque colonne. Ogni test case copre una di queste.</p> <ul style="list-style-type: none">TC1 e TC2 coprono la Regola 4TC3 e TC4 coprono la Regola 2TC5 copre la Regola 5 <p>Questi cinque test case coprono tre delle cinque colonne, ottenendo una copertura di $(3/5) \cdot 100\% = 60\%$. Pertanto, l'opzione b) è quella CORRETTA.</p> <p>Quindi:</p> <ul style="list-style-type: none">a) Non è correttab) È correttac) Non è correttad) Non è corretta	FL-4.2.3	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
23	C	<div style="text-align: center;"> <pre> graph LR START((START)) -- "Add / N := 1 E1" --> NOT_FULL((NOT FULL)) NOT_FULL -- "Add [N < 2] / N := N + 1 E2" --> NOT_FULL NOT_FULL -- "Add [N = 2] / N := N + 1 E4" --> FULL((FULL)) FULL -- "Remove / N := N - 1 E5" --> NOT_FULL NOT_FULL -- "Remove [N > 0] / N := N - 1 E3" --> NOT_FULL </pre> </div> <p>Le transizioni sono identificate con E1, ..., E5 come in figura. La variabile N indica il numero di elementi attualmente memorizzati. Ogni evento "Add" aumenta di 1 il valore N, e ogni evento "Remove" diminuisce di 1 il valore N. Si noti che quando si verifica l'evento "Add" nello stato NOT FULL, lo stato passa a FULL solo se N=2. Se N<2, il sistema rimane nello stato NOT FULL. Se N=0, non è possibile eseguire alcuna azione "Remove". Se N=3, non è possibile eseguire alcuna azione "Add".</p> <ul style="list-style-type: none"> • Il test a) può essere scritto come E1, E3, E3, E2, E4 (quindi copre 4 delle 5 transizioni valide, ottenendo una copertura dell'80% delle transizioni valide). • Il test b) non è fattibile, poiché dopo le prime tre azioni "Add" il sistema si trova nello stato FULL e non esiste una transizione valida attivata dall'evento "Add" per uscire da FULL. Dopo le prime tre transizioni si raggiunge solo il 60% di copertura delle transizioni valide. • Il test c) può essere scritto come E1, E2, E4, E5, E3 (quindi copre 5 transizioni valide su 5, ottenendo il 100% di copertura delle transizioni valide). 	FL-4.2.4	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		<ul style="list-style-type: none"> • Il test d) può essere scritto come E1, E2, E4, E5, E4 (quindi copre 4 transizioni valide su 5, ottenendo una copertura dell'80% delle transizioni valide). <p>Quindi:</p> <ul style="list-style-type: none"> a) Non è corretta b) Non è corretta c) È corretta d) Non è corretta 			
24	b	<ul style="list-style-type: none"> a) Non è corretta. La copertura è sempre definita come percentuale degli elementi coperti. Pertanto, non può superare il 100% b) È corretta. Se le istruzioni eseguite da T1 e T2 fossero disgiunte, la copertura della test suite {T1, T2} sarebbe del 105%, il che è impossibile (si veda la risposta a). Pertanto, almeno il 5% delle istruzioni eseguibili deve essere stato eseguito sia da T1 sia da T2 c) Non è corretta. La copertura delle istruzioni non dice nulla sul numero di istruzioni non eseguibili nel codice d) Non è corretta. Anche se una test suite raggiunge la copertura completa delle istruzioni, questo non implica il raggiungimento della copertura completa dei rami 	FL-4.3.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
25	c	<p>Il testing dei rami è una tecnica di test white-box in cui gli elementi di copertura sono rami. Un ramo è un trasferimento di controllo tra due nodi nel grafo del flusso di controllo, che visualizza le possibili sequenze in cui le istruzioni del codice sorgente vengono eseguite nell'oggetto di test. Ogni trasferimento di controllo può essere non condizionale (cioè, codice lineare) o condizionale (cioè, un risultato decisionale). La copertura viene misurata come il numero di rami esercitati dai test case diviso per il numero totale di rami ed è espressa come percentuale.</p> <p>Quindi:</p> <ul style="list-style-type: none">a) Non è corretta. Un esito decisionale è un ramo condizionale. Per il testing dei rami, X conta non solo i rami condizionali, ma anche quelli non condizionalib) Non è corretta. La copertura dei rami conta non solo i rami condizionali, ma anche quelli non condizionalic) È corretta. La copertura dei rami è misurata come il numero di rami esercitati dai test case diviso per il numero totale di rami ed è espressa in percentualed) Non è corretta. Sia X che Y contano solo i rami condizionali e non tengono conto dei rami non condizionali	FL-4.3.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
26	a, e	<p>Il testing esplorativo è utile quando le specifiche sono poche o inadeguate, oppure quando esiste una forte pressione dei tempi per il testing. Il testing esplorativo è utile anche per integrare altre tecniche di test più formali. Il testing esplorativo sarà più efficace se il tester è esperto, ha una conoscenza del dominio e possiede un alto grado di competenze fondamentali, come capacità analitiche, curiosità e creatività.</p> <p>Quindi:</p> <ul style="list-style-type: none"> a) È corretta. Il testing esplorativo è utile quando le specifiche sono poche o inadeguate o quando il tempo a disposizione per il testing è molto limitato b) Non è corretta. Il testing esplorativo non è una tecnica di test black-box c) Non è corretta. Il testing esplorativo è utile quando le specifiche sono scritte in modo inadeguato d) Non è corretta. In linea di principio, le competenze di programmazione non hanno nulla a che fare con il testing esplorativo e) È corretta. Il testing esplorativo sarà più efficace se il tester è esperto, ha una conoscenza del dominio e possiede un alto grado di competenze fondamentali, come capacità analitiche, curiosità e creatività 	FL-4.4.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
27	d	a) Non è corretta. Le checklist dovrebbero contenere le condizioni di test da verificare. Questo è un esempio di errore, non di una condizione di test; anche se il tester è stato in grado di dedurre alcune potenziali condizioni di test dagli esempi di errore, questa descrizione di errore è troppo generica b) Non è corretta. Le checklist non dovrebbero contenere elementi che sono più adatti come criteri di uscita. Questo è un esempio di criterio di uscita c) Non è corretta. Le checklist non dovrebbero contenere elementi troppo generici. Questo è un elemento molto generico, che praticamente descrive l'obiettivo del testing d) È corretta. Questo è un esempio di condizione di test che può essere verificata da una persona	FL-4.4.3	K2	1
28	b	a) Non è corretta. Il formato rule-oriented include formati come liste di verifica con punti elenco o forme tabellari del mapping input-output, che visualizzano in modo esplicito le regole da seguire. Given/When/Then è un formato scenario-oriented perché descrive uno scenario da verificare b) È corretta. Si tratta di un formato Given/When/Then, che è scenario-oriented c) Non è corretta. Non esiste un formato "product-oriented" dei criteri di accettazione d) Non è corretta. Non esiste un formato "process-oriented" dei criteri di accettazione	FL-4.5.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
29	d	a) Non è corretta. Il test case è relativo alla visualizzazione degli ordini precedenti nella cronologia degli ordini b) Non è corretta. Il test case è relativo alla visualizzazione degli ordini precedenti c) Non è corretta. Il test case è relativo alla visualizzazione degli ordini precedenti nella cronologia degli ordini d) È corretta. Il test case è relativo al processo di registrazione, che non viene trattato nella user story. La user story è relativa alla visualizzazione degli ordini precedenti	FL-4.5.3	K3	1
30	a	a) È corretta. Questo criterio può (e dovrebbe) essere controllato <i>prima</i> che il codice venga sottoposto al controllo della versione b) Non è corretta. Questo criterio può essere controllato <i>dopo</i> che la fase (2) è stata eseguita, perché la segnalazione dei conflitti durante il merge può essere fatta <i>dopo</i> che il codice è stato sottomesso ed eseguito il merge c) Non è corretta. Questo criterio è più adatto come criterio di ingresso per la fase (3) d) Non è corretta. Questo criterio è più adatto come criterio di uscita per la fase (3)	FL-5.1.3	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
31	b	<p>L'effort medio di sviluppo calcolato sui quattro progetti è $\\$900.000 = (800.000 + 1.200.000 + 600.000 + 1.000.000) / 4$ e l'effort medio del testing è $\\$90.000 = (40.000 + 130.000 + 70.000 + 120.000) / 4$. Il rapporto tra effort medio del testing e effort medio di sviluppo è di 1:10 ($90.000 / 900.000$), il che significa che storicamente, in media, l'effort di testing è pari al 10% dell'effort di sviluppo.</p> <p>Quindi, se l'effort di sviluppo è stimato $\\$800.000$, l'effort del testing viene stimato come:</p> $10\% * \$800,000 = 0.1 * \$800,000 = \$80,000.$ <p>Quindi:</p> <ul style="list-style-type: none"> a) Non è corretta b) È corretta c) Non è corretta d) Non è corretta 	FL-5.1.4	K3	1
32	b	<p>Dipendenze logiche significa che per ogni prodotto deve essere eseguita RICERCA → VISUALIZZARE → AGGIUNGERE prima di eseguire ORDINE. E' possibile aggiungere più prodotti (utilizzando lo stesso flusso) prima di eseguire ORDINE. In base a questo TC1 o TC2 deve essere eseguito per primo, altrimenti nessun avanzamento può essere possibile. La prima priorità dovrebbe essere data a VISUALIZZARE e AGGIUNGERE il prodotto B poiché i suoi test case (TC6, TC4) hanno priorità più alta. Quindi i primi 3 test case eseguiti sono TC2 → TC4 → TC6.</p> <p>Ora deve essere verificato se eseguire TC7 e successivamente l'intero flusso del prodotto A, oppure eseguire prima i test case del prodotto A. Poiché TC7 ha priorità più bassa degli altri test, gli altri test dovrebbero essere eseguiti prima. Quindi, l'ordine dovrebbe essere: TC2, TC4, TC6, TC1, TC3, TC5, TC7.</p>	FL-5.1.5	K3	1

		<p>Quindi:</p> <ul style="list-style-type: none">a) Non è corretta. TC1 deve essere eseguito prima di TC3b) È correttac) Non è corretta. TC7 è l'ultimo che deve essere eseguitod) Non è corretta. Il prodotto B deve essere eseguito prima del prodotto A			
--	--	---	--	--	--

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
33	d	a) Non è corretta. Il testing di usabilità è un testing orientato al business che critica il prodotto (Q3) b) Non è corretta. Il testing funzionale è un testing orientato al business che supporta il team di sviluppo (Q2) c) Non è corretta. Lo User Acceptance Testing è un testing orientato al business che critica il prodotto (Q3) d) È corretta. Il testing di integrazione dei componenti è un testing orientato alla tecnologia che supporta il team di sviluppo (Q1)	FL-5.1.7	K2	1
34	c	Si consideri ognuno dei rischi elencati e le relative azioni di mitigazione: 1. Le risposte di sistema molto lente (1) possono essere testate durante il performance testing (B) 2. I cambiamenti delle preferenze dei consumatori (2) sono di solito fuori dal nostro controllo, quindi di solito questo rischio viene accettato (A) 3. L'allagamento della sala server (3) può causare perdite significative, quindi si dovrebbe trasferire il rischio, ad esempio, acquistando una polizza assicurativa (D) 4. Il fatto che i pazienti al di sopra di una certa età ricevano report non accurati (4) suggerisce un potenziale problema di valori limite, che può essere efficacemente rilevato con tecniche come l'analisi ai valori limite (C) Quindi: a) Non è corretta b) Non è corretta c) È corretta. La corrispondenza corretta tra rischio e mitigazione è: 1B, 2A, 3D e 4C d) Non è corretta	FL-5.2.4	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
35	a	a) È corretta. Le metriche di qualità del prodotto misurano le caratteristiche di qualità. Il mean time to failure misura la maturità, quindi è una metrica di qualità del prodotto b) Non è corretta. Questo è un esempio di metrica dei difetti, non di metrica di qualità del prodotto c) Non è corretta. Questo è un esempio di metrica di copertura, non di metrica di qualità del prodotto d) Non è corretta. Questo è un esempio di metrica dei difetti, non di metrica di qualità del prodotto	FL-5.3.1	K1	1
36	a	a) È corretta. Il cliente si trova in una località e un fuso orario differenti, quindi potrebbe essere difficile comunicare face-to-face b) Non è corretta. Le dashboard sono generalmente disponibili per qualsiasi utente in qualsiasi momento, quindi la differenza di fuso orario non sarà un ostacolo alla comunicazione come la comunicazione verbale, face-to-face c) Non è corretta. Anche se la differenza di fuso orario tra Europa e Nord America è di diverse ore, e questo può causare qualche inconveniente, non è certamente così grande come nel caso di una comunicazione face-to-face d) Non è corretta. Gli strumenti di videoconferenza sono un mezzo di comunicazione conveniente. Sebbene la comunicazione tra Europa e Nord America durante l'orario di lavoro richieda di solito che una delle parti si colleghi nelle prime ore del mattino o nelle ore più tarde, questo non è un inconveniente come la comunicazione verbale, face-to-face	FL-5.3.3	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
37	a	a) È corretta. Per un elemento di configurazione complesso (p.e. un ambiente di test), lo strumento di Configuration Management (CM) memorizza gli elementi che lo compongono, le loro relazioni e le versioni b) Non è corretta. Gli strumenti di CM non eseguono i test case e non calcolano la copertura c) Non è corretta. Uno strumento di CM non è uno strumento di gestione delle licenze d) Non è corretta. Gli strumenti di CM non generano dati di test	FL-5.4.1	K2	1
38	b	a) Non è corretta. Sebbene la frase sia vera, non fornisce abbastanza valore per lo sviluppatore b) È corretta. Dai risultati dei test sembra che il sistema ignori i duplicati e ordini la lista ignorando le ripetizioni. Questa è probabilmente la causa dei failure in TC3, TC4, TC5. Queste informazioni possono aiutare lo sviluppatore a rilevare il difetto e a correggerlo in modo più efficiente c) Non è corretta. Il sistema non fallisce nell'ordinare i numeri negativi. Il problema è piuttosto quello di non considerare i duplicati d) Non è corretta. I test case TC3, TC4 e TC5 falliscono, ma non si ha conoscenza che i test case abbiano dei difetti	FL-5.5.1	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
39	c	<p>Si consideri ognuna delle categorie di strumenti elencati e le relative descrizioni:</p> <ul style="list-style-type: none"> A. Strumenti di testing statico - supportano il tester nell'esecuzione di review e analisi statiche (4) B. Strumenti che supportano la scalabilità e la standardizzazione del rilascio - per esempio, virtual machine, strumenti container (3) C. Strumenti DevOps - supportano la pipeline di delivery DevOps, il tracciamento del flusso di lavoro, il processo(i) di build automatizzato, la continuous integration/continuous delivery (CI/CD) (1) D. Strumenti di collaborazione - facilitano la comunicazione (2) <p>Quindi:</p> <ul style="list-style-type: none"> a) Non è corretta b) Non è corretta c) È corretta. L'abbinamento corretto è: 1C, 2D, 3B, 4A d) Non è corretta 	FL-6.1.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
40	a	<p>a) È corretta. La test automation può fornire misure troppo complicate da ricavare per l'uomo, come le misure di copertura dei test white-box per tutto il codice, tranne quello più banale</p> <p>b) Non è corretta. Utilizzando gli strumenti di test, la responsabilità del testing NON è condivisa con il fornitore dello strumento, poiché il fornitore non è coinvolto nel testing ed è responsabilità del tester. L'unica possibile responsabilità che potrebbe essere assegnata al fornitore dello strumento è se lo strumento non funziona come previsto e fornisce risultati di test errati</p> <p>c) Non è corretta. I tester devono comunque esercitare un pensiero critico quando analizzano le anomalie nei risultati dei test, per determinarne la probabile causa</p> <p>d) Non è corretta. Né i tester né gli strumenti possono generare test case semplicemente dall'analisi del codice del programma, poiché il codice è l'implementazione e non fornisce informazioni sui risultati attesi, che dovranno essere generati a partire da un'altra base di test, come le specifiche di progettazione</p>	FL-6.2.1	K1	1