Simulazione d’esame - Risposte

Simulazione d'esame set B
Versione 1.0

ISTQB® Certified Tester Syllabus Foundation Level

Compatibile con il Syllabus versione 4.0

International Software Testing Qualifications Board

© International Software Testing Qualifications Board

Avviso di copyright

Copyright Notice © International Software Testing Qualifications Board (di seguito denominato ISTQB®).

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

Tutti i diritti riservati.

Con la presente gli autori trasferiscono il copyright a ISTQB®. Gli autori (in quanto attuali detentori del copyright) e ISTQB® (in quanto futuro detentore del copyright) hanno concordato le seguenti condizioni di utilizzo:

Estratti, per uso non commerciale, da questo documento possono essere copiati se la fonte viene riconosciuta.

Qualsiasi Training Provider accreditato può utilizzare questa simulazione d'esame nel suo corso di formazione, a condizione che gli autori e ISTQB® siano riconosciuti come fonte e proprietari del copyright della simulazione d'esame e a condizione che qualsiasi pubblicità di tale corso di formazione sia fatta solo dopo che l'accreditamento ufficiale del materiale di formazione sia stato ricevuto da un ISTQB® Member Board riconosciuto.

Qualsiasi individuo o gruppo di individui può utilizzare questa simulazione d'esame in articoli e libri, a condizione che gli autori e ISTQB® siano riconosciuti come fonte e proprietari del copyright della simulazione d'esame.

Qualsiasi altro uso di questa simulazione d'esame è proibito senza aver prima ottenuto l'approvazione scritta dell'ISTQB®.

Qualsiasi ISTQB® Member Board riconosciuto può tradurre questa simulazione d'esame a condizione che riproduca la suddetta nota di copyright nella versione tradotta della simulazione d'esame.

Responsabilità del documento

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

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

Riconoscimenti

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

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

© International Software Testing Qualifications Board

Storia delle revisioni

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

Versione	Data	Osservazioni
1.0	16 ottobre 2023	Prima versione

© International Software Testing Qualifications Board

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 26
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 37
34 37
35 38
36 38
37 39
38 39
39 40
40 41

© International Software Testing Qualifications Board

Introduzione

Scopo del documento

Le domande e le risposte della simulazione e le relative giustificazioni contenute in questa simulazione d'esame sono state create da un team di esperti in materia e di redattori di domande con esperienza con l'obiettivo di:

- Assistere i Member Board e gli Exam Board di 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 tali e quali 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 può essere sia più difficile che meno difficile di qualsiasi esame ufficiale.

Istruzioni

In questo documento troverete:

- Tabella delle Chiavi di Risposta, inclusi per ogni risposta corretta:
  - Livello K, Learning Objective o obiettivo di apprendimento e valore del punteggio
- Set di risposte, inclusi per tutte le domande:
  - Risposta corretta
  - Giustificazione per ogni opzione di risposta
  - Livello K, Learning Objective o obiettivo di apprendimento e valore del punteggio
- Set di risposte aggiuntive, inclusi per tutte le domande [non si applica a tutte le simulazioni d'esame]:
  - Risposta corretta
  - Giustificazione per ogni opzione di risposta
  - Livello K, Learning Objective o obiettivo di apprendimento e valore del punteggio

- *Le domande sono contenute in un documento separato*

© International Software Testing Qualifications Board

Chiave di Risposta

Numero di 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 di 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	a	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

© International Software Testing Qualifications Board

Risposte

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
1	d	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 raggiunti 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 a testare failure che non possono verificarsi con gli utenti reali non è un uso efficiente del tempo del tester 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 di quanto si possa ottenere con il testing dinamico. Si noti, tuttavia, che il testing statico (e l'analisi statica) viene utilizzato per individuare i difetti, non i failure, che invece vengono individuati dal testing dinamico. Pertanto, è l'uso del termine "failure" che rende questa opzione non corretta c) Non è corretta. L'analisi statica rileva direttamente le anomalie nel codice, che possono essere difetti, e questo è in genere per lo sviluppatore, non per il cliente. La fornitura di prove per il rilascio mediante l'uso dell'analisi statica di elementi che non forniscono alcun risultato è insensata d) È corretta. Le review sono una forma di testing statico che può essere applicato fin dall'inizio del ciclo di vita dello sviluppo del software e servono a trovare i difetti che possono essere rimossi prima che le attività di sviluppo successive sprechino effort per requisiti difettosi. Se i difetti non vengono individuati e rimossi tempestivamente, quando il difetto viene individuato i prodotti di lavoro derivati, come la progettazione e il codice, dovranno essere modificati perché si basavano su requisiti difettosi	FL-1.2.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
2	b	a) Non è corretta. L'affermazione "La QA viene eseguita come parte del testing" non è corretta perché la QA si concentra sul miglioramento 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 b) È corretta. L'affermazione "Il testing viene eseguito come parte del QC" è corretta perché il QC mira a raggiungere livelli di qualità appropriati concentrandosi sull'identificazione e la correzione dei difetti del prodotto. Il testing è una parte importante del QC e aiuta a scoprire questi difetti c) Non è corretta. L'affermazione "Il testing è un altro termine per indicare il QC" non è corretta perché, sebbene il testing sia una parte importante del QC e aiuti a scoprire i difetti, altre tecniche (non di test) utilizzate nel QC includono metodi formali come la verifica del modello e la prova di correttezza, nonché la simulazione e la prototipazione d) Non è corretta. L'affermazione "Il testing viene eseguito come parte della QA" non è corretta perché la QA si concentra sul miglioramento e sull'implementazione del processo, utilizzando un approccio preventivo per evitare errori e difetti, mentre il testing è una forma di QC che viene utilizzato per rilevare i difetti	FL-1.2.2	K1	1

© International Software Testing Qualifications Board

3	d	Il principio "il testing esaustivo è impossibile" riguarda il fatto che non è possibile testare ogni possibile variazione degli input di test in tutte le diverse circostanze, tranne in casi banali. Al contrario, il testing utilizza tecniche di test, la prioritizzazione dei test case e il testing basato sul rischio per campionare l'insieme delle possibilità e concentrare gli effort del testing. a) Non è corretta. Il principio afferma che non è possibile testare tutto, tranne che in casi banali. Testare tutto richiederebbe di testare ogni possibile variazione degli input di test in tutte le diverse circostanze, il che è generalmente inattuabile poiché ce ne sarà un numero praticamente infinito. Testare ogni possibile output specificato non risolverà il problema, poiché la relazione tra gli input e gli output specificati può essere diversa per ogni oggetto di test. A volte ci può essere un numero praticamente infinito di possibili output specificati (ad esempio, quando ci sono diverse variabili che rappresentano numeri reali), mentre altre volte ci possono essere solo due output specificati, come nel caso di una singola variabile che può essere vera o falsa 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 banali il numero è praticamente infinito. Pertanto, nella pratica, documentare tutte le possibili variazioni degli input di test sarebbe impraticabile, in quanto richiederebbe un tempo infinito c) Non è corretta. Iniziare il testing il prima possibile con le review e altri approcci di testing statico non risolve 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 comparsa di difetti successivi nei prodotti di lavoro derivati, riducendo così i costi e la probabilità di failure	FL-1.3.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		d) È corretta. L'uso del partizionamento di equivalenza e dell'analisi ai valori limite per generare test case è un modo per affrontare il principio, poiché queste tecniche di test forniscono un modo sistematico per derivare un sottoinsieme finito di tutti i possibili test case			

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
4	a	a) È corretta. La progettazione dei test prevede l'utilizzo delle condizioni di test per creare test case e altro testware necessario, come i requisiti dei dati di test e le test charter per il testing esplorativo. Vengono inoltre specificati i requisiti dell'ambiente di test, compresi l'infrastruttura e gli strumenti necessari b) Non è corretta. L'esecuzione dei test comporta l'esecuzione dei test case (come parte delle test procedure), ma non copre direttamente l'altro testware menzionato nella domanda, come i requisiti dei dati di test, i requisiti dell'ambiente di test e le condizioni di test 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, a cui viene poi assegnata la priorità insieme ai rischi correlati. Sebbene questa attività implichi il lavoro con le condizioni di test, non copre l'altro testware menzionato nella domanda, come i requisiti dei dati di test, i requisiti dell'ambiente di test e i test case d) Non è corretta. L'implementazione dei test comprende la generazione di test procedure, come test script manuali e automatizzati, che vengono creati a partire dai test case e possono essere assemblati in test suite. Le test procedure vengono prioritizzate e organizzate in una schedulazione di esecuzione dei test. Si creano i dati di test, si costruisce l'ambiente di test e si verifica la sua configurazione. Sebbene questa attività implichi esplicitamente il lavoro con i test case e possa utilizzare i requisiti dei dati di test e dell'ambiente di test per creare i dati di test e l'ambiente di test, non riguarda le condizioni di test	FL-1.4.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
5	c	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 sul modo in cui viene eseguito il testing per un determinato oggetto di test b) Non è corretta. È improbabile che il livello di conoscenza degli utenti del fatto che si sta costruendo un nuovo sistema per loro influisca sulle modalità di esecuzione del testing. Qualsiasi coinvolgimento degli utenti che possa influenzare le modalità di esecuzione del testing è più probabile che sia il risultato di decisioni prese dai tester, dal cliente e dal project manager c) È corretta. Il numero di anni di esperienza dei membri del team del testing delle prestazioni aiuterà a determinare le capacità e le conoscenze (ad esempio, dei diversi strumenti e tipi di difetti) che i membri del team applicheranno durante il testing d) Non è corretta. La struttura organizzativa dei diversi utenti finali (che possono essere vari) cambierà da un utente all'altro. Quindi, potrebbe anche non essere nota quando l'applicazione viene testata, e la struttura organizzativa dell'utente finale può quindi avere un effetto minimo sul modo in cui viene eseguito il testing	FL-1.4.2	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
6	b	a) Non è corretta. La tracciabilità tra i rischi mitigati e i test case superati fornisce poche informazioni, perché per essere mitigati (tramite 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 quelli che non hanno un corrispondente test superato b) È corretta. La tracciabilità tra i requisiti dell'utente e i risultati dell'esecuzione dei test fornisce un'indicazione di quali requisiti dell'utente sono stati testati e quindi fornisce un mezzo per misurare i progressi del progetto (nel contesto del testing) rispetto agli obiettivi di business c) Non è corretta. Non è chiaro se i test case che falliscono forniscano un'indicazione delle capacità del tester più dei test case superati. Ciò dipenderebbe in parte dall'obiettivo di test (ad esempio, creare fiducia o causare failure). Inoltre, tale misurazione 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 di test d) Non è corretta. La tracciabilità tra i rischi identificati e le condizioni di test scritte fornisce un mezzo per determinare quali ulteriori condizioni di test devono essere scritte. Determinare quali rischi valga la pena di testare fa parte del risk management e, in particolare, della mitigazione del rischio	FL-1.4.4	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
7	b	a) Non è corretta. Forti capacità di comunicazione, ascolto attivo 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 abilità generica utile ai tester b) È corretta. La conoscenza del 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 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 gli analisti di business d) Non è corretta. Sebbene la scrupolosità, l'attenzione ai dettagli, la curiosità e l'approccio metodico all'identificazione di difetti difficili da trovare siano tutte abilità generiche utili per i tester, è improbabile che essi generino test case prima di iniziare il testing esplorativo. Infatti, uno dei principi fondamentali del testing esplorativo è che i test case sono generati durante il testing, non sono preparati in anticipo	FL-1.5.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
8	d	a) Non è corretta. Il whole-team approach consente a qualsiasi membro del team con le competenze e le conoscenze necessarie di svolgere qualsiasi compito, ma ciò 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 b) Non è corretta. Il whole-team approach si applica al modo in cui lavora un singolo team (in genere nello sviluppo software Agile); non riguarda il modo in cui si suppone che più team lavorino su progetti più grandi e non suggerisce che sia necessario un solo team "intero" per un progetto completo c) Non è corretta. Il whole-team approach non prevede che ogni membro del team sia coinvolto in ogni decisione importante. Per 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'attuazione di un tale approccio rallenterebbe inutilmente i progressi del team d) È corretta. Sfruttando al meglio le diverse competenze di ciascun membro del team, il whole-team approach favorisce dinamiche di team superiori, promuove una comunicazione e una collaborazione solide e genera una sinergia del team che va a vantaggio dell'intero progetto	FL-1.5.2	K1	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
9	b	a) Non è corretta. Nello sviluppo software Agile, i deliverable vengono prodotti in ogni iterazione e la consegna frequente di incrementi richiede un testing di regressione approfondito. Anche se alcuni (o tutti) questi test di regressione possono essere automatizzati, il testing di regressione (automatizzato o meno) non può essere sostituito dalla test automation del sistema b) È corretta. Se si utilizza un modello di sviluppo sequenziale, all'inizio del ciclo di vita il codice non è disponibile per l'esecuzione e quindi in questo periodo si esegue il testing statico (ad esempio, review). Più avanti 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 del ciclo di vita dello sviluppo software c) Non è corretta. Se si utilizza un modello di sviluppo iterativo, come lo sviluppo software Agile, i test di componente possono essere utilizzati per il testing di regressione per ogni iterazione. In questo caso, c'è una forte argomentazione per l'automatizzazione di questi test di componente, che dovranno essere eseguiti frequentemente, mentre è improbabile che ci sia una forte argomentazione per gli sviluppatori che eseguono questi test di componente manualmente d) Non è corretta. Nella maggior parte dei modelli di sviluppo incrementale, i deliverable vengono prodotti in ogni incremento, richiedendo sia testing statico che dinamico a tutti i livelli di test per ogni incremento consegnato	FL-2.1.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
10	b	a) Non è corretta. I tester dovrebbero revisionare i prodotti di lavoro non appena sono disponibili le bozze, per consentire un testing precoce come parte di un approccio shift-left. Se aspettassero la fase di sviluppo successiva, si potrebbe iniziare un lavoro di sviluppo (e di test) non necessario su prodotti di lavoro non revisionati e difettosi b) È corretta. I tester dovrebbero revisionare i prodotti di lavoro non appena sono disponibili le bozze, per consentire un testing precoce come parte di un approccio shift-left c) Non è corretta. I tester in genere revisionano i prodotti di lavoro che costituiscono la base di test come parte dell'analisi dei test, non prima dell'analisi e della progettazione dei test d) Non è corretta. I tester dovrebbero revisionare i prodotti di lavoro non appena sono disponibili le bozze, per consentire un testing precoce come parte di un approccio shift-left. Aspettare che vengano pubblicati significa che tutti i difetti che potrebbero essere trovati dalla review dei tester saranno presenti nel documento pubblicato	FL-2.1.2	K1	1
11	a	a) È corretta. Il Test-Driven Development (TDD) è un noto esempio di approccio test-first allo sviluppo b) Non è corretta. Il Coverage-Driven Development non è un esempio corretto di approccio test-first allo sviluppo c) Non è corretta. Il Quality-Driven Development non è un esempio corretto di approccio test-first allo sviluppo d) Non è corretta. Il Feature-Driven Development non è un esempio di approccio test-first allo sviluppo, ma è invece una metodologia di sviluppo software Agile basata sulla consegna di funzionalità (al contrario delle user story in Scrum)	FL-2.1.3	K1	1

© International Software Testing Qualifications Board

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

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
13	a	a) È corretta. Il testing di sistema esamina il comportamento e le capacità del sistema completo e copre il testing non funzionale delle caratteristiche qualitative, che include il testing di sicurezza. Questo tipo di testing viene spesso eseguito da un team di test indipendente sulla base delle specifiche del sistema b) Non è corretta. Il testing di integrazione del sistema esamina le interfacce con altri sistemi e servizi esterni c) Non è corretta. Il beta testing è un tipo di testing di accettazione eseguito in un sito esterno da ruoli esterni all'organizzazione di sviluppo 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	FL-2.2.1	K2	1

© International Software Testing Qualifications Board

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

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
15	b	a) Non è corretta. Una mancanza di usabilità fornita dall'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 utenti tipici di testare dinamicamente l'interfaccia utente e di fornire un feedback sulla sua usabilità b) È corretta. Una review del codice può rilevare il codice che non può essere raggiunto da nessun percorso, ma i test dinamici possono solo esercitare il codice raggiungibile e non possono determinare che il codice non può essere raggiunto senza eseguire tutte le possibili combinazioni di input e stati di input, il che è impraticabile per il codice reale c) Non è corretta. I tempi di risposta scadenti per la maggior parte degli utenti attesi 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 d) Non è corretta. Una review del codice 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	FL-3.1.2	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
16	c	a) Non è corretta. Il feedback proviene dagli stakeholder (ad esempio, rappresentanti di business, utenti finali), non dagli sviluppatori, quindi non è probabile che questo feedback informi i manager su quali sviluppatori siano più o meno produttivi b) Non è corretta. Il feedback precoce e frequente degli stakeholder non viene utilizzato dai project manager per assegnare la priorità di interazione con i diversi stakeholder c) È corretta. Ottenere il feedback degli stakeholder presto e frequentemente nel processo di sviluppo del software può essere molto vantaggioso, in quanto facilita la comunicazione precoce di potenziali problemi di qualità, può prevenire incomprensioni sui requisiti e garantisce che qualsiasi modifica dei requisiti degli stakeholder sia compresa e implementata prima d) Non è corretta. Un feedback precoce e frequente può impedire lo sviluppo di un prodotto che non soddisfa le esigenze degli stakeholder, con conseguenti costose rielaborazioni e scadenze non rispettate, quindi, idealmente, non ci dovrebbero essere ritardi. Inoltre, il feedback proviene dagli stakeholder (e non verso loro), che comprendono gli utenti finali, per cui gli utenti finali che forniscono il loro feedback non aiuteranno la comprensione di questi ultimi	FL-3.2.1	K1	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
17	d	Considerando ciascuna delle descrizioni dei compiti elencati: 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 revisionare, 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. Tutti hanno accesso al prodotto di lavoro - (Avvio 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 identificate le anomalie nel prodotto di lavoro - (Review individuale (A): Valutare la qualità del prodotto di lavoro, identificare e registrare le anomalie, le raccomandazioni e le domande utilizzando tecniche di review come le review checklist-based e scenario-based) 4. Le anomalie sono analizzate e discusse (comunicazione e analisi (D): Analizzare e discutere di ciascuna anomalia, determinarne lo stato, la proprietà e le azioni richieste e prendere decisioni di review, di solito in un meeting). Questo potrebbe includere determinare la necessità di una review di follow-up) Quindi: a) Non è corretta b) Non è corretta c) Non è corretta d) È corretta. L'abbinamento corretto è: 1C, 2B, 3A, 4D	FL-3.2.2	K2	1

© International Software Testing Qualifications Board

18	c	Considerando ciascuno dei ruoli elencati: 1. Scribe (o Recorder) - responsabile della raccolta dei feedback dei reviewer e della documentazione delle informazioni della review, come le decisioni prese e le nuove anomalie individuate durante il review meeting. (Registra le informazioni della review, come le decisioni e le nuove anomalie riscontrate durante il review meeting - B) 2. Review Leader - responsabile della supervisione del processo di review, come la selezione dei membri del team di review, la schedulazione dei review meeting e la garanzia che la review sia completata con successo. (Assume la responsabilità generale della review, organizzando ad esempio quando e dove si svolgerà la review - D) 3. Facilitator (o moderatore) - responsabile di assicurare l'efficace svolgimento dei review meeting, compresa la gestione del tempo, la mediazione delle discussioni e la creazione di un ambiente sicuro in cui tutti possano esprimere liberamente le proprie opinioni. (Assicura lo svolgimento efficace dei review meeting e la creazione di un ambiente di review sicuro - A) 4. Manager - responsabile di decidere cosa deve essere revisionato e di allocare le risorse, come il personale e il tempo, per la review. (Decide cosa deve essere revisionato e fornisce le risorse, come il personale e il tempo, per la review - C) Quindi: a) Non è corretta b) Non è corretta c) È corretta. L'abbinamento corretto è: 1B, 2D, 3A, 4C d) Non è corretta	FL-3.2.3	K1	1

© International Software Testing Qualifications Board

19	d	a) Non è corretta. Il testing della tabella delle decisioni è una tecnica di test black-box, quindi basata sulle specifiche e non sulla struttura: 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 b) Non è corretta. L'anticipazione dei potenziali difetti viene utilizzata nell'error guessing (una tecnica di test basata sull'esperienza), non nel testing dei rami (una tecnica basata sulla struttura). Nel testing della tabella delle decisioni, i test case sono derivati dalla specifica che descrive la logica di business 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 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 è 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 e sull'elaborazione. Poiché i test case dipendono dal modo in cui il software è progettato e codificato, possono essere creati solo dopo la progettazione o l'implementazione dell'oggetto di test	FL-4.1.1	K2	1
20	a	a) È corretta. 19 copre la partizione "nessuno sconto", 20 copre la partizione "50% di sconto" e 30 copre la partizione "10% di sconto".	FL-4.2.1	K3	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		Questi tre valori coprono tutte e tre le partizioni di equivalenza valide b) Non è corretta. 11 e 12 coprono la partizione "nessuno sconto", mentre 20 copre la partizione "50% di sconto", coprendo quindi due delle tre partizioni di equivalenza valide c) Non è corretta. 1 copre la partizione "nessuno sconto", mentre 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 d) Non è corretta. 29 e 31 coprono la partizione "nessuno sconto", mentre 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			

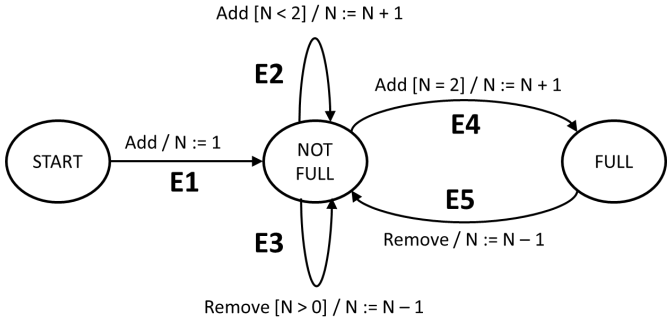
© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
21	d	Il dominio della lunghezza della password ha tre partizioni di equivalenza: • password troppo corte {0, 1, ..., 4, 5} • password OK {6, 7, ..., 11, 12} • password troppo lunghe {13, 14, ...} Per ottenere una copertura completa per la BVA a 3 valori, è necessario testare i seguenti valori: 0, 1, 4, 5, 6, 7, 11, 12, 13, 14. Poiché la BVA a 2 valori è già stata coperta, ciò significa che abbiamo già testato le password di lunghezza: 0, 5, 6, 12 e 13. Ciò significa che le lunghezze aggiuntive che devono essere coperte per passare da un valore di 2 a un valore di 3 sono: 1, 4, 7, 11 e 14. Quindi: a) Non è corretta b) Non è corretta c) Non è corretta d) È corretta	FL-4.2.2	K3	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
22	b	Nella tabella delle decisioni ci sono cinque colonne. Ogni test case copre una di esse. TC1 e TC2 coprono entrambi la Regola 4 TC3 e TC4 coprono entrambi la Regola 2 TC5 copre la Regola 5 Quindi, questi cinque test case coprono tre colonne su cinque, ottenendo una copertura di (3/5)*100% = 60%. Pertanto, l'opzione b) è quella CORRETTA. Quindi: a) Non è corretta b) È corretta c) Non è corretta d) Non è corretta	FL-4.2.3	K3	1

© International Software Testing Qualifications Board

23	c	 Facciamo riferimento alle transizioni con E1, ..., E5 come in figura. La variabile N indica il numero di elementi attualmente archiviati. Ogni evento "Add" lo aumenta di 1 e ogni evento "Remove" lo diminuisce di 1. Si noti che quando l'evento "Add" si verifica mentre si è 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<2, il sistema rimane nello stato NOT FULL. Se N=0, non è possibile effettuare alcuna azione "Remove". Analogamente, se N=3, non è possibile eseguire alcuna azione "Add". 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, perché dopo le prime tre azioni "Add" il sistema si trova nello stato FULL e non c'è una transizione valida che passi da FULL innescata dall'evento "Add". 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

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		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). Quindi: a) Non è corretta b) Non è corretta c) È corretta d) Non è corretta			
24	b	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 che da T2 c) Non è corretta. La copertura delle istruzioni non ci 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, ciò non implica il raggiungimento della copertura completa dei rami	FL-4.3.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
25	c	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 grafico del flusso di controllo, che mostra 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 è misurata come il numero di rami esercitati dai test case diviso per il numero totale di rami ed è espressa come percentuale. Quindi: a) Non è corretta. Un risultato decisionale è un ramo condizionale. Per il testing dei rami, X conta non solo i rami condizionali, ma anche quelli non condizionali b) Non è corretta. La copertura dei rami conta non solo i rami condizionali, ma anche quelli non condizionali c) 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 percentuale d) 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

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
26	a, e	Il testing esplorativo è utile quando le specifiche sono poche o inadeguate, oppure quando il tempo a disposizione per il testing è molto limitato. 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 essenziali, come capacità analitiche, curiosità e creatività. Quindi: 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 essenziali, come capacità analitiche, curiosità e creatività	FL-4.4.2	K2	1

© International Software Testing Qualifications Board

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 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 un umano	FL-4.4.3	K2	1
28	b	a) Non è corretta. Il formato rule-oriented include formati come elenchi di verifica con punti elenco o forme tabulate di mappature input-output, che mostrano esplicitamente 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

© International Software Testing Qualifications Board

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 discusso nella user story. La user story riguarda la visualizzazione degli ordini precedenti	FL-4.5.3	K3	1
30	a	a) È corretta. Questo è qualcosa che può (e dovrebbe) essere controllato *prima* che il codice venga inviato al controllo della versione b) Non è corretta. Questo è qualcosa che può essere controllato *dopo* che la fase (2) è stata eseguita, perché la segnalazione dei conflitti di unione può essere fatta *dopo* che il codice è stato inviato e unito c) Non è corretta. Questo è più adatto come criterio di ingresso per la fase (3) d) Non è corretta. Questo è più adatto come criterio di uscita per la fase (3)	FL-5.1.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
31	b	L'effort medio di sviluppo è di $900.000 e l'effort medio di test è di $90.000 (calcolato sui quattro progetti). Il rapporto medio tra effort di test e sviluppo è di 1:10 ($90.000 : $900.000), il che significa che storicamente, in media, l'effort di test è pari al 10% dell'effort di sviluppo. Quindi, se l'effort di sviluppo è stimato in $800.000, l'effort di test è stimato come: 10% * $800,000 = 0.1 * $800,000 = $80,000. Quindi: a) Non è corretta b) È corretta c) Non è corretta d) Non è corretta	FL-5.1.4	K3	1
32	a	In base alle dipendenze, i test RICERCA devono essere eseguiti per primi, poi i test VISUALIZZAZIONE, quindi i test AGGIUNGERE e, infine, i test ORDINA. All'interno di ciascuno di questi gruppi, l'ordine è determinato dalle priorità dei test case. Quindi, TC1 deve essere eseguito per primo, poi TC2, poi TC4 seguito da TC3 e TC5 come ultimo. Quindi, l'ordine è: TC1, TC2, TC4, TC3, TC5. Quindi: a) È corretta. TC3 sarà il quarto test case eseguito b) Non è corretta c) Non è corretta d) Non è corretta	FL-5.1.5	K3	1

© International Software Testing Qualifications Board

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 (Q2) c) Non è corretta. Il testing di accettazione dell'utente 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 (guida lo sviluppo) (Q1)	FL-5.1.7	K2	1
34	c	Considerando ciascuno dei rischi elencati e le relative mitigazioni: 1. Le risposte lunghe del sistema (1) possono essere testate nel testing delle prestazioni (B) 2. I cambiamenti delle preferenze dei consumatori (2) sono di solito fuori dal nostro controllo, quindi di solito accettiamo questo rischio (A) 3. L'allagamento della sala server (3) può causare perdite significative, quindi dovremmo 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 imprecisi (4) suggerisce un potenziale problema di limite, che può essere efficacemente rilevato con tecniche come la BVA (C) Quindi: a) Non è corretta b) Non è corretta c) È corretta. L'abbinamento corretto di rischio e mitigazione è: 1B, 2A, 3D e 4C d) Non è corretta	FL-5.2.4	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
35	a	a) È corretta. Le metriche della qualità del prodotto misurano le caratteristiche di qualità. Il mean time to failure misura la maturità, quindi è una metrica della qualità del prodotto b) Non è corretta. Questo è un esempio di metrica dei difetti, non di metrica della 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 della qualità del prodotto	FL-5.3.1	K1	1
36	a	a) È corretta. Il cliente si trova in una località e con un fuso orario diversi, quindi potrebbe essere difficile comunicare face-to-face b) Non è corretta. Le dashboard sono solitamente 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 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 l'Europa e l'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

© International Software Testing Qualifications Board

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 (ad esempio, un ambiente di test), il CM registra 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 molto valore per lo sviluppatore b) È corretta. Dai risultati di test sembra che il sistema ignori i duplicati e ordini l'elenco ignorando le ripetizioni. Questa è probabilmente la causa dei failure in TC3, TC4, TC5. Queste informazioni possono aiutare lo sviluppatore a trovare il difetto e a risolverlo 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 siamo a conoscenza del fatto che i test case abbiano dei difetti	FL-5.5.1	K3	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
39	c	Considerando ciascuna delle categorie di strumenti elencate e le relative descrizioni: 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 della distribuzione - per esempio, macchine virtuali, strumenti di containerizzazione (3) C. Strumenti DevOps - supportano la pipeline di consegna DevOps, il tracciamento del flusso di lavoro, il processo(i) di build automatizzati, la continuous integration/continuous delivery (CI/CD) (1) D. Strumenti di collaborazione - facilitano la comunicazione (2) Quindi: 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

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
40	a	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 b) Non è corretta. Utilizzando gli strumenti di test, la responsabilità del testing NON è condivisa con il fornitore dello strumento, poiché quest'ultimo 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 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 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 provenire da un'altra parte della base di test, come le specifiche di progettazione	FL-6.2.1	K1	1

© International Software Testing Qualifications Board