Simulazione d’esame - Risposte

Simulazione d'esame set C
Versione 1.0

ISTQB® Certified Tester Syllabus Foundation Level

Compatibile con il Syllabus versione 1.03

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 7
2 8
3 9
4 9
5 11
6 12
7 13
8 14
9 16
10 17
11 18
12 19
13 20
14 21
15 22
16 23
17 24
18 25
19 26
20 27
21 28
22 29
23 30
24 31
25 31
26 32
27 33
28 34
29 35
30 36
31 37
32 38
33 38
34 39
35 39
36 40
37 41
38 42
39 43
40 44

© 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	b	FL-1.1.1	K1	1
2	c	FL-1.1.2	K2	1
3	b	FL-1.3.1	K2	1
4	b, e	FL-1.4.1	K2	1
5	a	FL-1.4.3	K2	1
6	c	FL-1.4.5	K2	1
7	b	FL-1.5.2	K1	1
8	a	FL-1.5.3	K2	1
9	d	FL-2.1.2	K1	1
10	d	FL-2.1.3	K1	1
11	b	FL-2.1.5	K2	1
12	c	FL-2.1.6	K2	1
13	d	FL-2.2.1	K2	1
14	b	FL-2.2.3	K2	1
15	d	FL-3.1.3	K2	1
16	a	FL-3.2.1	K1	1
17	b	FL-3.2.4	K2	1
18	b	FL-3.2.5	K1	1
19	c	FL-4.1.1	K2	1
20	c	FL-4.2.1	K3	1

Numero di domanda (#)	Risposta corretta	LO	Livello K	Punti
21	d	FL-4.2.2	K3	1
22	d	FL-4.2.3	K3	1
23	a	FL-4.2.4	K3	1
24	c	FL-4.3.2	K2	1
25	a	FL-4.3.3	K2	1
26	b	FL-4.4.1	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-5.1.1	K2	1
31	c	FL-5.1.4	K3	1
32	a	FL-5.1.5	K3	1
33	b	FL-5.1.6	K1	1
34	d	FL-5.1.7	K2	1
35	c	FL-5.2.3	K2	1
36	b	FL-5.3.2	K2	1
37	d	FL-5.4.1	K2	1
38	b	FL-5.5.1	K3	1
39	d	FL-6.1.1	K2	1
40	d	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	b	a) Non è corretta. Validare che i requisiti documentati siano soddisfatti non è corretto, in quanto la validazione riguarda il soddisfacimento dei requisiti e delle aspettative dell'utente, mentre la verifica riguarda il soddisfacimento dei requisiti specificati, quindi sarebbe corretto se sostituissimo "validare" con "verificare" b) È corretta. Causare failure e identificare difetti è probabilmente l'obiettivo più comune del testing dinamico c) Non è corretta. Causare errori e identificare le root cause non è corretto, perché i tester non causano gli errori, ma cercano di causare failure. Gli errori sono tipicamente commessi dagli sviluppatori (e non possono essere realmente causati) e si traducono in difetti, che i tester cercano di identificare direttamente attraverso il testing statico o indirettamente attraverso i failure con il testing dinamico. L'identificazione delle root cause è utile, ma fa parte del debugging, che è un'attività separata dal testing d) Non è corretta. Verificare che l'oggetto di test soddisfi le aspettative dell'utente non è corretto, poiché la verifica si occupa di controllare che i requisiti specificati (documentati) siano soddisfatti, mentre la validazione si occupa di soddisfare i requisiti e le aspettative dell'utente, quindi sarebbe corretto se sostituissimo "verificare" con "validare"	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. Il testing dinamico causa failure (dai quali è possibile individuare e correggere i difetti). Tuttavia, il debugging si occupa di individuare i difetti e di correggerli. Pertanto, il debugging non risolve i failure b) Non è corretta. Sia il testing che il debugging contribuiscono a migliorare la qualità dell'oggetto di test, quindi dovrebbero essere considerati entrambi positivamente. Il debugging è generalmente considerato un'attività positiva, in quanto si tratta di correggere qualcosa. Il testing dinamico comporta l'intenzionalità di far fallire l'oggetto di test, motivo per cui alcune persone lo considerano un'attività negativa, ma si tratta di una visione molto ristretta (e non di una visione tipica dei tester). Sono possibili test case sia positivi che negativi. I test case positivi verificano che l'oggetto di test esegua correttamente ciò che si suppone debba fare, mentre il testing negativo verifica che l'oggetto di test non esegua ciò che non si suppone debba fare c) È corretta. Il testing determina l'esistenza dei difetti direttamente attraverso l'osservazione del difetto nelle review (o da parte di uno strumento nell'analisi statica), o indirettamente causando un failure nel testing dinamico. Il debugging è un'attività separata dal testing (normalmente eseguito dagli sviluppatori) e si occupa di individuare i difetti (solo per il testing dinamico) e di correggerli d) Non è corretta. Le cause dei difetti sono tipicamente errori umani. Il testing trova i difetti direttamente attraverso il testing statico o indirettamente generando failure nel testing dinamico, mentre il debugging risolve i difetti. Quindi, il testing non trova le cause dei difetti e il debugging non risolve le cause dei difetti	FL-1.1.2	K2	1

© International Software Testing Qualifications Board

3	b	"L'assenza di difetti è un'idea sbagliata" riguarda l'idea che assicurare la correttezza in conformità con i requisiti (cioè verificare l'assenza di difetti di implementazione) non garantisce la soddisfazione dell'utente con il sistema. Per affrontare questo aspetto è necessario validare che il sistema soddisfi le esigenze e le aspettative degli utenti, che soddisfi gli obiettivi di business e che superi le prestazioni dei sistemi concorrenti. a) Non è corretta. Il principio "il testing mostra la presenza di difetti, ma non la loro assenza" spiega che mentre il testing può rilevare l'esistenza di difetti nell'oggetto di test, non è possibile dimostrare che non ci sono difetti e, quindi, garantire la sua correttezza. Pertanto, spiegare che non è possibile per il testing dimostrare l'assenza di difetti affronterebbe parzialmente questo principio, non "l'assenza di difetti è un'idea sbagliata" b) È corretta. Supportando l'utente finale nell'esecuzione del testing di accettazione, dovrebbe essere possibile validare che il sistema soddisfi le esigenze e le aspettative degli utenti c) Non è corretta. Non è possibile garantire che non rimangano difetti di implementazione nel sistema consegnato, poiché il principio "il testing mostra la presenza di difetti, ma non la loro assenza" spiega che mentre il testing può rilevare l'esistenza di difetti nell'oggetto di test, non è possibile dimostrare l'assenza di difetti e, quindi, garantire la sua correttezza d) Non è corretta. Modificare i test che non generano failure per garantire che rimangano pochi difetti è un modo per affrontare il principio "i test perdono di efficacia". Questo principio riguarda l'idea che la ripetizione di test identici su un codice inalterato difficilmente porterà alla scoperta di nuovi difetti e quindi la modifica dei test può essere essenziale. Ciò non validerà che il sistema soddisfi le esigenze e le aspettative degli utenti	FL-1.3.1	K2	1
4	b, e	Data la seguente descrizione dell'analisi dei 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
		Per identificare le funzionalità che richiedono un testing, si analizza la base di test e si definiscono le condizioni di test, che vengono poi prioritizzate insieme ai rischi correlati. L'identificazione sistematica delle condizioni di test come elementi di copertura spesso implica l'uso di tecniche di test sia durante l'analisi dei test che come parte dell'attività di progettazione dei test. Dalla descrizione di cui sopra si evince che le tecniche di test sono spesso utilizzate nelle attività di analisi e progettazione dei test. L'analisi ai valori limite e il partizionamento di equivalenza sono tecniche di test. a) Non è corretta. È improbabile che l'implementazione dei test comporti l'uso di tecniche di test, in quanto si occupa principalmente di assemblare i test case in test procedure, mentre le tecniche di test creano i test case b) È corretta. La progettazione dei test implica probabilmente l'uso di tecniche di test per creare test case a partire dalle condizioni di test e dagli elementi di copertura c) Non è corretta. È improbabile che l'esecuzione dei test implichi l'uso di tecniche di test, in quanto riguarda principalmente l'esecuzione di procedure di test (e quindi di test case), mentre le tecniche di test creano test case d) Non è corretta. È improbabile che il monitoraggio dei test implichi l'uso di tecniche di test. Il monitoraggio dei test riguarda soprattutto i controlli in corso per garantire il rispetto del piano, mentre le tecniche di test creano test case e) È corretta. L'analisi dei test comporta probabilmente l'uso di tecniche di test per identificare le condizioni di test			

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
5	a	Considerando ciascuna delle attività di test elencate e il relativo testware prodotto: A. Analisi dei test - condizioni di test prioritizzate (4) (ad esempio, criteri di accettazione) e defect report per i difetti identificati nella base di test B. Progettazione dei test - test case prioritizzati, test charter, elementi di copertura (1), requisiti dei dati di test e requisiti dell'ambiente di test C. Implementazione dei test: test procedure, test script automatizzati, test suite, dati di test, schedulazione di esecuzione dei test (3) ed elementi dell'ambiente di test come stub, driver, simulatori e virtualizzazioni dei servizi D. Completamento dei test – test completion report, lezioni apprese documentate, elementi di azione per il miglioramento e richieste di modifica (2) (come elementi del product backlog) Quindi: a) È corretta. L'abbinamento corretto è: 1B, 2D, 3C, 4A b) Non è corretta c) Non è corretta d) Non è corretta	FL-1.4.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
6	c	a) Non è corretta. Sebbene sia corretto affermare che nello sviluppo software Agile, alcuni dei compiti del test management possono essere gestiti dal team Agile stesso, il ruolo del testing non è principalmente responsabilità di un singolo individuo esterno al team. È invece più probabile che il testing venga eseguito da diversi membri del team, seguendo il whole team approach b) Non è corretta. Il ruolo del test management coinvolge principalmente le attività relative alla pianificazione dei test, al monitoraggio e al controllo dei test e al completamento dei test. Quindi, anche se questa affermazione è parzialmente corretta, è sbagliato dire che il ruolo del testing è principalmente responsabile del monitoraggio e del controllo dei test c) È corretta. Nello sviluppo software Agile, alcune delle attività del test management possono essere gestite dal team Agile stesso. Tuttavia, per le attività di test che si estendono a più team all'interno di un'organizzazione, i test manager esterni al team di sviluppo possono svolgere questi compiti d) Non è corretta. Il ruolo del test management coinvolge principalmente le attività relative alla pianificazione dei test, al monitoraggio e al controllo dei test e al completamento dei test, mentre il ruolo del testing è responsabile principalmente degli aspetti tecnici e ingegneristici del testing, come l'analisi dei test, la progettazione dei test, l'implementazione dei test e l'esecuzione dei test. Pertanto, il ruolo del test management non è normalmente responsabile dell'analisi dei test e della progettazione dei test, anche se è corretto dire che il ruolo del testing è principalmente responsabile dell'implementazione e dell'esecuzione dei test	FL-1.4.5	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. Nel whole team approach, i tester svolgono un ruolo fondamentale condividendo la loro esperienza nel testing con il team e guidando lo sviluppo del prodotto. Collaborano con gli altri membri del team per raggiungere i livelli di qualità desiderati e lavorano con i rappresentanti di business per creare test di accettazione. I tester collaborano anche con gli sviluppatori per determinare la strategia di test ottimale e gli approcci all'automazione b) È corretta. Sfruttando al meglio le diverse competenze di ciascun membro del team, il whole team approach favorisce dinamiche di squadra superiori, promuove una comunicazione e una collaborazione solide e genera un effetto sinergico che va a vantaggio dell'intero progetto c) Non è corretta. Il whole team approach consente a qualsiasi membro del team con le competenze e le conoscenze necessarie di svolgere qualsiasi compito, quindi i membri specializzati del team non sono un vantaggio di questo approccio d) Non è corretta. Non ci sono indicazioni specifiche sulle dimensioni ottimali dei team che utilizzano il whole team approach e non c'è alcuna considerazione sul fatto che i team più grandi siano migliori	FL-1.5.2	K1	1

© International Software Testing Qualifications Board

8	a	a) È corretta. Il vantaggio principale dell'indipendenza nel testing è che i tester hanno maggiori probabilità di identificare diversi tipi di failure e difetti rispetto agli sviluppatori, a causa della loro diversità di background, di punti di vista tecnici e di potenziali pregiudizi, compresi quelli cognitivi. Tuttavia, il principale svantaggio dell'indipendenza nel 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 b) Non è corretta. La familiarità di uno sviluppatore con il codice non significa che raramente trovi difetti, ma piuttosto che questa familiarità consente di trovare efficacemente molti difetti nel proprio codice. Inoltre, più del fatto che gli sviluppatori e i tester abbiano un background comune, è il fatto che gli sviluppatori abbiano un background diverso da quello dei tester ad essere normalmente indicato come la ragione per cui i tester e gli sviluppatori trovano diversi tipi di difetti c) Non è corretta. Il testing può essere condotto a diversi livelli di indipendenza, da nessuna forma di indipendenza se condotti dall'autore a un'indipendenza molto elevata se condotti da tester esterni all'organizzazione. Nella maggior parte dei progetti si utilizzano più livelli di indipendenza: gli sviluppatori eseguono il testing di componente e di integrazione dei componenti, il team di test esegue il testing di sistema e di integrazione dei sistemi e i rappresentanti di business eseguono il testing di accettazione. Quindi, i tester possono far parte del team dello sviluppatore e non devono necessariamente provenire dall'esterno dell'organizzazione. La conoscenza del dominio di applicazione cambierà da caso a caso e non dipende dal livello di indipendenza d) Non è corretta. Il testing può essere condotto a diversi livelli di indipendenza, che vanno da nessuna forma di indipendenza se	FL-1.5.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		condotti dall'autore a un'indipendenza molto elevata se condotti da tester esterni all'organizzazione, con i tester esterni al team dello sviluppatore generalmente più indipendenti di quelli interni al team. Tuttavia, c'è maggior ragione di credere che i tester esterni al team siano probabilmente più isolati dagli sviluppatori e quindi abbiano maggiori probabilità di essere incolpati per i ritardi nel rilascio del prodotto			

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
9	d	a) 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, in questo caso stiamo cercando di equiparare i livelli di test con i livelli di sviluppo e, sebbene conosciamo cosa si intende per "livelli di test", non esiste un'interpretazione comune del termine "livello di sviluppo" b) Non è corretta. Ogni attività dello sviluppo software ha una corrispondente attività di test, ma gli obiettivi di test sono molto diversi. Ad esempio, un obiettivo di test potrebbe essere quello di garantire che un oggetto di test aderisca a un requisito contrattuale che prevede l'esecuzione di un certo tipo di testing prima della consegna. In questo caso non c'è motivo di avere un corrispondente obiettivo di sviluppo c) Non è corretta. Il quality control si applica a tutte le attività di sviluppo, il che significa che ogni attività dello sviluppo software ha una corrispondente attività di test. Tuttavia, la stessa simmetria non si applica al testing e alle attività degli utenti. Ad esempio, per alcuni sistemi è difficile anche solo identificare gli utenti finali. Inoltre, alcune attività di test si concentrano sugli sviluppatori (ad esempio, il testing per una facile manutenibilità), senza alcun aspetto legato all'utente d) È corretta. Il quality control si applica a tutte le attività di sviluppo, il che significa che a ogni attività dello sviluppo software corrisponde un'attività di test	FL-2.1.2	K1	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
10	d	a) Non è corretta. Il Component Test-Driven Development non è un esempio corretto di approccio test-first allo sviluppo b) Non è corretta. L'Integration Test-Driven Development non è un esempio corretto di approccio test-first allo sviluppo c) Non è corretta. Il System Test-Driven Development non è un esempio corretto di approccio test-first allo sviluppo d) È corretta. L'Acceptance Test-Driven Development (ATDD) è un noto esempio di approccio test-first allo sviluppo	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
11	b	a) Non è corretta. Le pratiche coinvolte nel testing shift-left mirano a implementare un maggior numero di attività di testing nelle prime fasi del ciclo di vita dello sviluppo, raffigurando il SDLC come se si muovesse da sinistra a destra. Non esiste il lato sinistro del processo di test b) È corretta. Lo shift-left sottolinea l'importanza di iniziare il testing quanto prima nel ciclo di vita dello sviluppo software (SDLC). L'implementazione del testing shift-left richiede una formazione aggiuntiva e un aumento dell'effort e dei costi durante le prime fasi del SDLC, tuttavia i risparmi complessivi dovrebbero essere maggiori c) Non è corretta. Sebbene i test di componente e di integrazione dei componenti automatizzati per il testing di regressione siano generalmente validi, la creazione di questi test è normalmente responsabilità degli sviluppatori e, se si segue un approccio di continuous integration/continuous delivery (CI/CD), questi test saranno stati inviati insieme al codice. In alcune situazioni il tester può automatizzare i test per il testing di regressione e talvolta anche i test di componente e di integrazione dei componenti, ma questo non fa parte di un approccio "shift-left" che sposta il testing all'inizio del SDLC d) Non è corretta. La formazione dei tester per l'esecuzione di attività nelle prime fasi del SDLC sosterrebbe l'approccio "shift-left", sottolineando l'importanza di iniziare il testing nelle prime fasi del SDLC. Tuttavia, l'automazione di un maggior numero di attività di test da eseguire successivamente nel SDLC non fa parte di un approccio "shift-left"	FL-2.1.5	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
12	c	a) Non è corretta. Uno degli scopi delle retrospettive è quello di identificare potenziali miglioramenti del processo che, se messi in pratica, dovrebbero portare a un aumento della qualità dei futuri risultati del processo di sviluppo (oggetti di test). Quindi, è probabile che questo si verifichi come risultato di una retrospettiva b) Non è corretta. Uno dei vantaggi delle retrospettive per il testing è l'aumento dell'efficienza dei test attraverso i miglioramenti del processo. Quindi, è probabile che questo si verifichi come risultato di una retrospettiva c) È corretta. I partecipanti alle retrospettive includono tipicamente tester, sviluppatori, architetti, product owner e business analyst, ma gli utenti finali sono raramente invitati o partecipano a questi meeting – ed è altrettanto improbabile che ricevano alcun tipo di report da questi ultimi. Pertanto, è molto improbabile che imparino e capiscano di più sui processi di sviluppo e di test attraverso le retrospettive d) Non è corretta. Tra i vantaggi delle retrospettive per il testing c'è il miglioramento della qualità del testware (compresi i test script automatizzati) grazie alle review congiunte con gli sviluppatori. Quindi, è probabile che ciò avvenga come risultato di una retrospettiva	FL-2.1.6	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
13	d	a) Non è corretta. Il testing di componente (chiamato anche unit testing) comporta il testing di singoli componenti in modo isolato ed è per lo più una verifica rispetto a una specifica, piuttosto che una validazione rispetto alle esigenze dell'utente. Tuttavia, questo tipo di testing non viene normalmente eseguito dai tester, poiché gli sviluppatori di solito lo eseguono nel loro ambiente di sviluppo b) Non è corretta. Il testing di integrazione dei componenti comporta il testing delle interfacce e delle interazioni tra i componenti ed è per lo più una verifica rispetto a una specifica, piuttosto che una validazione rispetto alle esigenze dell'utente. Tuttavia, questo tipo di testing non viene normalmente eseguito dai tester, in quanto di solito lo eseguono gli sviluppatori c) Non è corretta. Il testing di integrazione dei sistemi esamina le interfacce con altri sistemi e servizi esterni ed è per lo più una verifica rispetto a una specifica, piuttosto che una validazione rispetto alle esigenze dell'utente. Anche questo tipo di testing viene eseguito più spesso dai tester d) È corretta. Il testing di accettazione si concentra sul validare che il sistema soddisfi le esigenze di business dell'utente e sia pronto per la distribuzione. Idealmente, questo testing viene eseguito dagli utenti finali	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	b	a) Non è corretta. È necessario eseguire il testing confermativo per verificare che gli aggiornamenti abbiano portato a un'implementazione corretta; tuttavia, sarebbe opportuno eseguire poi il testing di regressione per garantire che non siano stati introdotti o scoperti difetti in aree del sistema non modificate b) È corretta. Il testing confermativo verificherà che gli aggiornamenti abbiano portato a un'implementazione corretta, e quindi il testing di regressione verrà utilizzato per garantire che non siano stati introdotti o scoperti difetti in aree del sistema non modificate c) Non è corretta. Il testing di regressione dovrebbe essere utilizzato per garantire che non siano stati introdotti o scoperti difetti nelle aree del sistema rimaste invariate al momento dell'aggiornamento, ma è anche necessario eseguire il testing confermativo per verificare che gli aggiornamenti abbiano portato a un'implementazione corretta d) Non è corretta. Il testing confermativo verificherà che gli aggiornamenti abbiano portato a un'implementazione corretta, mentre il testing di regressione servirà a garantire che non siano stati introdotti o scoperti difetti in aree del sistema non modificate. Tuttavia, quando viene eseguito (cioè quando è necessario testare un aggiornamento), il testing confermativo precede il testing di regressione	FL-2.2.3	K2	1

© International Software Testing Qualifications Board

15	d	Considerando ciascuno degli esempi di difetti elencati: i. Due diverse parti della specifica di progetto non concordano a causa della complessità del progetto - questo è un esempio di difetto di specifica, che include incoerenze, ambiguità, contraddizioni, omissioni, imprecisioni e duplicazioni, che possono essere più facilmente trovate con il testing statico ii. Un tempo di risposta troppo lungo fa perdere la pazienza agli utenti - questo è un esempio di difetto del tempo di risposta, che può essere rilevato solo nella pratica, eseguendo il programma e misurando il tempo di risposta, che può essere trovato più facilmente con il testing dinamico iii. Un percorso nel codice non può essere raggiunto durante l'esecuzione - questo è un esempio di difetto di codifica, che include variabili con valori non definiti, variabili non dichiarate, codice duplicato o irraggiungibile e un'eccessiva complessità del codice, che può essere più facilmente individuato con il testing statico iv. Una variabile viene dichiarata ma non viene mai utilizzata successivamente nel programma - questo è un esempio di difetto di codifica, che include variabili con valori non definiti, variabili non dichiarate, codice duplicato o irraggiungibile e un'eccessiva complessità del codice, che può essere più facilmente individuato con il testing statico v. La quantità di memoria necessaria al programma per generare un report è troppo elevata - questo è un esempio di difetto delle prestazioni, che può essere rilevato solo nella pratica, eseguendo il programma e misurando la memoria utilizzata, che può essere trovato più facilmente con il testing dinamico Quindi: a) Non è corretta b) Non è corretta	FL-3.1.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		c) Non è corretta d) È corretta. L'abbinamento corretto per il testing statico è i, iii e iv			
16	a	a) È corretta. Ottenere un feedback precoce e frequente dagli stakeholder nel processo di sviluppo del software può essere molto vantaggioso. 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 b) Non è corretta. Il feedback proviene dagli stakeholder, ed è improbabile che il loro feedback migliori la comprensione dei propri requisiti utente c) Non è corretta. Ottenere un feedback precoce e frequente degli stakeholder nel processo di sviluppo del software può essere molto vantaggioso. 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. Tuttavia, il fatto che le modifiche ai requisiti possano essere comprese e implementate prima, non significa che siano incoraggiate modifiche illimitate ai requisiti d) Non è corretta. Il feedback proviene dagli stakeholder e non riguarda la comunicazione agli utenti finali. Le comunicazioni agli utenti finali potrebbero includere la comunicazione di quali requisiti non saranno implementati prima del rilascio, ma idealmente questo non dovrebbe accadere affatto	FL-3.2.1	K1	1

© International Software Testing Qualifications Board

17	b	Prendendo in considerazione ciascuno dei tipi di review elencati: 1. Review tecnica - Questo tipo di review è eseguita da reviewer tecnicamente qualificati e guidata da un moderatore. Gli obiettivi sono ottenere il consenso e prendere decisioni su problemi tecnici, valutando al contempo la qualità e creando la fiducia nel prodotto di lavoro, generando nuove idee, motivando e consentendo agli autori di migliorare e rilevando le anomalie 2. Review informale - L'obiettivo principale è individuare le anomalie. Il processo non è definito e non richiede un output formale documentato 3. Ispezione - È il tipo di review più formale e segue il processo di review generico completo. L'obiettivo principale è quello di rilevare il maggior numero di anomalie, mentre altri obiettivi sono valutare la qualità e creare la fiducia nel prodotto di lavoro, motivare e consentire agli autori di migliorare e raccogliere metriche che possono essere utilizzate per migliorare il ciclo di vita dello sviluppo software (SDLC), compreso il processo di ispezione. L'autore non può essere il review leader o lo scribe 4. Walkthrough - Guidato dall'autore, questo tipo di review ha vari obiettivi, come valutare la qualità e creare fiducia nel prodotto di lavoro, formare i reviewer, ottenere il consenso, generare nuove idee, motivare e consentire agli autori di migliorare e rilevare le anomalie. I reviewer possono eseguire una review individuale prima del walkthrough, ma non è obbligatorio A. Include obiettivi quali ottenere il consenso, generare nuove idee e motivare gli autori a migliorare (A) B. L'obiettivo principale è individuare i potenziali difetti e non genera alcun output formale documentato (D)	FL-3.2.4	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		C. L'obiettivo principale è individuare i potenziali difetti e richiede la raccolta di metriche per supportare il miglioramento del processo (C) D. Include obiettivi quali formare i reviewer, ottenere il consenso, generare nuove idee e individuare i potenziali difetti (B) Quindi: a) Non è corretta b) È corretta. L'abbinamento corretto è: 1A, 2D, 3C, 4B c) Non è corretta d) Non è corretta			
18	b	a) Non è corretta. Per garantire il successo delle review, è importante assicurarsi il sostegno del management per il processo di review, ma ciò non significa che debba partecipare come reviewer b) È corretta. Per garantire il successo delle review, è importante suddividere il prodotto di lavoro in parti sufficientemente piccole da poter essere revisionate in tempi ragionevoli, per evitare che i reviewer perdano la concentrazione durante le review individuali o i review meeting c) Non è corretta. Per garantire il successo delle review, è importante definire chiaramente obiettivi e criteri di uscita misurabili, senza valutare i partecipanti d) Non è corretta. Per garantire il successo delle review, è importante suddividere la review in parti più piccole, per evitare che i reviewer perdano la concentrazione durante la review individuale o i review meeting. Pertanto, non dovreste prevedere di trattare un solo documento per ogni review	FL-3.2.5	K1	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
19	c	a) Non è corretta. Nella maggior parte dei casi, per gli stessi oggetti di test possono essere utilizzate sia tecniche di test black-box sia tecniche di test basate sull'esperienza b) Non è corretta. Sia le tecniche di test black-box che quelle basate sull'esperienza possono essere utilizzate su tutti i livelli di test c) È corretta. Le tecniche di test black-box (note anche come tecniche basate sulle specifiche) si basano sull'analisi del comportamento specificato dell'oggetto di test senza fare riferimento alla sua struttura interna. Pertanto, la base di test è solitamente una specifica. Le tecniche di test basate sull'esperienza utilizzano efficacemente la conoscenza e l'esperienza dei tester per la progettazione e l'implementazione dei test case. Ciò significa che il tester, quando progetta i test, può non utilizzare affatto le specifiche d) Non è corretta. Le tecniche di test basate sull'esperienza possono rilevare difetti che potrebbero sfuggire utilizzando le tecniche di test black-box (e white-box). Pertanto, le tecniche di test basate sull'esperienza sono complementari alle tecniche di test black-box e white-box e sia le tecniche di test black-box che quelle basate sull'esperienza possono essere utilizzate in tutti i SDLC	FL-4.1.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
20	c	a) Non è corretta. Questi tre valori raggiungono la copertura completa delle partizioni di equivalenza, ma la copertura completa può essere ottenuta con due valori b) Non è corretta. Questi tre valori raggiungono la copertura completa delle partizioni di equivalenza, ma la copertura completa può essere ottenuta con due valori c) È corretta. Il valore "1" copre "lunghezza non corretta" e "numero di cifre diverse non corretto". Il valore "1234" copre "lunghezza corretta" e "numero di cifre diverse corretto". Questi due valori coprono tutte e quattro le partizioni di equivalenza identificate d) Non è corretta. Questo insieme non copre la partizione di equivalenza "numero di cifre diverse non corretto"	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
21	d	Le partizioni di equivalenza sono: {..., 99, 100}, {101, 102, ..., 198, 199}, {200, 201, ...}. Pertanto, esistono 4 valori limite, che sono: 100, 101, 199 e 200. Nell'analisi ai valori limite (BVA) a 2 valori, per ogni valore limite ci sono due elementi di copertura (il valore limite e il suo vicino più prossimo appartenente alla partizione adiacente). Poiché i vicini più prossimi sono anch'essi valori limite della partizione adiacente, ci sono solo quattro elementi di copertura. Quindi: a) Non è corretta. Solo 100 e 200 sono elementi di copertura validi per la BVA a 2 valori, quindi otteniamo una copertura del 50% b) Non è corretta. Solo 100 e 200 sono elementi di copertura validi per la BVA a 2 valori, quindi otteniamo una copertura del 50% c) Non è corretta. Solo 100 e 101 sono elementi di copertura validi per la BVA a 2 valori, quindi otteniamo una copertura del 50% d) È corretta. 101, 199 e 200 sono elementi di copertura validi per la BVA a 2 valori, quindi otteniamo una copertura del 75%	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	d	a) Non è corretta. La combinazione (T, T, F) non corrisponde a nessuna regola. Questo è un esempio di omissione, non di contraddizione b) Non è corretta. La combinazione (T, F, T) corrisponde a una sola colonna, R2, quindi non c'è contraddizione c) Non è corretta. Entrambe le combinazioni (T, T, T) e (F, T, T) corrispondono a una sola colonna, R1, quindi non c'è contraddizione d) È corretta. La combinazione (F, F, F) corrisponde sia a R2 che a R3, ma R2 e R3 hanno azioni diverse, quindi questo mostra una contraddizione tra R2 e R3	FL-4.2.3	K3	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
23	a	Le tre transizioni seguenti: "REQUESTING -> CONFIRMED" "WAITING LIST -> CONFIRMED" "WAITING LIST -> END" non possono comparire nello stesso test case, il che suggerisce che sono necessari almeno tre test case. Tutte le altre transizioni possono comparire in combinazione con una o più di queste tre transizioni, quindi abbiamo bisogno di un minimo di tre test case. In realtà, sono possibili solo tre sequenze: TC1: START (Room request) → REQUESTING (Available) → CONFIRMED (Pay) → END TC2: START (Room request) → REQUESTING (Not available) → WAITING LIST (Available) → CONFIRMED (Pay) → END TC3: START (Room request) → REQUESTING (Not available) → WAITING LIST (Cancel) → END Quindi: a) È corretta b) Non è corretta c) Non è corretta d) Non è corretta	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
24	c	Nel testing dei rami gli elementi di copertura sono i rami, che sono rappresentati dai bordi di un grafico del flusso di controllo. Nel grafico del flusso di controllo ci sono 8 edge. Quindi: a) Non è corretta b) Non è corretta c) È corretta d) Non è corretta	FL-4.3.2	K2	1
25	a	a) È corretta. L'esecuzione del solo testing black-box non fornisce una misura dell'effettiva copertura del codice. Le misure di copertura white-box forniscono una misurazione oggettiva della copertura e forniscono le informazioni necessarie per consentire la generazione di test aggiuntivi per aumentare tale copertura e, di conseguenza, aumentare la fiducia nel codice b) Non è corretta. Questa affermazione è falsa e non risponde alla domanda (non ha nulla a che fare con il testing black-box) c) Non è corretta. In generale, non esistono relazioni di sussunzione tra tecniche white-box e black-box d) Non è corretta. Le tecniche white-box sono utilizzate per progettare test basati sull'oggetto di test stesso, mentre le tecniche black-box sono utilizzate per progettare test basati sulle specifiche. Pertanto, non esiste alcuna relazione tra gli elementi di copertura derivati da questi due tipi di tecniche	FL-4.3.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
26	b	a) Non è corretta. Il testing esplorativo utilizza test charter, non un elenco di possibili difetti/failure. Sebbene il testing esplorativo può incorporare l'uso di altre tecniche di test, in questo caso il fault attack è l'opzione più probabile b) È corretta. Si tratta di un elenco di possibili failure. Il fault attack è un approccio metodico all'implementazione dell'error guessing e richiede che il tester crei o acquisisca un elenco di possibili errori, difetti e failure, e che progetti test che identifichino i difetti associati agli errori, espongano i difetti o generino i failure c) Non è corretta. Il tester sta usando una checklist di elementi per supportare il proprio testing. Sia l'error guessing che il testing checklist-based utilizzano tali elenchi, ma in questo caso si tratta di possibili failure, non di condizioni di test, e quindi la tecnica di test PIÙ PROBABILE è il fault attack, che si concentra su errori, difetti e failure d) Non è corretta. La BVA si basa sull'analisi ai valori limite delle partizioni di equivalenza. L'elenco precedente non menziona le partizioni di equivalenza o i loro limiti	FL-4.4.1	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. Sebbene sia vero che il tester può implementare ed eseguire test case dettagliati basati su una checklist, non si spiega come questo possa portare a un aumento della copertura b) Non è corretta. Gli elementi della checklist non dovrebbero essere automatizzati. Ma anche se lo sono, i test script automatizzati eseguono i test sempre nello stesso modo, il che di solito non porta a un aumento della copertura c) Non è corretta. È vero che ogni elemento della checklist dovrebbe essere testato separatamente e in modo indipendente. Ma questo ha un impatto sull'ordine di esecuzione dei test e non ha un impatto sulla copertura ottenuta, e quindi non comporta un aumento della copertura d) È corretta. Se le checklist sono di alto livello, è probabile che si verifichi una certa variabilità nel testing effettivo, con conseguente copertura potenzialmente maggiore ma minore ripetibilità. Se due tester seguono una checklist di elementi di alto livello, ciascuno di essi può utilizzare diversi dati di test, fasi di test ecc. In questo modo, un tester probabilmente coprirà alcune aree non coperte dall'altro tester e ciò si tradurrà in una maggiore copertura	FL-4.4.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
28	b	a) Non è corretta. Questo criterio di accettazione descrive quali regole o normative il sistema deve rispettare (in questo caso, il diritto alla cancellazione). Questo è un esempio di un criterio di accettazione rule-oriented b) È corretta. Questo criterio di accettazione descrive un esempio di scenario che deve essere realizzabile dal sistema. Questo è un esempio di criterio di accettazione scenario-oriented c) Non è corretta. Questa sequenza assomiglia più a una riga di codice che implementa una regola di business. I criteri di accettazione dovrebbero essere scritti in collaborazione con i rappresentanti di business e quindi in un linguaggio a loro comprensibile. Questa sequenza sarà molto probabilmente incomprensibile per questi stakeholder d) Non è corretta. Questo criterio di accettazione descrive quali sono le regole o le normative che il sistema deve rispettare e come viene garantita la conformità. Pertanto, si tratta di un esempio di criterio di accettazione rule-oriented, non di un criterio di accettazione scenario-oriented	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. Vogliamo verificare che gli utenti Speciali abbiano i diritti degli utenti Regolari, quindi dobbiamo testare i diritti di accesso per un utente Speciale, non per un utente Regolare b) Non è corretta. Vogliamo verificare che gli utenti Speciali abbiano i diritti degli utenti Regolari, quindi dobbiamo testare i diritti di accesso per un utente Speciale, non per un utente Regolare c) Non è corretta. Non c'è un piano 5 descritto nei criteri di accettazione. I test case non dovrebbero estendere l'ambito della user story. Ma anche se volessimo eseguire un testing negativo, questo test non è direttamente correlato a AC3 d) È corretta. In questo modo è possibile verificare se un utente Speciale può accedere a piani che sono accessibili a un utente Regolare	FL-4.5.3	K3	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
30	a	a) È corretta. Il test plan può includere i *requisiti* dei dati di test (come parte dell'approccio di test), ma non i dati di test dettagliati per i test case. I dati di test fanno parte dei test case, non del test plan. Inoltre, di solito è impossibile definire tali dati al momento della creazione del test plan, perché non si conosce esattamente l'aspetto dei componenti b) Non è corretta. Uno degli scopi di un test plan è quello di garantire che le attività di test eseguite soddisfino i criteri stabiliti, includendo criteri di ingresso e criteri di uscita. I criteri di copertura del codice sono un esempio di criteri di questo tipo per il livello di test di componente c) Non è corretta. I modelli di documentazione sono il contenuto tipico di un test plan. Questo aiuta a facilitare la comunicazione tra gli stakeholder, definendo un modo standard di comunicare o di reporting d) Non è corretta. Uno degli scopi di un test plan è dimostrare che il testing aderirà alla politica di test e alla strategia di test esistenti, oppure spiegare perché il testing se ne discosterà. Questo è un esempio di spiegazione della deviazione, riguardante i livelli di test che saranno (o non saranno) seguiti	FL-5.1.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
31	c	Dal grafico abbiamo: A(4)=6 e A(3)=8 (le ultime due colonne grigie) Dalla formula ottieniamo: E(5) = (3*A(4) + A(3)) / 4 = (3*6+8) / 4 = 26 / 4 = 6,5 giorni-persona Quindi: a) Non è corretta b) Non è corretta c) È corretta d) Non è corretta	FL-5.1.4	K3	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
32	a	Vogliamo eseguire i test case in base alle loro priorità, ma dobbiamo anche considerare le dipendenze. Se consideriamo solo le priorità, vogliamo eseguire prima TC 5 e TC 7 (priorità più alta), poi TC 1, TC 3 e TC 4 e infine TC 2 e TC 6 (priorità più bassa) Tuttavia, per poter eseguire TC 7, è necessario eseguire prima TC 4. Per poter eseguire TC 5, è necessario eseguire TC 4 e TC 2, ma TC 2 è bloccato da TC 1, che dovrebbe essere eseguito prima di TC 2. Quindi, per eseguire i test case con priorità 1 il prima possibile, i primi cinque test case dovrebbero essere: TC 4 - TC 7 - TC 1 - TC 2 - TC 5. Successivamente, sarà necessario eseguire TC 3, perché ha una priorità più alta di TC 6. La schedulazione completa sarà quindi: TC 4 - TC 7 - TC 1 - TC 2 - TC 5 - TC 3 - TC 6. Quindi, il sesto test case sarà TC 3. Quindi: a) È corretta b) Non è corretta c) Non è corretta d) Non è corretta	FL-5.1.5	K3	1
33	b	a) Non è corretta. Il modello della piramide di test non fornisce informazioni sulle priorità dei test b) È corretta. Il modello della piramide di test mostra che i diversi test hanno diversi livelli di granularità c) Non è corretta. Il modello della piramide di test è indipendente dai criteri di copertura d) Non è corretta. Il modello della piramide di test non mostra alcuna relazione tra i diversi test	FL-5.1.6	K1	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
34	d	a) Non è corretta. I quadranti del testing raggruppano separatamente i livelli e i tipi di test in base a diversi criteri. Non rappresentano alcuna combinazione di livelli e tipi di test e non sono correlati ad alcuna posizione all'interno del ciclo di vita dello sviluppo software. Sia i livelli che i tipi di test sono trattati separatamente nel modello dei quadranti del testing b) Non è corretta. I quadranti del testing raggruppano i livelli e i tipi di test in base a diversi criteri. Non descrivono il grado di granularità dei singoli tipi di test eseguiti a ciascun livello di test. Un modello di questo tipo, relativo ai livelli di test, è chiamato piramide di test c) Non è corretta. L'affermazione è errata, perché in generale qualsiasi tipo di test può essere eseguito a qualsiasi livello di test d) È corretta. I quadranti del testing raggruppano i livelli di test, i tipi di test, le attività, le tecniche di test e i prodotti di lavoro nello sviluppo software Agile. In questo modello, i test possono essere orientati al business o alla tecnologia. I test possono supportare il team (cioè guidare lo sviluppo) o criticare il prodotto (cioè misurare il suo comportamento rispetto alle aspettative). La combinazione di questi due punti di vista determina i quattro quadranti	FL-5.1.7	K2	1
35	c	a) Non è corretta. Il monitoraggio del rischio fa parte del controllo del rischio, non dell'analisi del rischio b) Non è corretta. L'identificazione del rischio in sé non ci permette di implementare le attività di mitigazione del rischio. Le azioni di mitigazione vengono definite durante la fase di controllo del rischio c) È corretta. Questo è un esempio di come l'analisi del rischio influenzi la completezza e l'ambito del testing d) Non è corretta. Gli elementi di copertura vengono ricavati utilizzando tecniche di test, non attraverso l'analisi del rischio	FL-5.2.3	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
36	b	a) Non è corretta. I test progress report sono utilizzati principalmente durante il monitoraggio e il controllo dei test e il completamento dei test, non durante la progettazione dei test b) È corretta. Un test completion report viene preparato durante il completamento dei test, quando un progetto, un livello di test o un tipo di test è completo e quando, idealmente, i suoi criteri di uscita sono stati soddisfatti. Questo report utilizza le informazioni contenute nei test progress report e altri dati c) Non è corretta. I test progress report sono utilizzati principalmente durante il monitoraggio e il controllo dei test e il completamento dei test, non durante l'analisi dei test d) Non è corretta. I test progress report sono utilizzati principalmente durante il monitoraggio e il controllo dei test e il completamento dei test, non durante la pianificazione dei test	FL-5.3.2	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
37	d	a) Non è corretta. Quando un utente segnala un failure del software, grazie all'identificazione univoca dei commit, è possibile ri-assemblare i file della versione del software utilizzata dall'utente (nonché le versioni corrispondenti dei test script) e quindi riprodurre il failure e individuare il difetto più rapidamente b) Non è corretta. Se una modifica all'ambiente di test causa problemi imprevisti durante il testing, il configuration management consente ai tester di tornare a una versione precedente dell'ambiente. In questo modo si garantisce che il testing possa continuare senza essere influenzato dalla modifica c) Non è corretta. Il configuration management assicura che tutta la documentazione identificata (ad esempio, le specifiche dei requisiti) e gli elementi del software siano referenziati senza ambiguità nella documentazione di test (ad esempio, i test plan) d) È corretta. Ciò è garantito dal defect management, non dal configuration management	FL-5.4.1	K2	1

© International Software Testing Qualifications Board

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
38	b	a) Non è corretta. Questo è importante, ma non quanto gli elementi dell'ambiente di test b) È corretta. L'elemento importante che manca è l'identificazione del browser e del dispositivo utilizzato per il testing. Le informazioni sul browser e sul dispositivo sono importanti perché il difetto può essere specifico del browser o del dispositivo. Ad esempio, un pulsante di login può funzionare bene su un browser (o su una versione di un browser specifico) ma non su un altro. Pertanto, le informazioni sul browser e sul dispositivo possono aiutare gli sviluppatori a riprodurre il problema e a trovarne più rapidamente la root cause c) Non è corretta. L'oggetto di test è stato identificato (WebShop v0.99) d) Non è corretta. L'impatto è incluso - si tratta della gravità (alta)	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	d	a) Non è corretta. Gli strumenti di esecuzione e copertura dei test facilitano l'esecuzione automatizzata dei test case e la misurazione della copertura ottenuta con l'esecuzione di tali test case. Tuttavia, questi strumenti non aiutano con l'organizzazione dei difetti e il configuration management b) Non è corretta. Gli strumenti di progettazione e implementazione dei test facilitano la generazione di test case, dati di test e test procedure, ma non aiutano con l'organizzazione dei difetti e il configuration management c) Non è corretta. Gli strumenti di defect management vengono utilizzati per gestire i difetti, ma non sono strumenti di testing e non vengono utilizzati per organizzare i test case o per il configuration management d) È corretta. Gli strumenti di test management aumentano l'efficienza del processo di test facilitando la gestione del ciclo di vita dello sviluppo software (SDLC), dei requisiti, dei test, dei difetti e del configuration management	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	d	a) Non è corretta. "La possibilità di generare test case senza accedere alla base di test" non è possibile. La generazione di test case da parte dei tester o degli strumenti di test richiede l'accesso alla base di test b) Non è corretta. "Il raggiungimento di una maggiore copertura attraverso un assessment più obiettivo" non è un beneficio diretto della test automation. La test automation fornirà un assessment più obiettivo della copertura, ma tale assessment obiettivo non aumenterà la copertura. Solo utilizzando i risultati della copertura per scrivere ulteriori test case è possibile aumentare la copertura c) Non è corretta. "L'aumento dei tempi disponibili di esecuzione dei test con una maggiore potenza di elaborazione" è un'affermazione contraddittoria, in quanto una maggiore potenza di elaborazione normalmente ridurrebbe i tempi di esecuzione, e l'aumento dei tempi di esecuzione non è un vantaggio in quanto il testing richiederebbe più tempo d) È corretta. "La prevenzione degli errori umani grazie a una maggiore coerenza e ripetibilità" è un vantaggio della test automation, poiché quest'ultima non può essere affetta da errori umani. Ad esempio, significa che i test sono derivati in modo coerente dai requisiti, i dati di test sono creati in modo sistematico e i test sono eseguiti da uno strumento nello stesso ordine e con la stessa frequenza	FL-6.2.1	K1	1

© International Software Testing Qualifications Board