

# Simulazione d'Esame - Risposte

Simulazione d'Esame set C

Versione 1.5

## ISTQB® Certified Tester Syllabus Foundation Level

Compatibile con il Syllabus versione 4.0

---

International Software Testing Qualifications Board

---



## Avviso di copyright

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

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

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

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

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

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

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

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

.

## Responsabilità del documento

Exam Working Group ISTQB® è responsabile di questo documento.

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

## Riconoscimenti

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

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

## Storia delle revisioni

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

Versione	Data	Osservazioni
1.5	4 luglio 2024	Versione allineata alla versione inglese V1.5 (4 luglio 2024). Correzione della domanda: #17, #20
1.3	14 marzo 2024	Versione allineata alla versione inglese V1.3 (14 marzo 2024). Correzione della domanda: #20, #25
1.0	16 ottobre 2023	Prima versione

# Indice dei contenuti

Avviso di copyright .....	2
Responsabilità del documento.....	2
Riconoscimenti .....	2
Storia delle revisioni .....	3
Indice dei contenuti .....	4
Introduzione .....	5
Scopo del documento .....	5
Istruzioni .....	5
Chiave di risposta.....	6
Risposte .....	7
1 .....	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

## Introduzione

### Scopo del documento

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

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

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

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

### Istruzioni

In questo documento potete trovare:

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

## Risposta Chiave

Numero 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 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

## Risposte

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
1	b	<p>a) Non è corretta. Validare che i requisiti documentati siano soddisfatti non è corretto, poiché la validazione riguarda il soddisfare i requisiti e le aspettative dell'utente, mentre la verifica riguarda il soddisfare i requisiti specificati. Sarebbe quindi corretto se "validare" fosse sostituito con "verificare"</p> <p>b) È corretta. Causare failure e identificare difetti è probabilmente l'obiettivo più comune del testing dinamico</p> <p>c) Non è corretta. Generare errori e identificare le root cause non è corretto, poiché i tester non causano gli errori, ma cercano di causare failure. Gli errori sono tipicamente commessi dagli sviluppatori (e non possono essere realmente generati) e si traducono in difetti, che i tester cercano di rilevare direttamente attraverso il testing statico, o indirettamente attraverso i failure con il testing dinamico. Identificare le root cause è utile, ma è parte del debugging, che è un'attività separata dal testing</p> <p>d) Non è corretta. Verificare che l'oggetto di test soddisfi le aspettative dell'utente non è corretto, poiché la verifica riguarda il soddisfare i requisiti specificati, mentre la validazione riguarda il soddisfare i requisiti e le aspettative dell'utente. Sarebbe quindi corretto se "verificare" fosse sostituito con "validare"</p>	FL-1.1.1	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
2	c	<p>a) Non è corretta. Il testing dinamico causa failure (dai quali è possibile localizzare e correggere i difetti). Il debugging si occupa di localizzare e correggere i difetti. Pertanto, il debugging non corregge i failure</p> <p>b) Non è corretta. Sia il testing sia 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, perché corregge qualcosa. Il testing dinamico fa fallire in modo intenzionale 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 sia negativi. I test case positivi verificano che l'oggetto di test esegua correttamente quello che si suppone debba fare, mentre il testing negativo verifica che l'oggetto di test non esegua quello che si suppone non debba fare</p> <p>c) È corretta. Il testing determina l'esistenza di difetti, sia direttamente attraverso l'osservazione del difetto nelle review (o da parte di uno strumento di analisi statica), sia indirettamente causando un failure nel testing dinamico. Il debugging è un'attività separata dal testing (normalmente eseguita dagli sviluppatori) e si occupa di localizzare i difetti (solo per il testing dinamico) e correggere i difetti</p> <p>d) Non è corretta. Le cause dei difetti sono generalmente errori umani. Il testing trova i difetti direttamente attraverso il testing statico o indirettamente causando failure nel testing dinamico, mentre il debugging corregge i difetti. Quindi, il testing non individua le cause dei difetti e il debugging non risolve le cause dei difetti</p>	FL-1.1.2	K2	1



3	b	<p>"L'assenza di difetti è un'idea sbagliata" riguarda l'idea che assicurare la correttezza in accordo con i requisiti (cioè verificare l'assenza di difetti di implementazione) non garantisca la soddisfazione dell'utente con il sistema. Per indirizzare 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.</p> <p>a) Non è corretta. Il principio "il testing mostra la presenza di difetti, ma non la loro assenza" spiega che il testing può rilevare l'esistenza di difetti nell'oggetto di test, ma non è possibile dimostrare che non siano presenti difetti e, quindi, garantire la sua correttezza. Pertanto, spiegare che non è possibile per il testing dimostrare l'assenza di difetti indirizzerebbe parzialmente questo principio, non "l'assenza di difetti è un'idea sbagliata"</p> <p>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</p> <p>c) Non è corretta. Non è possibile garantire che non rimangano difetti di implementazione nel sistema rilasciato, 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</p> <p>d) Non è corretta. Modificare i test che non generano failure per garantire che rimangano pochi difetti è un modo per indirizzare il principio "i test perdono di efficacia". Questo principio riguarda l'idea che ripetere test identici su un codice inalterato difficilmente porterà a scoprire nuovi difetti, e quindi modificare i test può essere fondamentale. Questo non validerà che il sistema soddisfi le esigenze e le aspettative degli utenti</p>	FL-1.3.1	K2	1
4	b, e	Si consideri la seguente descrizione dell'analisi dei test:	FL-1.4.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		<p>Per identificare le funzionalità che richiedono un testing, si analizza la base di test e si definiscono le condizioni di test, che vengono poi priorizzate insieme ai rischi correlati. L'identificazione sistematica delle condizioni di test come elementi di copertura spesso implica l'utilizzo di tecniche di test sia durante l'analisi dei test che come parte dell'attività di progettazione dei test.</p> <p>Dalla descrizione di cui sopra si evince che le tecniche di test vengono spesso utilizzate nelle attività di analisi dei test e progettazione dei test.</p> <p>L'analisi ai valori limite e il partizionamento di equivalenza sono tecniche di test.</p> <p>a) Non è corretta. È improbabile che l'implementazione dei test comporti l'utilizzo di tecniche di test, in quanto si occupa principalmente di assemblare i test case in procedure di test, mentre le tecniche di test creano i test case</p> <p>b) È corretta. La progettazione dei test implica probabilmente l'utilizzo di tecniche di test per creare i test case a partire dalle condizioni di test e dagli elementi di copertura</p> <p>c) Non è corretta. È improbabile che l'esecuzione dei test implichi l'utilizzo 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</p> <p>d) Non è corretta. È improbabile che il monitoraggio dei test implichi l'utilizzo 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</p> <p>e) È corretta. L'analisi dei test comporta probabilmente l'utilizzo di tecniche di test per identificare le condizioni di test</p>			

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

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
6	c	<p>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 responsabilità primaria di un singolo individuo esterno al team. È invece più probabile che il testing venga eseguito dai diversi membri del team, seguendo l'approccio whole-team</p> <p>b) Non 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. Quindi, anche se questa affermazione è parzialmente corretta, è sbagliato dire che il ruolo del testing è principalmente responsabile del monitoraggio e controllo dei test</p> <p>c) È corretta. Nello sviluppo software Agile, alcune delle attività di test management possono essere gestite dal team Agile stesso. Tuttavia, per le attività di test che riguardano più team all'interno di un'organizzazione, i Test Manager esterni al team di sviluppo possono svolgere questi compiti</p> <p>d) Non 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, mentre il ruolo del testing è responsabile principalmente degli aspetti tecnici e di ingegneria del testing, come l'analisi dei test, la progettazione dei test, l'implementazione dei test e l'esecuzione dei test. Pertanto, il ruolo di 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 dei test ed esecuzione dei test</p>	FL-1.4.5	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
7	b	<p>a) Non è corretta. Nell'approccio whole-team, i tester svolgono un ruolo vitale 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 di automazione</p> <p>b) È corretta. Sfruttando al meglio le diverse competenze di ogni membro del team, l'approccio whole-team favorisce dinamiche di squadra migliori, promuove una robusta comunicazione e collaborazione, e genera un effetto sinergico che va a vantaggio dell'intero progetto</p> <p>c) Non è corretta. L'approccio whole-team consente a qualsiasi membro del team, con le competenze e le conoscenze necessarie, di svolgere qualsiasi compito, quindi i membri del team specialisti non sono un vantaggio di questo approccio</p> <p>d) Non è corretta. Non esistono indicazioni specifiche sulle dimensioni ottimali dei team che utilizzano l'approccio whole-team, e non esiste alcuna considerazione sul fatto che i team più grandi siano migliori</p>	FL-1.5.2	K1	1

8	a	<p>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 dei differenti background, punti di vista tecnici e potenziali pregiudizi (biases), compresi i pregiudizi 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</p> <p>b) Non è corretta. La familiarità di uno sviluppatore con il codice non significa che raramente rilevi difetti, ma piuttosto che questa familiarità consenta di trovare in modo efficace molti difetti nel proprio codice. Inoltre, più che gli sviluppatori e i tester abbiano un background comune, è invece indicato che il fatto che gli sviluppatori abbiano un background diverso da quello dei tester sia la ragione per cui i tester e gli sviluppatori rilevino diversi tipi di difetti</p> <p>c) Non è corretta. Il testing può essere condotto a diversi livelli di indipendenza, a partire da nessuna forma di indipendenza per l'autore fino a un'indipendenza alta per tester esterni all'organizzazione. Nella maggior parte dei progetti si utilizzano diversi livelli di indipendenza, dove 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 essere esterni all'organizzazione. La conoscenza del dominio applicativo cambierà da caso a caso e non dipende dal livello di indipendenza</p> <p>d) Non è corretta. Il testing può essere condotto a diversi livelli di indipendenza, a partire da nessuna forma di indipendenza per</p>	FL-1.5.3	K2	1
---	---	--	----------	----	---

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		l'autore fino a un'indipendenza alta per tester esterni all'organizzazione, con tester esterni al team dello sviluppatore generalmente più indipendenti dai tester interni al team. Tuttavia, esiste una maggior ragione di credere che i tester esterni al team siano probabilmente più isolati degli sviluppatori e quindi abbiano maggiori probabilità di essere incolpati per ritardi nel rilascio del prodotto			

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
9	d	<p>a) Non è corretta. Il quality control si applica a tutte le attività di sviluppo, e questo significa che ogni attività di sviluppo software ha una corrispondente attività di test. Tuttavia, in questo caso si sta cercando di equiparare i livelli di test con i livelli di sviluppo e, sebbene si sappia cosa significhi "livelli di test", non esiste un'interpretazione comune del termine "livello di sviluppo"</p> <p>b) Non è corretta. Ogni attività di sviluppo software ha una corrispondente attività di test, ma gli obiettivi di test sono molto differenti. 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 test prima del rilascio. In questo caso non esiste una ragione per avere un corrispondente obiettivo di sviluppo</p> <p>c) Non è corretta. Il quality control si applica a tutte le attività di sviluppo, e questo significa che ogni attività di sviluppo software ha una corrispondente attività di test. Tuttavia, la stessa simmetria non si applica al testing e alle attività utente. Ad esempio, per alcuni sistemi è difficile anche solo identificare gli utenti finali. Inoltre, alcune attività di test si focalizzano sugli sviluppatori (p.e. il testing per una facile manutenibilità), senza alcun aspetto legato all'utente</p> <p>d) È corretta. Il quality control si applica a tutte le attività di sviluppo, e questo significa che per ogni attività dello sviluppo software corrisponde un'attività di test</p>	FL-2.1.2	K1	1



Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
10	d	a) Non è corretta. Component Test-Driven Development non è un esempio corretto di approccio test-first allo sviluppo b) Non è corretta. Integration Test-Driven Development non è un esempio corretto di approccio test-first allo sviluppo c) Non è corretta. System Test-Driven Development non è un esempio corretto di approccio test-first allo sviluppo d) È corretta. Acceptance Test-Driven Development (ATDD) è un esempio conosciuto di approccio test-first allo sviluppo	FL-2.1.3	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
11	b	<p>a) Non è corretta. Le pratiche considerate nel testing shift-left hanno lo scopo di implementare un maggior numero di attività di test nelle prime fasi del ciclo di vita dello sviluppo, rappresentando il ciclo di vita dello sviluppo software (SDLC) da sinistra a destra. Non esiste il lato sinistro del processo di test</p> <p>b) È corretta. Shift-left enfatizza l'importanza di anticipare il testing nelle fasi iniziali del ciclo di vita dello sviluppo software. 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</p> <p>c) Non è corretta. Sebbene i test di componente e di integrazione dei componenti automatizzati per il regression testing siano generalmente validi, la creazione di questi test è normalmente di responsabilità degli sviluppatori e, se si segue un approccio di continuous integration/continuous delivery (CI/CD), questi test saranno stati sottomessi insieme al codice. In alcune situazioni il tester può automatizzare i test per il regression testing 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</p> <p>d) Non è corretta. La formazione dei tester per l'esecuzione di attività nelle prime fasi del SDLC supporterebbe un 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 è parte di un approccio "shift-left"</p>	FL-2.1.5	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
12	c	<p>a) Non è corretta. Uno degli scopi delle retrospettive è 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</p> <p>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</p> <p>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 raramente partecipano a questi meeting – ed è altrettanto improbabile che ricevano alcun tipo di report da questi meeting. Pertanto, è molto improbabile che imparino e capiscano di più sui processi di sviluppo e di test attraverso le retrospettive</p> <p>Non è corretta. Uno dei benefici delle retrospettive per il testing include il miglioramento della qualità del testware (compresi i test script automatizzati) grazie alle review congiunte con gli sviluppatori. Quindi, è probabile che questo avvenga come risultato di una retrospettiva</p>	FL-2.1.6	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
13	d	<p>a) Non è corretta. Il testing di componente (chiamato anche unit testing) comporta il testing di singoli componenti ed è per lo più una verifica rispetto a una specifica, piuttosto che una validazione rispetto alle esigenze dell'utente. Questo testing non viene normalmente eseguito dai tester, poiché gli sviluppatori di solito lo eseguono nel loro ambiente di sviluppo</p> <p>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. Questo testing non viene normalmente eseguito dai tester, in quanto di solito lo eseguono gli sviluppatori</p> <p>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. Questo testing viene di solito eseguito dai tester</p> <p>d) È corretta. Il testing di accettazione si focalizza sul validare che il sistema soddisfi le esigenze di business dell'utente e sia pronto per il rilascio. Generalmente, questo testing viene svolto dagli utenti finali</p>	FL-2.2.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
14	b	<p>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 successivamente il regression testing per garantire che non siano stati introdotti o scoperti difetti in aree del sistema non modificate</p> <p>b) È corretta. Il testing confermativo verificherà che gli aggiornamenti abbiano portato a un'implementazione corretta, e quindi il regression testing verrà utilizzato per garantire che non siano stati introdotti o scoperti difetti in aree del sistema non modificate</p> <p>c) Non è corretta. Il regression testing dovrebbe essere utilizzato per garantire che non siano stati introdotti o scoperti difetti nelle aree del sistema non modificate al momento dell'aggiornamento. Tuttavia, è anche necessario eseguire il testing confermativo per verificare che gli aggiornamenti abbiano portato a un'implementazione corretta</p> <p>d) Non è corretta. Il testing confermativo verificherà che gli aggiornamenti abbiano portato a un'implementazione corretta, mentre il regression testing 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 regression testing</p>	FL-2.2.3	K2	1

15	d	<p>Si consideri ognuno degli esempi di difetti elencati:</p> <ul style="list-style-type: none"> <li>i. Due parti differenti della specifica di progetto sono inconsistenti tra loro a causa della complessità del progetto - questo è un esempio di difetto di specifica, che include inconsistenze, ambiguità, contraddizioni, omissioni, inaccuratezze e duplicazioni, che possono essere più facilmente rilevate con il testing statico</li> <li>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, e che può essere rilevato più facilmente con il testing dinamico</li> <li>iii. Un cammino nel codice non può essere raggiunto durante l'esecuzione - questo è un esempio di difetto di codifica, che include variabili con valori indefiniti, variabili non dichiarate, codice duplicato o non raggiungibile, e un'eccessiva complessità del codice, che può essere più facilmente rilevata con il testing statico</li> <li>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 indefiniti, variabili non dichiarate, codice duplicato o non raggiungibile, e un'eccessiva complessità del codice, che può essere più facilmente rilevata con il testing statico</li> <li>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, e che può essere rilevato più facilmente con il testing dinamico</li> </ul> <p>Quindi:  a) Non è corretta</p>	FL-3.1.3	K2	1
----	---	---	----------	----	---

		<p>b) Non è corretta c) Non è corretta d) È corretta. L'abbinamento corretto per il testing statico è i, iii e iv</p>			
--	--	---	--	--	--

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
16	a	<p>a) È corretta. Ottenere un feedback anticipato e frequente dagli stakeholder nel processo di sviluppo del software può essere molto vantaggioso. Facilita la comunicazione anticipata di potenziali problemi di qualità, può prevenire incomprensioni sui requisiti e assicura che qualsiasi modifica nei requisiti degli stakeholder sia compresa e implementata prima</p> <p>b) Non è corretta. Il feedback proviene dagli stakeholder, ed è improbabile che il loro feedback migliori la comprensione dei propri requisiti degli utenti</p> <p>c) Non è corretta. Ottenere un feedback anticipato e frequente degli stakeholder nel processo dello sviluppo software può essere molto vantaggioso. Facilita la comunicazione anticipata di potenziali problemi di qualità, può prevenire incomprensioni sui requisiti e assicura che qualsiasi modifica nei 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</p> <p>d) Non è corretta. Il feedback proviene dagli stakeholder e non copre la comunicazione agli utenti finali. Le comunicazioni agli utenti finali potrebbero includere quali requisiti non saranno implementati prima del rilascio, ma idealmente questo non dovrebbe accadere affatto</p>	FL-3.2.1	K1	1



17	b	<p>Si consideri ognuno dei tipi di review elencati:</p> <ol style="list-style-type: none"> <li>1. Review tecnica - Questo tipo di review viene eseguita da reviewer tecnicamente qualificati e guidata da un moderatore. Gli obiettivi sono ottenere il consenso e prendere decisioni su problemi tecnici, valutando anche la qualità e creando confidenza nel prodotto di lavoro, generando nuove idee, rilevando anomalie, motivando e permettendo agli autori di migliorare</li> <li>2. Review informale - L'obiettivo principale è rilevare anomalie. Il processo non è definito e non richiede un output formale documentato</li> <li>3. Ispezione - È il tipo di review più formale e segue il processo generico di review completo. L'obiettivo principale è rilevare il maggior numero di anomalie, mentre altri obiettivi sono valutare la qualità e creare confidenza nel prodotto di lavoro, motivare e permettere agli autori di migliorare, 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</li> <li>4. Walkthrough - Guidato dall'autore, questo tipo di review ha diversi obiettivi, come valutare la qualità e creare confidenza nel prodotto di lavoro, educare i reviewer, ottenere il consenso, generare nuove idee, rilevare anomalie, motivare e permettere agli autori di migliorare. I reviewer potrebbero eseguire una review individuale prima del walkthrough, ma non è obbligatorio</li> </ol> <p>Quindi:</p> <ol style="list-style-type: none"> <li>A. Include obiettivi quali ottenere il consenso, generare nuove idee e motivare gli autori a migliorare</li> <li>B. Include obiettivi come educare i reviewer, ottenere il consenso, generare nuove idee e rilevare potenziali difetti</li> </ol>	FL-3.2.4	K2	1
----	---	--	----------	----	---

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
		<p>C. L'obiettivo principale è rilevare potenziali difetti e richiede la raccolta di metriche per supportare il miglioramento del processo</p> <p>D. L'obiettivo principale è rilevare potenziali difetti e genera un output documentato non formale</p> <p>Quindi:</p> <p>a) Non è corretta</p> <p>b) È corretta</p> <p>c) Non è corretta</p> <p>d) Non è corretta</p>			
18	b	<p>a) Non è corretta. Per garantire il successo delle review, è importante assicurarsi il supporto del management per il processo di review, ma questo non significa che dovrebbe partecipare come reviewer</p> <p>b) È corretta. Per garantire il successo delle review, è importante suddividere un prodotto di lavoro di grandi dimensioni in parti sufficientemente piccole da poter essere sottoposte a review in tempi ragionevoli, per prevenire che i reviewer perdano il focus durante le review individuali o i review meeting</p> <p>c) Non è corretta. Per garantire il successo delle review, è importante definire chiaramente obiettivi e criteri di uscita misurabili, senza valutare i partecipanti</p> <p>d) Non è corretta. Per garantire il successo delle review, è importante suddividere la review di documenti di grandi dimensioni in parti più piccole, per evitare che i reviewer perdano il focus durante la review individuale o i review meeting. Pertanto, non si dovrebbe sempre pianificare di coprire un documento completo in ogni review</p>	FL-3.2.5	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
19	c	<p>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</p> <p>b) Non è corretta. Sia le tecniche di test black-box sia le tecniche di test basate sull'esperienza possono essere utilizzate a tutti i livelli di test</p> <p>c) È corretta. Le tecniche di test black-box (note anche come tecniche basate sulle specifiche) si basano sull'analisi del comportamento dell'oggetto di test senza fare riferimento alla sua struttura interna. Pertanto, la base di test è generalmente 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. Questo significa che il tester, quando progetta i test, può non utilizzare le specifiche</p> <p>d) Non è corretta. Le tecniche di test basate sull'esperienza possono rilevare difetti che potrebbero non essere rilevati 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 alle tecniche di test white-box, che possono essere entrambe utilizzate in tutti i cicli di vita dello sviluppo software</p>	FL-4.1.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
20	c	<p>Esistono cinque classi di equivalenza (INPUT) Lunghezza:                      L1: tra 1 e 3                      L2: 4                      L3: &gt; 4                      (INPUT) Cifre:                      D1: IL PIN contiene solo la stessa cifra                      D2: Il PIN contiene almeno due cifre differenti                      (OUTOUT) 2 partizioni:                      G1: PIN corretto (copre L2 e D2)                      G2: PIN non corretto</p> <p>a) È corretta. 112 copre D2, L1, G2; 1111 copre D1, L2, G2; 1234 copre D2, L2, G2, 123456 copre D2, L3, G2. Questi tre valori raggiungono la copertura completa delle partizioni di equivalenza, ed evitano il mascheramento dei difetti</p> <p>b) Non è corretta. 1 copre D1, L1, G2; 123 copre D2, L1, G2; 1111 copre D1, L2, G2, 12345 copre D2, L2, G1</p> <p>c) Non è corretta. Non copre D1. 12 copre D2, L1, G2; 112 copre D2, L1, G2; 1112 copre D2, L2, G1, 11112 copre D2, L3, G2                      Non è corretta. Non copre D2 (e G1). 1 copre D1, L1, G2; 111 copre D1, L1, G2; 1111 copre D1, L2, G2, 11111 copre D1, L3, G2</p>	FL-4.2.1	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
21	d	<p>Le partizioni di equivalenza sono: {..., 99, 100}, {101, 102, ..., 198, 199}, {200, 201, ...}.</p> <p>Pertanto, esistono 4 valori limite: 100, 101, 199 e 200. Nell'analisi ai valori limite (BVA) a 2 valori, per ogni valore limite esistono 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, esistono solo quattro elementi di copertura.</p> <p>Quindi:</p> <ul style="list-style-type: none"> <li>a) Non è corretta. Solo 100 e 200 sono elementi di copertura validi per BVA a 2 valori, quindi si ottiene una copertura del 50%</li> <li>b) Non è corretta. Solo 100 e 200 sono elementi di copertura validi per BVA a 2 valori, quindi si ottiene una copertura del 50%</li> <li>c) Non è corretta. Solo 100 e 101 sono elementi di copertura validi per BVA a 2 valori, quindi si ottiene una copertura del 50%</li> <li>d) È corretta. 101, 199 e 200 sono elementi di copertura validi per BVA a 2 valori, quindi si ottiene una copertura del 75%</li> </ul>	FL-4.2.2	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
22	d	<p>a) Non è corretta. La combinazione (T, T, F) non corrisponde a nessuna regola. Questo è un esempio di omissione, non di contraddizione</p> <p>b) Non è corretta. La combinazione (T, F, T) corrisponde a una sola colonna, R2, quindi non esiste contraddizione</p> <p>c) Non è corretta. Entrambe le combinazioni (T, T, T) e (F, T, T) corrispondono a una sola colonna, R1, quindi non esiste contraddizione</p> <p>d) È corretta. La combinazione (F, F, F) corrisponde sia a R2 che a R3, ma R2 e R3 hanno azioni diverse, quindi evidenzia una contraddizione tra R2 e R3</p>	FL-4.2.3	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
23	a	<p>Le seguenti tre transizioni:                      "REQUESTING -&gt; CONFIRMED"                      "WAITING LIST -&gt; CONFIRMED"                      "WAITING LIST -&gt; END"</p> <p>non possono essere presenti nello stesso test case, e questo suggerisce che sono necessari almeno tre test case. Tutte le altre transizioni possono essere combinate con una o più di queste tre transizioni, quindi sono necessari almeno tre test case. Infatti, sono possibili solo tre sequenze:</p> <ul style="list-style-type: none"> <li>• TC1: START (Room request) → REQUESTING (Available) → CONFIRMED (Pay) → END</li> <li>• TC2: START (Room request) → REQUESTING (Not available) → WAITING LIST (Available) → CONFIRMED (Pay) → END</li> <li>• TC3: START (Room request) → REQUESTING (Not available) → WAITING LIST (Cancel) → END</li> </ul> <p>Quindi:                      a) È corretta                      b) Non è corretta                      c) Non è corretta                      d) Non è corretta</p>	FL-4.2.4	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
24	c	<p>Nel testing dei rami gli elementi di copertura sono i rami, che sono rappresentati dai segmenti orientati di un grafo del flusso di controllo. Nel grafo del flusso di controllo esistono 8 segmenti orientati.</p> <p>Quindi:</p> <ul style="list-style-type: none"> <li>a) Non è corretta</li> <li>b) Non è corretta</li> <li>c) È corretta</li> <li>d) Non è corretta</li> </ul>	FL-4.3.2	K2	1
25	a	<ul style="list-style-type: none"> <li>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 misura oggettiva della copertura e forniscono le informazioni necessarie per consentire la generazione di test aggiuntivi per aumentare questa copertura e, di conseguenza, aumentare la fiducia nel codice</li> <li>b) Non è corretta. Questa affermazione è falsa e non ha nulla a che fare con il testing black-box</li> <li>c) Non è corretta. In generale, non esistono relazioni di sussunzione tra tecniche white-box e tecniche black-box</li> <li>d) Non è corretta. Le tecniche white-box vengono utilizzate per progettare i test basati sull'oggetto di test, mentre le tecniche black-box vengono utilizzate per progettare i test basati sulle specifiche. Pertanto, non esiste alcuna relazione tra gli elementi di copertura derivati da questi due tipi di tecniche</li> </ul>	FL-4.3.3	K2	1



Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
26	b	<p>a) Non è corretta. Il testing esplorativo utilizza i test charter, non una lista di possibili difetti/failure. Sebbene il testing esplorativo possa incorporare l'uso di altre tecniche di test, in questo caso il fault attack è l'opzione più probabile</p> <p>b) È corretta. Si tratta di una lista di possibili failure. Il fault attack è un approccio metodico all'implementazione dell'error guessing e richiede che il tester crei o acquisisca una lista di possibili errori, difetti e failure, e che progetti i test che identifichino i difetti associati agli errori, espongano i difetti o causino i failure</p> <p>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 liste, ma in questo caso è una lista di possibili failure, non di condizioni di test, e quindi la tecnica di test PIÙ PROBABILE è il fault attack, che si focalizza su errori, difetti e failure</p> <p>d) Non è corretta. L'analisi ai valori limite si basa sull'analisi dei limiti delle partizioni di equivalenza. La lista precedente non menziona le partizioni di equivalenza o i loro limiti</p>	FL-4.4.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
27	d	<p>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</p> <p>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, che generalmente non porta a un aumento della copertura</p> <p>c) Non è corretta. È vero che ogni elemento della checklist dovrebbe essere testato separatamente e in modo indipendente. Ma questo impatta l'ordine di esecuzione dei test e non ha un impatto sulla copertura ottenuta, e quindi non comporta un aumento della copertura</p> <p>d) È corretta. Se le checklist sono di alto livello, è probabile che si verifichi una certa variabilità nel testing effettivo, con conseguente <u>copertura potenzialmente maggiore</u> ma minore ripetibilità. Se due tester seguono una checklist di elementi di alto livello, ognuno di questi può utilizzare diversi dati di test, passi del test ecc. In questo modo, un tester probabilmente coprirà alcune aree non coperte dall'altro tester e questo risulterà in una maggiore copertura</p>	FL-4.4.3	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
28	b	<p>a) Non è corretta. Questo criterio di accettazione descrive a quali regole o normative il sistema deve aderire (in questo caso, il diritto alla cancellazione). Questo è un esempio di un criterio di accettazione rule-oriented</p> <p>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</p> <p>c) Non è corretta. Questa sequenza assomiglia a una linea 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 frase sarà molto probabilmente incomprensibile per questi stakeholder</p> <p>d) Non è corretta. Questo criterio di accettazione descrive quali sono le regole o le normative a cui il sistema deve aderire e come viene garantita la conformità. Pertanto, questo è un esempio di criterio di accettazione rule-oriented, non di un criterio di accettazione scenario-oriented</p>	FL-4.5.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
29	d	a) Non è corretta. 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 esiste 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 sarebbe 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

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
30	a	<p>a) È corretta. Il test plan può includere i <i>requisiti</i> dei dati di test (come parte dell'approccio del test), ma non i dati di test dettagliati per i test case. I dati di test sono parte dei test case, non del test plan. Inoltre, è normalmente impossibile definire tali dati al momento della creazione del test plan, perché non si conosce esattamente l'aspetto dei componenti</p> <p>b) Non è corretta. Uno degli scopi di un test plan è aiutare a garantire che le attività di test eseguite soddisfino i criteri stabiliti, includendo i criteri di ingresso e i criteri di uscita. I criteri di copertura del codice sono un esempio di criteri di questo tipo per il livello di test di componente</p> <p>c) Non è corretta. I template della 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 fare reporting</p> <p>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 devierà da questi. Questo è un esempio di spiegazione della deviazione, relativamente ai livelli di test che saranno (o non saranno) seguiti</p>	FL-5.1.1	K2	1

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

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
32	a	<p>Vogliamo eseguire i test case in base alle loro priorità, ma dobbiamo anche considerare le dipendenze.                      Se consideriamo solo le priorità, vogliamo eseguire prima TC5 e TC7 (priorità più alta), poi TC1, TC3, TC4, e infine TC2 e TC6 (priorità più bassa)                      Tuttavia, per poter eseguire TC7, è necessario eseguire prima TC4.                      Per poter eseguire TC5, è necessario eseguire TC4 e TC2, ma TC2 è bloccato da TC1, che dovrebbe essere eseguito prima di TC2.                      Quindi, per eseguire i test case con priorità 1 il prima possibile, i primi cinque test case dovrebbero essere: TC4 - TC7 - TC1 - TC2 - TC5.                      Successivamente, sarà necessario eseguire TC3, perché ha una priorità più alta di TC6.                      La schedulazione completa sarà quindi: TC4 - TC7 - TC1 - TC2 - TC5 - TC3 - TC6.                      Quindi, il sesto test case sarà TC3.</p> <p>Quindi:                      a) È corretta                      b) Non è corretta                      c) Non è corretta                      d) Non è corretta</p>	FL-5.1.5	K3	1
33	b	<p>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 test differenti 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 test differenti</p>	FL-5.1.6	K1	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
34	d	<p>a) Non è corretta. I quadranti del testing raggruppano separatamente i livelli di test e i tipi di test in base a diversi criteri. Non rappresentano alcuna combinazione di livelli di test e tipi di test, e non sono correlati ad alcuna posizione all'interno del ciclo di vita dello sviluppo software. Sia i livelli di test sia i tipi di test sono trattati separatamente nel modello dei quadranti del testing</p> <p>b) Non è corretta. I quadranti del testing raggruppano i livelli di test e i tipi di test in base a diversi criteri. Non descrivono il grado di granularità dei singoli tipi di test eseguiti a ogni livello di test. Un modello di questo tipo, relativo ai livelli di test, è chiamato piramide di test</p> <p>c) Non è corretta. L'affermazione è errata, perché in generale qualsiasi tipo di test può essere eseguito a qualsiasi livello di test</p> <p>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</p>	FL-5.1.7	K2	1
35	c	<p>a) Non è corretta. Il monitoraggio del rischio è parte del controllo del rischio, non dell'analisi del rischio</p> <p>b) Non è corretta. L'identificazione del rischio non permette di implementare le attività di mitigazione del rischio. Le azioni di mitigazione vengono definite durante la fase di controllo del rischio</p> <p>c) È corretta. Questo è un esempio di come l'analisi del rischio influenzi la completezza e l'ambito del testing</p> <p>d) Non è corretta. Gli elementi di copertura vengono derivati utilizzando tecniche di test, non attraverso l'analisi del rischio</p>	FL-5.2.3	K2	1



Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
36	b	<p>a) Non è corretta. I test progress report vengono utilizzati principalmente durante il monitoraggio e controllo dei test, non durante la progettazione dei test</p> <p>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</p> <p>c) Non è corretta. I test progress report sono utilizzati principalmente durante il monitoraggio e controllo dei test, non durante l'analisi dei test</p> <p>d) Non è corretta. I test progress report sono utilizzati principalmente durante il monitoraggio e il controllo dei test, non durante la pianificazione dei test</p>	FL-5.3.2	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
37	d	<p>a) Non è corretta. Quando un utente segnala un failure del software, grazie all'identificazione univoca degli elementi di documentazione approvati, 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 rilevare il difetto più velocemente</p> <p>b) Non è corretta. Se una modifica all'ambiente di test causa problemi imprevisti durante il testing, il configuration management consente ai tester di eseguire un rollback a una versione precedente dell'ambiente. Questo assicura che il testing possa continuare senza essere influenzato dalla modifica</p> <p>c) Non è corretta. Il configuration management assicura che tutta la documentazione identificata (p.e. le specifiche dei requisiti) e gli elementi del software siano referenziati senza ambiguità nella documentazione di test (p.e.i test plan)</p> <p>d) È corretta. Questo è garantito dal defect management, non dal configuration management</p>	FL-5.4.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
38	b	<p>a) Non è corretta. Questo è importante, ma non è importante come gli elementi dell'ambiente di test</p> <p>b) È corretta. L'elemento importante mancante 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</p> <p>c) Non è corretta. L'oggetto di test è stato identificato (WebShop v0.99)</p> <p>d) Non è corretta. L'impatto è definito come severità (alta)</p>	FL-5.5.1	K3	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
39	d	<p>a) Non è corretta. Gli strumenti di esecuzione dei test e copertura facilitano l'esecuzione automatizzata dei test case e la misura della copertura ottenuta con l'esecuzione di quei test case. Tuttavia, questi strumenti non supportano l'organizzazione dei difetti e il configuration management</p> <p>b) Non è corretta. Gli strumenti per la progettazione e l'implementazione dei test facilitano la generazione dei test case, dei dati di test e delle procedure di test, ma non supportano l'organizzazione dei difetti, la tracciabilità con la base di test e il configuration management</p> <p>c) Non è corretta. Gli strumenti di defect management vengono utilizzati per gestire i difetti, ma non sono strumenti di test e non vengono utilizzati per organizzare i test case o il configuration management</p> <p>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, dei requisiti, dei test, della tracciabilità con la base di test, dei difetti e del configuration management</p>	FL-6.1.1	K2	1

Numero domanda (#)	Risposta corretta	Spiegazione / Motivazione	Obiettivo di apprendimento (LO)	Livello K	Numero di punti
40	d	<p>a) Non è corretta. "Capacità di generare test case senza accedere alla base di test" non è possibile. La generazione dei test case da parte dei tester o degli strumenti di test richiede l'accesso alla base di test</p> <p>b) Non è corretta. "Ottenere una maggiore copertura attraverso una valutazione più obiettiva" non è un beneficio diretto della test automation. La test automation fornirà una valutazione più obiettiva della copertura, ma tale valutazione non aumenterà la copertura. E' possibile aumentare la copertura utilizzando solo i risultati della copertura per scrivere ulteriori test case</p> <p>c) Non è corretta. "Aumento dei tempi di esecuzione dei test con una maggiore potenza di elaborazione" è un'affermazione contraddittoria, poiché una maggiore potenza di elaborazione riduce normalmente i tempi di esecuzione, e l'aumento dei tempi di esecuzione non è un vantaggio poiché il testing richiederebbe più tempo</p> <p>d) È corretta. "Prevenzione degli errori umani attraverso una maggiore consistenza e ripetibilità" è un vantaggio della test automation, poiché la test automation non può essere affetta da errori umani. Ad esempio, significa che i test sono derivati in modo consistente 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</p>	FL-6.2.1	K1	1