Simulazione d’esame - Risposte

Simulazione d'esame set D
Versione 1.0

ISTQB® Certified Tester Syllabus Foundation Level

Compatibile con il Syllabus versione 4.0

International Software Testing Qualifications Board

Versione 1.0 Release 16 ottobre 2023

© 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 7
2 8
3 9
4 10
5 11
6 12
7 13
8 14
9 15
10 16
11 16
12 17
13 18
14 19
15 20
16 21
17 22
18 23
19 23
20 24
21 24
22 25
23 26
24 27
25 28
26 28
27 29
28 29
29 30
30 30
31 31
32 31
33 32
34 32
35 32
36 33
37 34
38 35
39 36
40 37

© 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-1.1.1	K1	1
2	c	FL-1.2.3	K2	1
3	a	FL-1.3.1	K2	1
4	b	FL-1.4.1	K2	1
5	a	FL-1.4.3	K2	1
6	d	FL-1.4.5	K2	1
7	a	FL-1.5.2	K1	1
8	b	FL-1.5.3	K2	1
9	a	FL-2.1.2	K1	1
10	a	FL-2.1.3	K1	1
11	d	FL-2.1.4	K2	1
12	b	FL-2.1.6	K2	1
13	a	FL-2.2.2	K2	1
14	b	FL-2.3.1	K2	1
15	c	FL-3.1.1	K1	1
16	c	FL-3.1.2	K2	1
17	b	FL-3.2.2	K2	1
18	b	FL-3.2.3	K1	1
19	b	FL-4.1.1	K2	1
20	b, e	FL-4.2.1	K3	1

Numero di domanda (#)	Risposta corretta	LO	Livello K	Punti
21	c	FL-4.2.2	K3	1
22	a	FL-4.2.3	K3	1
23	b	FL-4.2.4	K3	1
24	c	FL-4.3.1	K2	1
25	a	FL-4.3.3	K2	1
26	c	FL-4.4.1	K2	1
27	d	FL-4.4.2	K2	1
28	d	FL-4.5.1	K2	1
29	a	FL-4.5.3	K3	1
30	b, d	FL-5.1.3	K2	1
31	a	FL-5.1.4	K3	1
32	b	FL-5.1.5	K3	1
33	c	FL-5.1.7	K2	1
34	b	FL-5.2.1	K1	1
35	b, e	FL-5.2.2	K2	1
36	c	FL-5.3.2	K2	1
37	d	FL-5.4.1	K2	1
38	a	FL-5.5.1	K3	1
39	b	FL-6.1.1	K2	1
40	c	FL-6.2.1	K1	1

© International Software Testing Qualifications Board

Risposte

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
1	d	a) Non è corretta. Individuare e correggere i difetti nell'oggetto di test non è un obiettivo tipico di test, poiché sebbene individuare i difetti sia un obiettivo del testing, correggere i difetti non è un'attività del testing b) Non è corretta. Mantenere una comunicazione efficace con gli sviluppatori non è un obiettivo tipico di test, poiché, sebbene sia utile per raggiungere altri obiettivi del testing, ad esempio fornire agli stakeholder informazioni che consentano di prendere decisioni informate, non è una ragione primaria per eseguire il testing c) Non è corretta. Validare il rispetto dei requisiti legali non è un tipico obiettivo di test, perché la validazione si occupa di verificare se il sistema soddisfa le esigenze degli utenti e di altri stakeholder nel suo ambiente operativo. Controllare che i requisiti legali siano stati rispettati è una forma di verifica d) È corretta. Si crea fiducia nella qualità dell'oggetto di test attraverso l'esecuzione di test che hanno successo	FL-1.1.1	K1	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
2	c	a) Non è corretta. L'errore di calcolo dei bonus è un failure del sistema, non un difetto b) Non è corretta. Il sistema che non supporta adeguatamente gli utenti disabili è un failure che alla fine si traduce in una multa, ma la multa stessa non è un failure (sembra essere il corretto funzionamento del sistema normativo) c) È corretta. L'errore è stato commesso dal programmatore e questo errore è stato causato dal fatto che ha lavorato sotto una forte pressione di tempo, che è la root cause del successivo difetto d) Non è corretta. La cattiva progettazione dell'interfaccia utente, che non si rivolge adeguatamente agli utenti disabili, è un difetto di progettazione causato da un errore del progettista. Pertanto, la progettazione dell'interfaccia utente include un difetto di progettazione, non un errore del progettista	FL-1.2.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
3	a	a) È corretta. Il principio "i test perdono di efficacia" riguarda l'idea che la ripetizione di test identici su codice inalterato difficilmente può portare alla scoperta di nuovi difetti e quindi la modifica dei test può essere essenziale. Se si utilizzano condizioni di test di alto livello per generare ogni volta nuovi test, i test non saranno identici e non dovrebbero "perdere di efficacia" b) Non è corretta. Il principio "l'assenza di difetti è un'idea sbagliata" riguarda la necessità di garantire che le esigenze degli utenti siano soddisfatte anche se si esegue molto testing e non si trovano difetti (cioè, è necessaria anche la validazione). L'uso di condizioni di test di alto livello per generare test case ed eseguire test non affronta direttamente questo problema c) Non è corretta. 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. Questo principio viene tipicamente affrontato iniziando il testing (sia statico che dinamico) il più presto possibile, non utilizzando condizioni di test di alto livello per generare test case ed eseguire i test d) Non è corretta. Il principio "i difetti si raggruppano in cluster" riguarda la distribuzione dei difetti in un sistema, che in genere segue una distribuzione di Pareto. L'uso di condizioni di test di alto livello per generare test case ed eseguire test non risolve questo problema, che è tipicamente affrontato dal testing basato sul rischio	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
4	b	Considerando le attività di test elencate e i relativi compiti: A. Analisi dei test - Per identificare le funzionalità che richiedono testing, si analizza la base di test e si definiscono le condizioni di test, che vengono poi prioritizzate insieme ai rischi correlati. Durante l'analisi dei test, di solito si scoprono i difetti nella base di test e si può anche valutare la testabilità dell'oggetto di test. (Compito 4) B. Progettazione dei test - Consiste nell'utilizzare le condizioni di test per creare test case e altro testware necessario, quali i requisiti dei dati di test e le test charter per il testing esplorativo. (Compito 1) C. Implementazione dei test - Le test procedure, ad esempio i test script manuali e automatizzati, vengono create dai test case e possono essere assemblate in test suite. Le test procedure vengono prioritizzate e organizzate in una schedulazione di esecuzione dei test. (Compito 3) D. Completamento dei test - Si verifica in occasione di milestone di progetto quali il rilascio, la fine dell'iterazione o la fine del livello di test. Il testware viene identificato e archiviato o consegnato ai team appropriati per fini di riutilizzo, l'ambiente di test viene chiuso e le attività di test vengono analizzate per derivare lezioni apprese e miglioramenti futuri. (Compito 2) Quindi: a) Non è corretta b) È corretta. L'abbinamento CORRETTO è: 1B, 2D, 3C, 4A c) Non è corretta d) Non è corretta	FL-1.4.1	K2	1

© International Software Testing Qualifications Board

Non è corretto

5	a	Considerando ogni testware elencato e l'attività di test che lo produce: i. Il test completion report è un output dell'attività di completamento dei test ii. I dati contenuti in un database utilizzato per gli input di test e i risultati attesi sono i dati di test – un output dell'attività di implementazione dei test iii. L'elenco degli elementi necessari per costruire l'ambiente di test è il requisito dell'ambiente di test – un output dell'attività di progettazione dei test iv. Le sequenze documentate di test case in ordine di esecuzione sono le test procedure – un output dell'attività di implementazione dei test v. Test case – un output dell'attività di progettazione dei test L'implementazione dei test produce i seguenti output: test procedure (iv), test script automatizzati, test suite, dati di test (ii), schedulazione di esecuzione dei test ed elementi dell'ambiente di test quali stub, driver, simulatori e virtualizzazioni dei servizi. Quindi: a) È corretta. Gli elementi ii e iv dell'elenco sono prodotti come risultato dell'implementazione dei test b) Non è corretta c) Non è corretta d) Non è corretta	FL-1.4.3	K2	1

© International Software Testing Qualifications Board

6	d	a) Non è corretta. Il ruolo del testing è principalmente responsabile degli aspetti tecnici e ingegneristici del testing, quali l'analisi dei test, la progettazione dei test, l'implementazione dei test e l'esecuzione dei test. La valutazione dei difetti della base di test e della testabilità dell'oggetto di test sono compiti eseguiti come parte dell'analisi dei test, quindi è probabile che siano compiti svolti dal ruolo del testing b) Non è corretta. Il ruolo del testing è principalmente responsabile degli aspetti tecnici e ingegneristici del testing, quali l'analisi dei test, la progettazione dei test, l'implementazione dei test e l'esecuzione dei test. La definizione dei requisiti dell'ambiente di test è un compito svolto come parte della progettazione dei test, quindi è probabile che sia un compito svolto dal ruolo del testing c) Non è corretta. Il ruolo del testing è principalmente responsabile degli aspetti tecnici e ingegneristici del testing, quali l'analisi dei test, la progettazione dei test, l'implementazione dei test e l'esecuzione dei test. La valutazione della testabilità di un oggetto di test è un compito svolto come parte dell'analisi dei test, quindi è probabile che sia un compito svolto dal ruolo del testing d) È corretta. Il ruolo di test management coinvolge principalmente le attività relative alla pianificazione dei test, al monitoraggio e controllo dei test e al completamento dei test. Pertanto, la creazione del test completion report, che è l'output principale dell'attività di completamento dei test, è probabile che sia un compito svolto dal ruolo di test management	FL-1.4.5	K2	1

© International Software Testing Qualifications Board

7	a	a) È corretta. Il whole team approach promuove una solida comunicazione e collaborazione tra i membri del team b) Non è corretta. Sebbene il whole team approach dia priorità alla responsabilità collettiva della qualità, ogni singolo membro del team è comunque ugualmente responsabile della qualità c) Non è corretta. Il whole team approach riguarda il modo in cui il team lavora insieme, con l'obiettivo di ottenere deliverable di qualità superiore, ma non si traduce necessariamente in una distribuzione più rapida agli utenti finali d) Non è corretta. Quando si utilizza il whole team approach, i tester lavorano con i rappresentanti di business per creare i test di accettazione. Non esiste alcun suggerimento che l'approccio riduca la collaborazione con gli utenti di business esterni	FL-1.5.2	K1	1

© International Software Testing Qualifications Board

8	b	Considerando ognuno dei vantaggi e degli svantaggi elencati dell'indipendenza del testing: i. I tester lavorano in una sede diversa da quella degli sviluppatori - Idealmente, vogliamo una stretta collaborazione tra tester e sviluppatori, che non aumenta con l'isolamento. Questo è quindi uno svantaggio ii. I tester mettono in discussione le ipotesi che i programmatori fanno durante la scrittura del codice - I tester e gli sviluppatori hanno background, punti di vista tecnici e potenziali bias diversi, cosa che consente ai tester di mettere utilmente in discussione le ipotesi fatte dagli stakeholder durante la specifica e l'implementazione del sistema. Si tratta quindi di un vantaggio iii. Si instaura una dinamica conflittuale tra tester e sviluppatori - il principale svantaggio dell'indipendenza del testing è che i tester possono rimanere isolati dal team di sviluppo, causando problemi di comunicazione, mancanza di collaborazione e potenzialmente un rapporto conflittuale, con i tester che vengono incolpati di ritardi e di essere colli di bottiglia nel processo di rilascio. Si tratta quindi di uno svantaggio iv. Gli sviluppatori si sono convinti che i tester siano i principali responsabili della qualità - Uno degli svantaggi dell'indipendenza del testing è che i tester possono isolarsi dal team di sviluppo, facendo sì che gli sviluppatori si sentano meno responsabili della qualità. Si tratta quindi di uno svantaggio v. I tester hanno bias diversi da quelli degli sviluppatori - Il vantaggio principale dell'indipendenza del testing è che i tester hanno maggiori probabilità di identificare diversi tipi di failure e difetti rispetto agli sviluppatori, a causa dei loro diversi background, punti di vista tecnici e potenziali bias, compresi i bias cognitivi Quindi: a) Non è corretta	FL-1.5.3	K2	1

© International Software Testing Qualifications Board

		b) È corretta. Le voci dell'elenco che mostrano i benefici sono ii e v c) Non è corretta d) Non è corretta			
9	a	a) È corretta. Ogni livello di test ha obiettivi di test specifici e distinti, poiché in ogni livello di test viene testata una forma diversa di oggetto di test (ad esempio, singolo componente, sistema completo) e la sovrapposizione degli obiettivi di test porterebbe a inutili duplicazioni b) Non è corretta. L'analisi e la progettazione dei test per un determinato livello di test dovrebbero iniziare durante la corrispondente fase di sviluppo, per facilitare l'esecuzione del testing anticipato (ad esempio, l'analisi e la progettazione dei test di accettazione dovrebbero iniziare durante l'analisi dei requisiti). L'implementazione dei test inizierà generalmente più tardi e l'esecuzione dei test inizierà durante il livello di test c) Non è corretta. La progettazione dei test per un determinato livello di test dovrebbe iniziare durante la corrispondente fase di sviluppo per facilitare l'esecuzione del testing anticipato, tuttavia la progettazione dei test (ad esempio, la generazione dei test case) deve basarsi su una base di test concordata, non su una bozza iniziale, altrimenti si rischia di sprecare un significativo effort di test per creare test case per un progetto che in seguito cambia d) Non è corretta. Il quality control si applica a tutte le attività di sviluppo, il che significa che ogni attività di sviluppo del software ha una corrispondente attività di test. Tuttavia, la stessa simmetria non si applica al testing dinamico e statico. Ci sono alcune attività di testing statico (ad esempio, l'analisi statica) per le quali non esiste un'ovvia attività di testing dinamico corrispondente	FL-2.1.2	K1	1

© International Software Testing Qualifications Board

10	a	a) È corretta. Il Behavior-Driven Development (BDD) è un noto esempio di approccio test-first allo sviluppo b) Non è corretta. Il Test Level Driven Development non è un esempio corretto di approccio test-first allo sviluppo c) Non è corretta. Il Function-Driven Development non è un esempio corretto di approccio test-first allo sviluppo d) Non è corretta. Il Performance-Driven Development non è un esempio corretto di approccio test-first allo sviluppo	FL-2.1.3	K1	1
11	d	a) Non è corretta. DevOps in genere aumenta la visibilità delle caratteristiche di qualità non funzionali, quali le prestazioni e l'affidabilità b) Non è corretta. I processi automatizzati come la continuous integration/continuous delivery (CI/CD) utilizzati in DevOps facilitano la stabilità degli ambienti di test c) Non è corretta. I processi automatizzati come CI/CD utilizzati in DevOps riducono generalmente la necessità di eseguire testing manuale d) È corretta. L'implementazione di DevOps può comportare diversi rischi e sfide, tra cui la necessità di definire e impostare la pipeline di delivery, introdurre e mantenere strumenti di CI/CD e stabilire e mantenere la test automation	FL-2.1.4	K2	1

© International Software Testing Qualifications Board

12	b	a) Non è corretta. I vantaggi delle retrospettive includono rafforzare il legame nel team e imparare dalla condivisione dei problemi, nonché una migliore collaborazione tra sviluppatori e tester attraverso la revisione e il miglioramento delle pratiche di lavoro. Riprendere le persone che un membro del team ritiene non aver contribuito pienamente al raggiungimento della qualità, come richiesto dal whole team approach, non contribuirà a rafforzare il legame e la collaborazione di questo team b) È corretta. Durante la retrospettiva, il gruppo discute gli aspetti del progetto che hanno avuto successo e che dovrebbero essere mantenuti, nonché le aree che potrebbero essere migliorate e come farlo c) Non è corretta. I vantaggi delle retrospettive si basano sull'aumento dell'efficacia e dell'efficienza attraverso i miglioramenti del processo; non sono un'occasione per sfogarsi e criticare il management e i clienti. Inoltre, i risultati vengono registrati, di solito nel test completion report, quindi tutto ciò che viene detto durante il meeting potrebbe essere letto da altri stakeholder d) Non è corretta. Le retrospettive sono meeting che si tengono in genere alla fine di un'iterazione, in cui i membri del team si concentreranno sulla discussione dei problemi legati alla qualità che si sono verificati nell'iterazione corrente. Non sono utilizzate per fare piani o prendere decisioni tecniche per l'iterazione successiva; questo verrebbe fatto nel meeting di pianificazione dell'iterazione all'inizio dell'iterazione successiva	FL-2.1.6	K2	1

© International Software Testing Qualifications Board

13	a	a) È corretta. Verificare che la funzione di ordinamento metta gli elementi della lista o della matrice in ordine crescente significa valutare la correttezza funzionale della funzione di ordinamento, il che fa parte del testing funzionale b) Non è corretta. Verificare se la funzione di ordinamento soddisfa il suo requisito non funzionale di completare l'ordinamento entro un secondo fa parte del testing di efficienza delle prestazioni, che fa parte del testing non funzionale c) Non è corretta. Verificare la facilità con cui la funzione di ordinamento può essere modificata da ordinamento crescente a ordinamento decrescente significa testare la sua modificabilità, una forma di testing di manutenibilità non funzionale, che fa parte del testing non funzionale d) Non è corretta. Verificare che la funzione di ordinamento continui a funzionare correttamente quando si passa da un'architettura a 32 bit a una a 64 bit significa testare la sua adattabilità, una forma di testing di portabilità, che fa parte del testing non funzionale	FL-2.2.2	K2	1

© International Software Testing Qualifications Board

14	b	a) Non è corretta. Ammesso che i tester possano verificare la facilità di cambiare il sistema di cambio della valuta, ciò verrebbe fatto con il testing di manutenibilità piuttosto che con il testing di manutenzione, quindi questo non è un fattore scatenante che porta al testing di manutenzione b) È corretta. Una modifica del sistema (quale una correzione o un miglioramento) è un esempio di fattore scatenante che porta al testing di manutenzione. La rimozione dell'opzione di rimborso del sistema di cambio valuta è una correzione che porterebbe al testing di manutenzione c) Non è corretta. Se il team Agile ha iniziato a sviluppare una user story che aggiunge una nuova funzionalità di fidelizzazione dei clienti al sistema di cambio valuta, allora si procederà a testare la nuova funzionalità e poi si eseguirà il testing di regressione. In questa situazione non è richiesto alcun testing di manutenzione d) Non è corretta. La riconfigurazione del sistema di cambio valuta per supportare sia le transazioni in lingua locale che in lingua inglese non è una modifica del sistema, né un cambiamento dell'ambiente operativo, né una dismissione del sistema, che sono i tre fattori scatenanti che portano al testing di manutenzione	FL-2.3.1	K2	1

© International Software Testing Qualifications Board

15	c	a) Non è corretta. La maggior parte dei prodotti di lavoro può essere esaminata utilizzando una qualche forma di testing statico, e un contratto deve essere interpretabile da esseri umani e quindi potrebbe essere oggetto di review, che è una forma di testing statico b) Non è corretta. La maggior parte dei prodotti di lavoro può essere esaminata utilizzando una qualche forma di testing statico, e un test plan deve essere interpretabile da esseri umani e quindi può essere oggetto di review, il che è una forma di testing statico c) È corretta. La maggior parte dei prodotti di lavoro può essere esaminata utilizzando una qualche forma di testing statico; tuttavia, non è adatto per i prodotti di lavoro che sono troppo complessi per l'interpretazione umana e che non dovrebbero essere analizzati con strumenti, dove il codice criptato è troppo complesso per gli esseri umani e se è correttamente criptato non sarà analizzabile dalla maggior parte degli strumenti d) Non è corretta. La maggior parte dei prodotti di lavoro può essere esaminata utilizzando una qualche forma di testing statico, e una test charter deve essere interpretabile da esseri umani e quindi potrebbe essere oggetto di review, che è una forma di testing statico	FL-3.1.1	K1	1

© International Software Testing Qualifications Board

16	c	a) Non è corretta. Ci sono alcuni tipi di difetti che possono essere trovati sia dal testing statico che dal testing dinamico, quale ad esempio un difetto di programmazione che può essere osservato da un reviewer durante una review del codice e che causa un failure osservabile durante il testing dinamico b) Non è corretta. Ci sono alcuni tipi di difetti che possono essere rilevati solo dal testing statico, quali codice irraggiungibile, modelli di progettazione non implementati come desiderato e difetti nei prodotti di lavoro non eseguibili c) È corretta. Ci sono alcuni tipi di difetti che possono essere rilevati sia dal testing statico che dal testing dinamico, quale ad esempio un difetto di programmazione che può essere osservato da un reviewer durante una review del codice e che causa un failure osservabile durante il testing dinamico. Esistono anche alcuni tipi di difetti che possono essere rilevati solo dal testing statico, quali codice irraggiungibile, modelli di progettazione non implementati come desiderato e difetti nei prodotti di lavoro non eseguibili d) Non è corretta. Ci sono alcuni tipi di difetti che possono essere rilevati solo con il testing dinamico, quali i problemi di prestazioni o di memoria che possono essere osservati solo durante l'esecuzione del codice o del sistema	FL-3.1.2	K2	1

© International Software Testing Qualifications Board

17	b	Le cinque descrizioni elencate e le corrispondenti attività del processo di review sono: 1. Descrive una parte dell'attività di "comunicazione e analisi" 2. Descrive una parte dell'attività di "correzione e reporting" 3. Descrive una parte dell'attività di "review individuale" 4. Descrive una parte dell'attività di "pianificazione" 5. Descrive una parte dell'attività di "avvio della review" Il processo di review generico della ISO/IEC 20246, delineato nel syllabus, comprende le seguenti attività in questo ordine logico: • Pianificazione (4) • Avvio della review (5) • Review individuale (3) • Comunicazione e analisi (1) • Correzione e reporting (2) Quindi: a) Non è corretta b) È corretta. La sequenza corretta delle attività è: 4 - 5 - 3 - 1 - 2 c) Non è corretta d) Non è corretta	FL-3.2.2	K2	1

© International Software Testing Qualifications Board

18	b	a) Non è corretta. Il manager è responsabile di decidere cosa deve essere revisionato e di allocare le risorse, ad esempio tempo e persone, per la review b) È corretta. Il moderatore (o facilitatore) ha la responsabilità di garantire che i review meeting si svolgano in modo efficace, gestendo il tempo, mediando le discussioni e creando un ambiente sicuro in cui tutti possano esprimere liberamente le proprie opinioni c) Non è corretta. Il presidente non è un ruolo riconosciuto nelle review d) Non è corretta. Il review leader è 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	FL-3.2.3	K1	1
19	b	a) Non è corretta. Il documento non si riferisce alla struttura interna dell'oggetto di test, ma specifica il comportamento desiderato dell'oggetto di test. Pertanto, le tecniche di test white-box non saranno utili nella progettazione dei test case b) È corretta. Il documento è un requisito che specifica il comportamento desiderato dell'oggetto di test. Pertanto, le tecniche di test più adatte in questo caso sono le tecniche di test black-box (ad esempio, analisi ai valori limite o testing della tabella delle decisioni) c) Non è corretta. Sebbene sia possibile utilizzare tecniche di test basate sull'esperienza per progettare test case in base a questo documento, saranno più adatte le tecniche di test black-box. Il documento descrive una regola di business precisa e, inoltre, una dicitura come "supera i $100" suggerisce l'esistenza di importanti confini di partizione di equivalenza che dovrebbero essere testati utilizzando tecniche di test black-box quali l'analisi ai valori limite d) Non è corretta. Le tecniche di test basate sul rischio non sono un tipo riconosciuto di tecnica di test	FL-4.1.1	K2	1

© International Software Testing Qualifications Board

20	b, e	Ci sono due partizioni di equivalenza non ancora coperte, che corrispondono a "sconto per studenti" e "sconto per pensionati". a) Non è corretta. CY - BY = 64, quindi questi input corrispondono alla partizione "nessuno sconto" già coperta b) È corretta. CY - BY = 65, quindi questi input corrispondono a una partizione non ancora coperta ("sconto per pensionati") c) Non è corretta. CY - BY = -65, quindi questi input corrispondono alla partizione "messaggio di errore" già coperta d) Non è corretta. CY - BY = 18, quindi questi input corrispondono alla partizione "nessuno sconto" già coperta e) È corretta. CY - BY = 0, quindi questi input corrispondono a una partizione non ancora coperta ("sconto per studenti")	FL-4.2.1	K3	1
21	c	Esistono tre partizioni di equivalenza: {..., -2, -1}, {0, 1, 2}, {3, 4, ...}. Per la BVA a 2 valori devono essere coperti tutti i valori limite di tutte le partizioni di equivalenza. I valori limite sono -1 (per la partizione "temperatura troppo bassa"), 0, 2 (per la partizione "temperatura OK") e 3 (per la partizione "temperatura troppo alta"). Quindi: a) Non è corretta b) Non è corretta c) È corretta. L'opzione corretta è: -1, 0, 2, 3 d) Non è corretta	FL-4.2.2	K3	1

© International Software Testing Qualifications Board

22	a	I test case TC1, TC2, TC3 e TC4 coprono, rispettivamente, le regole R2, R3, R7 e R6 nella tabella delle decisioni. a) È corretta. Le condizioni "66 anni", "non registrato" e "nessuna esperienza" corrispondono alla regola R4, che non è coperta dai test case esistenti, quindi dopo l'aggiunta di questo test case, la copertura della tabella delle decisioni aumenterà b) Non è corretta. Le condizioni "55 anni", "non registrata" e "2 anni di esperienza" corrispondono alla regola R2, già coperta da TC1. Pertanto, l'aggiunta di questo test case non aumenterà la copertura c) Non è corretta. Le condizioni "19 anni", "registrata" e "5 anni di esperienza" corrispondono alla regola R6, già coperta da TC4. Pertanto, l'aggiunta di questo test case non aumenterà la copertura d) Non è corretta. I test case esistenti coprono solo 4 delle 7 colonne della tabella delle decisioni. La copertura può essere aumentata aggiungendo test case che coprano le colonne non ancora coperte, cioè R1, R4 e R5	FL-4.2.3	K3	1

© International Software Testing Qualifications Board

23	b	a) Non è corretta. Questa sequenza di cinque eventi copre 4 diverse transizioni valide (entrambi gli eventi "Non disponibile" corrispondono alla stessa transizione tra S1 e S3). Questo test case copre 4 delle 7 transizioni valide b) È corretta. Questa sequenza di cinque eventi copre 5 transizioni diverse (il primo evento "Disponibile" corrisponde a una transizione tra S1 e S2 e il secondo evento "Disponibile" corrisponde a una transizione tra S3 e S2, quindi sono coperte due transizioni diverse). Questo test case copre 5 delle 7 transizioni valide e raggiunge la copertura valida delle transizioni più alta c) Non è corretta. Questa sequenza di cinque eventi copre 3 transizioni diverse (entrambi gli eventi "Disponibile" corrispondono alla stessa transizione da S1 a S2; entrambi gli eventi "Cambio Camera" corrispondono alla stessa transizione da S2 a S1). Questo test case copre 3 delle 7 transizioni valide d) Non è corretta. Questa sequenza di cinque eventi non rappresenta un test case fattibile, perché dopo "Annullamento" il sistema finisce nello stato Fine e non è possibile eseguire altre transizioni valide	FL-4.2.4	K3	1

© International Software Testing Qualifications Board

24	c	a) Non è corretta. Una linea di codice con un difetto, quando viene eseguita, non deve necessariamente causare un failure. Per esempio, una linea x := y / z causerà un failure *solo* quando z è uguale a 0 b) Non è corretta. Il 100% di copertura delle istruzioni non garantisce il 100% di copertura dei rami. Ad esempio, un test case con x=0 per il codice 1. IF (x=0) THEN 2. A; 3. ENDIF raggiunge il 100% di copertura delle istruzioni, ma non copre il ramo da 1 a 3 c) È corretta. Il 100% di copertura delle istruzioni significa che ogni istruzione eseguibile è stata eseguita almeno una volta d) Non è corretta. Il test case rimosso può fornire la copertura di alcune istruzioni che non sono coperte da nessuno degli altri due test case, nel qual caso i due test case rimanenti non raggiungeranno insieme il 100% di copertura delle istruzioni	FL-4.3.1	K2	1

© International Software Testing Qualifications Board

25	a	a) È corretta. Un punto di forza fondamentale che accomuna tutte le tecniche di test white-box è che l'intera implementazione del software viene presa in considerazione durante il testing, il che facilita l'individuazione dei difetti anche quando le specifiche del software sono vaghe, obsolete o incomplete. Ciò significa che il testing white-box può trovare difetti quali una funzionalità in più aggiunta al codice (accidentalmente o deliberatamente) che non dovrebbe essere presente, cosa che il testing black-box non è in grado di rilevare b) Non è corretta. Il fatto che la copertura possa essere definita con precisione non è la ragione giusta. Il livello di copertura raggiunto avrebbe un impatto molto maggiore rispetto alla possibilità di misurare la copertura c) Non è corretta. Se il software non implementa uno o più requisiti, è improbabile che il testing white-box rilevi i difetti di omissione che ne derivano d) Non è corretta. Anche se è vero, non è la risposta giusta, perché non c'è alcun collegamento tra la capacità di essere utilizzato sia nel testing statico che in quello dinamico e l'affermazione che il testing white-box facilita l'individuazione dei difetti con specifiche vaghe	FL-4.3.3	K2	1
26	c	L'error guessing consiste nel prevedere gli errori, i difetti e i failure sulla base delle conoscenze del tester. a) Non è corretta. Questo è un esempio di previsione dell'errore dello sviluppatore b) Non è corretta. Questo è un esempio di previsione del difetto c) È corretta. Questo è un esempio di potenziale root cause di un difetto, che non è né un errore, né un difetto, né un failure, e che è difficile da prevedere per un tester d) Non è corretta. Questo è un esempio di previsione di un failure, forse basato sull'esperienza di sistemi precedenti in questo dominio di applicazione	FL-4.4.1	K2	1

© International Software Testing Qualifications Board

27	d	a) Non è corretta. Nel testing esplorativo, i test case vengono solitamente creati durante la sessione di testing esplorativo, insieme all'analisi dei test, all'implementazione dei test e all'esecuzione dei test b) Non è corretta. Nel testing esplorativo, i test vengono progettati, eseguiti e valutati contemporaneamente, mentre il tester impara a conoscere l'oggetto di test c) Non è corretta. I risultati del testing esplorativo dipendono fortemente dall'esperienza del tester, quindi anche se i risultati del testing esplorativo possono essere usati come predittori del rischio e utilizzati per valutare se ci saranno meno o più difetti, ad esempio, rispetto alla precedente sessione di testing esplorativo, non sono un buon esempio di modello affidabile di previsione dei difetti in grado di prevedere il numero di difetti rimanenti d) È corretta. Durante il testing esplorativo, i tester possono utilizzare tutte le tecniche che ritengono utili	FL-4.4.2	K2	1
28	d	a) Non è corretta. Il planning poker può stimare l'effort per una user story già scritta. Non aiuta a capire cosa dovrebbe essere consegnato b) Non è corretta. Le review non sono una pratica di scrittura collaborativa delle user story c) Non è corretta. La pianificazione delle iterazioni è una pratica legata al progetto, utilizzata per pianificare il lavoro, non per capire cosa dovrebbe essere consegnato d) È corretta. La conversazione spiega come verrà utilizzato il software e spesso consente al team di definire criteri di accettazione significativi, ottenendo così una visione condivisa di ciò che dovrebbe essere consegnato	FL-4.5.1	K2	1

© International Software Testing Qualifications Board

29	a	a) È corretta. Questo test case è correlato ai criteri di accettazione 2 e 3, perché verifichiamo se possiamo impostare la fascia di prezzo (criterio di accettazione 2) e se i risultati si aggiornano dinamicamente dopo aver regolato il filtro della fascia di prezzo (criterio di accettazione 3) b) Non è corretta. Questo test case non è correlato a nessuno dei criteri di accettazione. Verifica se il filtro imposta dinamicamente l'intervallo di prezzo minimo e massimo predefinito e non che un cliente possa farlo c) Non è corretta. Questo test case non è correlato a nessuno dei criteri di accettazione. Verifica la funzionalità di cambio valuta, che non è discussa in questa user story d) Non è corretta. Questo test case non è correlato a nessuno dei criteri di accettazione. Verifica la compatibilità dell'applicazione con diversi browser, che non viene discussa in questa user story	FL-4.5.3	K3	1
30	b, d	a) Non è corretta. L'approvazione del budget è un esempio di criterio di ingresso. Non avrebbe senso approvare il budget per un'attività che è già stata svolta b) È corretta. L'esaurimento del budget può essere considerato un valido criterio di uscita c) Non è corretta. La disponibilità di risorse è un esempio di criterio di ingresso per il testing d) È corretta. La copertura è una misura di completezza, quindi è un tipico criterio di uscita e) Non è corretta. Questo è un esempio di criterio di ingresso, verificato prima dell'avvio del progetto	FL-5.1.3	K2	1

© International Software Testing Qualifications Board

31	a	Utilizzando la tecnica di stima a tre punti, la stima finale (E) è calcolata come: E = (a + 4*m + b) / 6, dove a è la stima più ottimistica, m è la stima più probabile e b è la stima più pessimistica. Quindi: a) È corretta. In questo caso, la stima per l'esecuzione di un singolo test case è: E = (1h + 4*3h + 8h) / 6 = 3,5 ore Quindi, il tempo totale necessario al tester per eseguire 4 test case è: 3,5h * 4 = 14 ore b) Non è corretta c) Non è corretta d) Non è corretta	FL-5.1.4	K3	1
32	b	TC1 raggiunge la copertura più alta (4/7 - Req1, Req3, Req4 e Req7), quindi dovrebbe essere eseguito per primo. Req2, Req5 e Req6 non sono ancora coperti. Il test case successivo che raggiunge la copertura aggiuntiva più alta dei requisiti rimanenti è TC3, che copre 2 dei 3 requisiti (Req5 e Req6). Quindi, TC3 dovrebbe essere eseguito come secondo. Ora l'unico requisito non ancora coperto è Req2, che è coperto da TC4. Di conseguenza, TC4 dovrebbe essere eseguito come terzo test case. Infine, l'ultimo test case eseguito sarà TC2. Quindi: a) Non è corretta b) È corretta c) Non è corretta d) Non è corretta	FL-5.1.5	K3	1

© International Software Testing Qualifications Board

33	c	a) Non è corretta. I quadranti del testing non hanno nulla a che fare con la descrizione delle relazioni tra i livelli di test b) Non è corretta. I quadranti del testing non possono aiutare a valutare alcun tipo di copertura c) È corretta. I quadranti del testing consentono ai manager e agli altri stakeholder di comprendere le relazioni tra i tipi di test, le attività che supportano (supportare il team o criticare il prodotto) e il punto di vista su cui si concentrano (orientato al business o alla tecnologia) d) Non è corretta. I quadranti del testing non sono un modello psicologico	FL-5.1.7	K2	1
34	b	La valutazione del rischio può utilizzare un approccio quantitativo o qualitativo, o un mix di entrambi. Nell'approccio quantitativo il livello di rischio viene calcolato come moltiplicazione della probabilità di rischio e dell'impatto del rischio. Quindi, Livello di rischio = Probabilità di rischio * Impatto del rischio Quindi, Impatto del rischio = Livello di rischio / Probabilità di rischio. Nel nostro caso, Impatto del rischio = $1.000 / 50% = $1.000 / 0,5 = $2.000. Quindi: a) Non è corretta b) È corretta c) Non è corretta d) Non è corretta	FL-5.2.1	K1	1
35	b, e	a) Non è corretta. Lo Scope Creep è un esempio di rischio di progetto legato a problemi tecnici b) È corretta. Un'architettura scadente è un esempio di rischio di prodotto, poiché si riferisce a una caratteristica del prodotto c) Non è corretta. La riduzione dei costi è un esempio di rischio di progetto, legato a problemi organizzativi d) Non è corretta. Lo scarso supporto da parte degli strumenti è un esempio di rischio di progetto legato a problemi tecnici e) È corretta. Il tempo di risposta troppo lungo è un esempio di rischio di prodotto, poiché si riferisce a una caratteristica del prodotto	FL-5.2.2	K2	1

© International Software Testing Qualifications Board

36	c	a) Non è corretta. Tracciare l'avanzamento dei test e identificare le aree che richiedono ulteriore attenzione è un esempio di supporto al controllo continuo del testing. Questo è uno degli obiettivi dei test report b) Non è corretta. Fornire informazioni sui test eseguiti, sui loro risultati e su eventuali problemi o difetti riscontrati è un esempio di riepilogo delle attività di test eseguite a un determinato livello di test. Questo è uno degli obiettivi dei test report c) È corretta. Fornire informazioni sui difetti è lo scopo di un defect report, non di un test report d) Non è corretta. Fornire informazioni sul testing pianificato per il periodo successivo è uno degli obiettivi dei test report	FL-5.3.2	K2	1

© International Software Testing Qualifications Board

37	d	a) Non è corretta. Il risk management consiste nell'analisi del rischio e nel controllo del rischio. Nessuna di queste attività supporta il ri-assemblaggio dei file che compongono il rilascio, perché queste attività si occupano dei rischi, non degli elementi di configurazione b) Non è corretta. Il monitoraggio dei test si occupa di raccogliere informazioni sul testing. Queste informazioni vengono utilizzate per valutare l'avanzamento dei test e per misurare se i criteri di uscita dei test o le attività di test associate ai criteri di uscita sono stati soddisfatti, ad esempio il raggiungimento degli obiettivi di copertura dei rischi di prodotto, dei requisiti o di altri criteri di accettazione. Il controllo dei test utilizza le informazioni ottenute dal monitoraggio dei test per fornire, sotto forma di direttive di controllo, una guida e le azioni correttive necessarie per ottenere un testing il più efficace ed efficiente possibile. Nessuna di queste attività si occupa della gestione degli elementi di configurazione c) Non è corretta. Il whole team approach si basa sull'abilità del tester di lavorare efficacemente in un contesto di squadra e di contribuire positivamente agli obiettivi del team. Pertanto, si concentra sui problemi relativi al team, non sugli elementi di configurazione d) È corretta. Il configuration management fornisce una disciplina per identificare, controllare e tracciare i prodotti di lavoro. Il configuration management mantiene un registro degli elementi di configurazione modificati quando viene creata una nuova baseline. Grazie al configuration management, è possibile tornare a una baseline precedente per riprodurre i risultati dei test precedenti	FL-5.4.1	K2	1

© International Software Testing Qualifications Board

38	a	a) È corretta. L'aggiunta di queste informazioni consente allo sviluppatore di utilizzare gli stessi dati di input, per cui è più probabile che saranno in grado di riprodurre rapidamente il failure e quindi di identificare più velocemente il difetto b) Non è corretta. L'aggiunta della Priorità non aiuterà a riprodurre il difetto stesso c) Non è corretta. Sebbene alcune di queste informazioni possano essere utili, l'aggiunta dei dump della memoria e delle istantanee del database dopo ogni passaggio è eccessiva, perché la maggior parte di questi artefatti conterrà informazioni inutili per lo sviluppatore e renderà il report meno leggibile. Inoltre, lo sviluppatore dovrà dedicare molto tempo all'analisi di queste informazioni, allungando così il processo di correzione d) Non è corretta. La domanda riguardava la possibilità di aiutare lo sviluppatore a riprodurre il difetto osservato per una specifica configurazione dell'ambiente	FL-5.5.1	K3	1

© International Software Testing Qualifications Board

39	b	Considerando ognuna delle categorie di strumenti elencate: i. Strumenti di collaborazione - facilitano la comunicazione. La comunicazione non include la facilitazione dell'esecuzione dei test. ii. Strumenti DevOps - supportano la pipeline di delivery DevOps, il tracciamento del flusso di lavoro, i processi di creazione automatizzati e la CI/CD. La pipeline di consegna e la CI/CD includono entrambi la facilitazione dell'esecuzione dei test, ad esempio il testing di componente per la CI. iii. Strumenti di management - aumentano l'efficienza del processo di test facilitando il management del SDLC, dei requisiti, dei test, dei difetti e della configurazione. Il management di questi elementi non include la facilitazione dell'esecuzione dei test. iv. Strumenti di testing non funzionale - consentono al tester di eseguire testing non funzionale difficile o impossibile da eseguire manualmente. Il testing non funzionale può includere sia testing statico sia testing dinamico, compresa l'esecuzione dei test. v. Strumenti di progettazione e implementazione dei test - facilitano la generazione di test case, dati di test e test procedure. La generazione di questo testware non include la facilitazione dell'esecuzione dei test. Quindi: a) Non è corretta b) È corretta. Sia gli strumenti DevOps (ii) che gli strumenti di testing non funzionale (iv) facilitano l'esecuzione dei test c) Non è corretta d) Non è corretta	FL-6.1.1	K2	1

© International Software Testing Qualifications Board

40	c	a) Non è corretta. L'individuazione di ulteriori difetti ad alta gravità sarebbe un vantaggio della test automation, piuttosto che un rischio b) Non è corretta. La fornitura di misure troppo complicate da ricavare per gli esseri umani è normalmente considerata un vantaggio della test automation c) È corretta. Se la test automation è incompatibile con la piattaforma di sviluppo, non sarà in grado di integrarla e, ad esempio, di passare gli input di test all'oggetto di test e di ricevere i risultati dei test dall'oggetto di test d) Non è corretta. La riduzione sostanziale dei tempi di esecuzione dei test sarebbe normalmente considerata un beneficio fornito dalla test automation	FL-6.2.1	K1	1

© International Software Testing Qualifications Board