



SIMULAZIONE D'ESAME ITA-1 SOLUZIONI

ISTQB® Specialist Test Automation Engineer
Versione 2016



VERSIONE 1.0

**THIS DOCUMENT CONTAINS PROPRIETARY AND CONFIDENTIAL MATERIAL OF ASSOCIAZIONE ITA-STQB.
ANY UNAUTHORIZED REPRODUCTION, USE, OR DISCLOSURE OF THIS MATERIAL, OR ANY PART THEREOF, IS
STRICTLY PROHIBITED WITHOUT EXPLICIT AUTHORIZATION OF ASSOCIAZIONE ITA-STQB
(EXAMS-SUPPORT@ITA-STQB.ORG)**

Avviso di copyright

Copyright Notice © Italian Software Testing Qualifications Board (di seguito chiamato ITA-STQB).

Con la presente gli autori trasferiscono il copyright all'ITA-STQB. Gli autori (come attuali detentori del copyright) e ITA-STQB (come 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 fornitore di formazione accreditato può usare questa simulazione d'esame nel suo corso di formazione se l'ITA-STQB è riconosciuta come proprietaria 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, se l'ITA-STQB è riconosciuta come proprietaria del copyright della simulazione d'esame.

Qualsiasi altro uso di questa simulazione d'esame è proibito senza aver prima ottenuto l'approvazione scritta di ITA-STQB.

Scopo di questo documento

Le domande e le risposte e le relative spiegazioni in questo set di simulazione d'esame sono state create da un team di esperti in materia e autori di domande d'esame con l'obiettivo di supportare i candidati all'esame di certificazione in lingua italiana.

Queste domande non possono essere usate così come sono in nessun esame ufficiale.

Istruzioni

In questo documento potete trovare:

- Domande¹, con relative opzioni di risposta e indicazione del valore in punti, e l'indicazione delle risposte corrette, compresa la motivazione.

¹ In questa simulazione d'esame le domande sono ordinate in base ai LO elencati nel syllabus; non ci si deve aspettare che le domande siano presentate in questo ordine in un questionario d'esame ufficiale.

Sommario

1. Capitolo 1	4
Domanda #1 – ITA (Punti: 1).....	4
Domanda #2 – ITA (Punti: 1).....	5
2 Capitolo 2	6
Domanda #3 – ITA (Punti: 3).....	6
Domanda #4 – ITA (Punti: 3).....	7
Domanda #5 – ITA (Punti: 3).....	8
Domanda #6 – ITA (Punti: 1).....	9
Domanda #7 – ITA (Punti: 1).....	10
3 Capitolo 3	11
Domanda #8 – ITA (Punti: 1).....	11
Domanda #9 – ITA (Punti: 3).....	12
Domanda #10 – ITA (Punti: 1).....	13
Domanda #11 – ITA (Punti: 1).....	14
Domanda #12 – ITA (Punti: 1).....	15
Domanda #13 – ITA (Punti: 3).....	16
Domanda #14 – ITA (Punti: 3).....	18
Domanda #15 – ITA (Punti: 2).....	19
Domanda #16 – ITA (Punti: 2).....	20
Domanda #17 – ITA (Punti: 1).....	21
4 Capitolo 4	22
Domanda #18 – ITA (Punti: 2).....	22
Domanda #19 – ITA (Punti: 2).....	23
Domanda #20 – ITA (Punti: 3).....	24
Domanda #21 – ITA (Punti: 3).....	26
Domanda #22 – ITA (Punti: 1).....	27
5 Capitolo 5	28
Domanda #23 – ITA (Punti: 1).....	28
Domanda #24 – ITA (Punti: 1).....	29
Domanda #25 – ITA (Punti: 2).....	30
Domanda #26 – ITA (Punti: 2).....	31
Domanda #27 – ITA (Punti: 3).....	32
Domanda #28 – ITA (Punti: 2).....	34
6 Capitolo 6	35
Domanda #29 – ITA (Punti: 2).....	35
Domanda #30 – ITA (Punti: 1).....	36
Domanda #31 – ITA (Punti: 1).....	37
Domanda #32 – ITA (Punti: 1).....	38
Domanda #33 – ITA (Punti: 1).....	39
7 Capitolo 7	40
Domanda #34 – ITA (Punti: 2).....	40
Domanda #35 – ITA (Punti: 2).....	41
Domanda #36 – ITA (Punti: 2).....	42
Domanda #37 – ITA (Punti: 2).....	43
8 Capitolo 8	45
Domanda #38 – ITA (Punti: 3).....	45
Domanda #39 – ITA (Punti: 3).....	46
Domanda #40 – ITA (Punti: 3).....	47

1. Capitolo 1

Domanda #1 – ITA (Punti: 1)

Sezione

1.1 Scopo dell'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-1.1.1 (K2) Spiegare gli obiettivi, i vantaggi, gli svantaggi e le limitazioni dell'automazione del testing

Domanda

L'automazione del testing:

- A. mira a ridurre il cosiddetto fenomeno di “automation bias”, in cui le persone tendono ad accettare con eccessiva facilità le raccomandazioni fornite da sistemi di supporto alle decisioni basate su intelligenza artificiale
- B. richiede spesso un rilevante investimento iniziale nel predisporre una TAS, ma non richiede alcun intervento manuale di manutenzione in quanto questa viene svolta automaticamente dalla TAS stessa
- C. svolgendo automaticamente una serie di attività ripetitive legate all'esecuzione manuale di test, può consentire ai tester di dedicarsi ad attività più impegnative e soddisfacenti
- D. consente di eseguire in modo automatizzato non solo i test funzionali ma anche tutti i test non funzionali associati alle altre caratteristiche di qualità di un prodotto software definite nello standard ISO 25010

Risposta corretta

C

Soluzione

A è sbagliata perché si riferisce ad un bias caratteristico dell'intelligenza artificiale nell'ambito dei sistemi di supporto alle decisioni (decision making) e l'automazione del testing non si pone alcun obiettivo in riferimento alla riduzione di tale bias (si veda [d] §7.4).

B è sbagliata. Nonostante la prima parte (“[...] richiede spesso un rilevante investimento iniziale nel predisporre una TAS”) sia vera, la seconda parte (“[...] ma non richiede alcun intervento manuale di manutenzione in quanto questa viene svolta automaticamente dalla TAS stessa”) è falsa (si veda [a] §1.1).

C è corretta. Infatti, [a] §1.1, nel descrivere i vantaggi dell'automazione dei testing, specifica anche un uso più efficace ed efficiente delle risorse di testing come quello descritto.

D è sbagliata. Infatti, ad esempio, molti test non funzionali associati all'usabilità (lo stesso vale per alcune altre caratteristiche) non sono eseguibili in modo automatizzato.

Perciò, C è la sola risposta corretta.

Domanda #2 – ITA (Punti: 1)

Sezione

1.2 Fattori di Successo nell'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-1.2.1 (K2) Identificare i fattori tecnici di successo di un progetto di automazione del testing

Domanda

Quale delle seguenti risposte si riferisce ad un fattore di successo per un progetto di automazione del testing?

- A. L'automazione di test case manuali eseguiti frequentemente può consentire di allocare i tester su attività più impegnative e gratificanti
- B. La TAA dovrebbe essere progettata tenendo in considerazione anche eventuali requisiti non funzionali come quelli relativi alla manutenibilità
- C. La strategia di automazione del testing dovrebbe prevedere di scrivere test automatizzati per un solo tipo di interfaccia del SUT (e.g., solo GUI o solo API)
- D. L'approccio all'implementazione automatizzata dei test dovrebbe essere il più flessibile possibile ed essere scelto tra keyword-driven e process-driven

Risposta corretta

B

Soluzione

A è sbagliata. Infatti, pur essendo un'affermazione vera, essa si riferisce ad un vantaggio dell'automazione del testing, non ad un fattore di successo per un progetto di automazione del testing.

B è corretta. Infatti, [a] §1.2 dice: “La Test Automation Architecture (TAA) è strettamente collegata all'architettura del prodotto software. Deve essere chiaro quali requisiti funzionali e non funzionali l'architettura deve supportare. Tipicamente questi saranno i requisiti più importanti” e “Spesso la TAA è progettata per la manutenibilità, le prestazioni e la facilità di apprendimento (si veda ISO/IEC 25000:2014 per i dettagli di queste e altre caratteristiche non funzionali). È utile coinvolgere ingegneri del software che comprendano l'architettura del SUT”.

C è sbagliata. Infatti, al contrario, una strategia di automazione del testing deve tenere in considerazione le caratteristiche (e.g., fragilità di manutenzione, velocità di esecuzione, determinismo dei risultati, testabilità) dei test automatizzati per le varie tipologie di interfaccia fornite dal SUT e determinare quali test automatizzare per tali tipologie in base agli obiettivi del progetto e a vari aspetti legati al SUT (e.g., architettura del SUT, ciclo di vita usato per il SUT).

D è sbagliata. Infatti, l'approccio all'implementazione automatizzata dei casi di test più adatto per un progetto di automazione del testing dipende da vari fattori, tra cui la dimensione del SUT e la sua complessità. La scelta di un framework che implementa un approccio descrittivo e ultra-flessibile può non essere quella migliore (si veda [a] §2.1 e [a] §3.2.2).

Perciò, B è la sola risposta corretta.

2 Capitolo 2

Domanda #3 – ITA (Punti: 3)

Sezione

2.1 Aspetti del SUT che influenzano l'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-2.1.1 (K4) Analizzare un sistema sotto test per determinare la soluzione di automazione appropriata

Domanda

Un servizio web chiamato WEFOR, sviluppato all'interno della vostra organizzazione, svolge elaborazioni di dati meteorologici producendo previsioni per i 7 giorni successivi a quello attuale per una serie di comuni italiani e rende disponibili tali previsioni tramite API che vengono chiamate da applicazioni web esterne (che a loro volta le utilizzano per offrire servizi ai propri utenti). WEFOR è accessibile esclusivamente a livello di tali API e i dati che elabora li ottiene chiamando servizi web esterni forniti da terze parti. In qualità di TAE, il vostro obiettivo è predisporre una TAS che consenta di eseguire test automatizzati mirati esclusivamente a verificare che le elaborazioni di WEFOR siano svolte correttamente in corrispondenza di un'ampia gamma di dati meteorologici messi a disposizione per i vari comuni. Tali test dovranno essere il più deterministici possibile. Sulla base delle sole informazioni fornite, quale tra le seguenti azioni repute essere la più rilevante per raggiungere il vostro obiettivo?

- A. Predisporre l'aggiunta di codice (e delle relative API che ne consentano l'accesso) in WEFOR che consenta di supportare la testabilità necessaria a implementare i test automatizzati a livello di API su di esso
- B. Predisporre un framework keyword-driven per implementare test automatizzati di tipo end-to-end a livello di GUI sulle applicazioni web che chiamano le API di WEFOR
- C. Predisporre opportuni mock per i servizi web di terze parti chiamati da WEFOR che sostituiscano tali servizi durante l'esecuzione di test automatizzati implementati a livello di API su WEFOR
- D. Predisporre un framework process-driven per implementare test automatizzati di tipo end-to-end a livello di GUI sulle applicazioni web che chiamano le API di WEFOR

Risposta corretta

C

Soluzione

C è corretta. In tal caso, il SUT sarebbe rappresentato da WEFOR e la TAS si interfaccerebbe ad esso a livello di API. Tuttavia, WEFOR ha delle dipendenze esterne. Dato che i test dovranno essere il più deterministici possibile e dovranno essere svolti in corrispondenza di un'ampia gamma di dati meteorologici per i vari comuni, sarà necessario eliminare queste dipendenze esterne sostituendole con dei mock. In questo modo la TAS avrà controllo completo sull'esecuzione garantendo determinismo e i mock potranno essere predisposti opportunamente per restituire i dati meteorologici di interesse per lo svolgimento dei test automatizzati.

A è sbagliata. Infatti, sulla base delle sole informazioni fornite, non è possibile capire se le API attualmente messe a disposizione di WEFOR supportino o meno la testabilità richiesta rispetto alle esigenze dei test automatizzati da implementare.

B e D sono sbagliate. Test automatizzati end-to-end a livello delle GUI delle applicazioni web sarebbero meno deterministici e più fragili di quelli indicati per la risposta C. Inoltre, essi non

riuscirebbero a concentrarsi esclusivamente sul testare le elaborazioni di WEFOR (quantomeno dovrebbero considerare anche la parte di presentazione delle applicazioni web. Perciò, C è la sola risposta corretta.

Domanda #4 – ITA (Punti: 3)

Sezione

2.1 Aspetti del SUT che influenzano l'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-2.1.1 (K4) Analizzare un sistema sotto test per determinare la soluzione di automazione appropriata

Domanda

In qualità di TAE, state valutando un'architettura a layer ("layered architecture") dal punto di vista della testabilità, al fine di predisporre un'opportuna TAS da utilizzare per svolgere test automatizzati su alcune applicazioni che sono basate su tale architettura. L'architettura consiste di layer distinti impilati verticalmente uno in cima all'altro, in cui i componenti in un layer possono interagire solo con componenti nello stesso layer o con componenti di qualsiasi layer inferiore. Sulla base delle sole informazioni fornite, quale delle seguenti affermazioni non è vera?

- A. Tale architettura può aumentare la testabilità fornendo interfacce ben definite su tutti i layer
- B. Tale architettura può aiutare l'isolamento di singoli componenti per scopi di testing
- C. Tale architettura può semplificare la sostituzione dei componenti dei layer inferiori con componenti per il testing
- D. Tale architettura conduce automaticamente ad applicazioni altamente testabili

Risposta corretta

D

Soluzione

Le architetture a layer supportano la testabilità in diversi modi. Un'architettura a layer appropriata può aiutare a supportare una forte 'separation of concerns' che, a sua volta, può aiutare a supportare il raggiungimento di controllabilità e osservabilità desiderate. Ciò non significa che un'architettura a layer di per sé assicuri automaticamente un elevato grado di controllabilità e osservabilità (ad esempio, la creazione di applicazioni 'loosely coupled' richiede più della semplice adozione di architetture a layer). Inoltre, [a] §2.1.3 dice: "Architettura chiaramente definita: la terza parte importante della testabilità è un'architettura che fornisca interfacce chiare e comprensibili, che diano controllo e visibilità a tutti i livelli di test". Quindi, D è corretta perché è un'affermazione falsa, mentre A è sbagliata perché è un'affermazione vera.

In un'architettura a layer le dipendenze tra i componenti sono vincolate in modo tale che i componenti nei layer superiori possano chiamare solo componenti nei layer inferiori. Ciò aiuta a isolare i singoli componenti per i test e facilita la sostituzione di componenti di livello inferiore con componenti personalizzati a scopo di test. Quindi B e C sono sbagliate perché sono entrambe affermazioni vere. Perciò, D è l'unica risposta corretta.

Domanda #5 – ITA (Punti: 3)

Sezione

2.2 Valutazione e Selezione degli Strumenti

Obiettivo di Apprendimento

ALTA-E-2.2.1 (K4) Analizzare gli strumenti per l'automazione del testing per un dato progetto ed effettuare un resoconto sugli aspetti tecnici e sulle scelte

Domanda

In qualità di TAE, nell'ambito di attività di testing di compatibilità cross-browser, state attualmente valutando uno strumento di automazione del testing per condurre test funzionali automatizzati a livello di GUI su applicazioni web. L'esigenza principale che lo strumento deve supportare è quella di consentire di condurre tali test su vari browser per un'ampia gamma di dispositivi (smartphone, tablet, desktop) nel modo più efficiente possibile. Le applicazioni web da testare sono progettate con approccio 'responsive' e dovrebbero essere quindi in grado di adattarsi in modo automatico al dispositivo sulla base delle sue caratteristiche tecniche, scalando adeguatamente i componenti (e i contenuti associati) delle pagine web. Sulla base delle sole informazioni fornite, quale tra le seguenti funzionalità dello strumento ritenete essere la più importante ai fini della vostra valutazione?

- A. Lo strumento consente di generare report di esecuzione dei test automatizzati che contengono collegamenti ipertestuali ai log di esecuzione
- B. Lo strumento consente di impostare le azioni da eseguire a fronte di errori fatali delle applicazioni in modo da ripristinarne il funzionamento corretto
- C. Lo strumento consente di eseguire i test automatizzati in parallelo su varie macchine fisiche o virtuali
- D. Lo strumento offre una funzionalità di cattura/riesecuzione (capture/playback) a livello del protocollo HTTP

Risposta corretta

C

Soluzione

L'approccio 'responsive' citato, che consente alle applicazioni web di adattarsi automaticamente ai requisiti tecnici del rispettivo dispositivo su cui queste vengono fruite, si è ormai affermato come standard indiscusso. L'esigenza di eseguire test funzionali automatizzati relativi a tali applicazioni su vari browser per un'ampia gamma di dispositivi (smartphone, tablet, desktop) nel modo più efficiente possibile è fondamentale nell'ambito di attività di testing di compatibilità cross-browser. Tale esigenza può essere indirizzata dalla possibilità di eseguire tali test in parallelo su varie macchine fisiche o virtuali (questo è quanto, per esempio, fa il componente 'Selenium Grid' del noto framework di web testing Selenium). La simultaneità di esecuzione consentirebbe infatti di ridurre notevolmente i tempi di esecuzione dei test di compatibilità cross-browser. Quindi C è corretta.

A, B e D si riferiscono a funzionalità dello strumento 'nice to have' o comunque potenzialmente interessanti in generale. Sulla base delle sole informazioni fornite, tuttavia non sono così rilevanti come la funzionalità descritta nella risposta C per indirizzare l'esigenza espressa nello scenario.

Perciò, C è la sola risposta corretta.

Domanda #6 – ITA (Punti: 1)

Sezione

2.3 Progettare per la Testabilità e l'Automazione

Obiettivo di Apprendimento

ALTA-E-2.3.1 (K2) Comprendere i metodi per "progettare per la testabilità" e "progettare per la l'automazione del testing" applicabili al SUT

Domanda

La testabilità di un SUT include la sua:

- A. Portabilità
- B. Manutenibilità
- C. Adattabilità
- D. Controllabilità

Risposta corretta

D

Soluzione

Come descritto in [a] §2.3, la testabilità di un SUT include la sua controllabilità e la sua osservabilità. Quindi, D è corretta, mentre A, B e C sono sbagliate. Perciò, D è la sola risposta corretta.

Domanda #7 – ITA (Punti: 1)

Sezione

2.3 Progettare per la Testabilità e l'Automazione

Obiettivo di Apprendimento

ALTA-E-2.3.1 (K2) Comprendere i metodi per "progettare per la testabilità" e "progettare per la automazione del testing" applicabili al SUT

Domanda

La disponibilità di interfacce che forniscono informazioni sul SUT che possono essere utilizzate da test automatizzati, eseguiti da una TAS, per verificare la corrispondenza tra il risultato atteso e quello effettivo, fa riferimento a:

- A. Riutilizzabilità della TAS
- B. Osservabilità del SUT
- C. Modularità del SUT
- D. Controllabilità del SUT

Risposta corretta

B

Soluzione

B è corretta, mentre A, C e D sono sbagliate. Infatti, [a] §2.3, nel descrivere l'osservabilità del SUT come sotto-caratteristica della sua testabilità, specifica quanto segue "Osservabilità: è necessario che il SUT fornisca interfacce che diano visibilità all'interno del sistema. I casi di test possono allora usare queste interfacce per controllare, per esempio, se il comportamento atteso corrisponde a quello effettivo."

Perciò, B è la sola risposta corretta.

3 Capitolo 3

Domanda #8 – ITA (Punti: 1)

Sezione

3.1 Introduzione alla gTAA

Obiettivo di Apprendimento

ALTA-E-3.1.1 (K2) Spiegare la struttura della gTAA

Domanda

Quale delle seguenti risposte contiene esclusivamente componenti che appartengono al layer di esecuzione del testing (test execution layer) della gTAA?

- A. Test data, Test library, Test execution
- B. Test logging, Test reporting, Test execution
- C. Test models, Test cases, Test reporting
- D. Manual design, Test procedures, Test execution

Risposta corretta

B

Soluzione

Tra quelli citati nelle risposte, i componenti:

- “Test data”, “Test library”, “Test cases” e “Test procedures” appartengono al layer di definizione del testing (test definition layer),
- “Test models” e “Manual design” appartengono al layer di generazione del testing (test generation layer).
- “Test logging”, “Test reporting” e “Test execution” appartengono al layer di esecuzione del testing (test execution layer).

Quindi, B è corretta, mentre A, C e D sono sbagliate.

Perciò, B è la solta risposta corretta.

Domanda #9 – ITA (Punti: 3)

Sezione

3.2 Progettazione di una TAA

Obiettivo di Apprendimento

ALTA-E-3.2.1 (K4) Progettare la TAA appropriata per un dato progetto

Domanda

In qualità di TAE, state attualmente progettando il layer di adattamento del testing per una TAA (basata sulla gTAA) sulla base della quale realizzare una TAS che verrà usata per effettuare test automatizzati a livello di UI per mobile app native sviluppate per Android. I requisiti di tali applicazioni vengono definiti sotto forma di use case. Per tale TAA, avete già preso tutte le decisioni relative alla progettazione del layer di generazione del testing, del layer di definizione del testing e del layer di esecuzione del testing. Sulla base delle sole informazioni fornite, quale tra le seguenti potrebbe essere una delle vostre decisioni future di progettazione compatibile con lo scenario fornito?

- A. La scelta del driver da utilizzare per interagire con la UI delle mobile app native indicate
- B. La scelta di UTP (UML Testing Profile) come notazione da utilizzare per la specifica dei test indicati
- C. La scelta di Ruby come linguaggio da utilizzare per implementare le procedure di test per i test indicati
- D. La scelta delle linee guida da utilizzare e del formato per definire adeguatamente i test indicati

Risposta corretta

A

Soluzione

A è corretta perché si riferisce ad una scelta relativa al layer di adattamento del testing (si veda “Confrontare e valutare differenti approcci di progettazione/architettura” in [a] §3.2.1) compatibile con lo scenario fornito (e.g., esistono vari driver, come ad esempio UIAutomation2 fornito da Google).

B è sbagliata perché si riferisce ad una scelta relativa al layer di definizione del testing (si veda “Confrontare e valutare differenti approcci di progettazione/architettura” in [a] §3.2.1), che quindi dovrebbe essere quindi già stata presa.

C è sbagliata perché si riferisce ad una scelta relativa al layer di esecuzione del testing (si veda “Confrontare e valutare differenti approcci di progettazione/architettura” in [a] §3.2.1), che dovrebbe essere quindi già stata presa.

D è sbagliata perché si riferisce ad una scelta relativa al layer di definizione del testing (si veda “Confrontare e valutare differenti approcci di progettazione/architettura” in [a] §3.2.1), che dovrebbe essere quindi già stata presa.

Perciò, A è la sola risposta corretta.

Domanda #10 – ITA (Punti: 1)

Sezione

3.2 Progettazione di una TAA

Obiettivo di Apprendimento

ALTA-E-3.2.2 (K2) Spiegare il ruolo che i layer svolgono all'interno di una TAA

Domanda

Una TAS basata sulla gTAA deve produrre dei file in formato HTML, sia per i report che per i log, in modo da soddisfare la seguente esigenza: i report devono contenere al loro interno dei collegamenti ipertestuali ai log per consentire una rapida navigazione tra le varie informazioni in essi contenute. Quale dei seguenti layer di una gTAA viene direttamente coinvolto da questa esigenza?

- A. Layer di esecuzione del testing
- B. Layer di reporting del testing
- C. Layer di generazione del testing
- D. Layer di definizione del testing

Risposta corretta

A

Soluzione

L'esigenza specificata coinvolge i componenti di reporting ("test reporting") e di logging ("test logging") del layer di esecuzione del testing ("test execution layer"). Quindi, A è corretta, mentre C e D sono sbagliate.

B è sbagliata perché il layer di reporting del testing non è un layer della gTAA (quello che è previsto nella gTAA è il componente di reporting all'interno del layer di esecuzione del testing).

Perciò, A è la sola risposta corretta.

Domanda #11 – ITA (Punti: 1)

Sezione

3.2 Progettazione di una TAA

Obiettivo di Apprendimento

ALTA-E-3.2.3 (K2) Comprendere le valutazioni per la progettazione di una TAA

Domanda

Una maggiore astrazione in una TAA è più probabile che:

- A. aumenti la testabilità del SUT sia in termini di osservabilità che di controllabilità
- B. riduca la portabilità dei test automatizzati e di eventuali altri artefatti prodotti da TAS basate su tale TAA
- C. renda più accessibile agli stakeholder con profili meno tecnici l'uso di TAS basate su tale TAA
- D. aumenti le prestazioni da parte di TAS basate su tale TAA

Risposta corretta

C

Soluzione

A è sbagliata. Infatti, la testabilità del SUT (rappresentata sia dalla sua osservabilità che controllabilità) è una caratteristica del SUT e non è legata all'astrazione della TAA.

B è sbagliata. Infatti, come descritto in [a] §3.2.1 (si veda "Identificare le aree dove l'astrazione può portare benefici"), una maggiore astrazione in una TAA è più che probabile che aumenti la portabilità dei test automatizzati e di eventuali altri artefatti prodotti da TAS basate su tale TAA.

C è corretta. Infatti, [a] §3.2.1 (si veda "Identificare le aree dove l'astrazione può portare benefici") dice: "Inoltre, l'astrazione aiuta a rendere una TAA (e le sue implementazioni in TAS) più accessibili ai non tecnici, poiché le suite di test possono essere documentate (anche in modo grafico) e spiegate a più alto livello, il che migliora leggibilità e comprensione".

D è sbagliata. Infatti, come descritto in [a] §3.2.1 (si veda "Identificare le aree dove l'astrazione può portare benefici"), una maggiore astrazione in una TAA è più probabile che riduca le prestazioni da parte di TAS basate su tale TAA.

Perciò, C è la sola risposta corretta.

Domanda #12 – ITA (Punti: 1)

Sezione

3.2 Progettazione di una TAA

Obiettivo di Apprendimento

ALTA-E-3.2.3 (K2) Comprendere le valutazioni per la progettazione di una TAA

Domanda

Per quale dei seguenti layer della gTAA viene richiesto di scegliere la tecnologia più adatta ad implementare, tramite script di test automatizzati, le procedure di test relative a un dato insieme di test manuali?

- A. Layer di generazione del testing
- B. Layer di definizione del testing
- C. Layer di esecuzione del testing
- D. Layer di adattamento del testing

Risposta corretta

C

Soluzione

C è corretta, mentre A, B e D sono sbagliate. Infatti, [a] §3.2.1, nel descrivere le considerazioni relative alle scelte da intraprendere nell'ambito del layer di esecuzione del testing della gTAA, dice: "Scelta della tecnologia di implementazione per implementare le procedure di test (imperativa, come C; funzionale come Haskell o Erlang; object-oriented, come C++, C#, Java; scripting, come Python o Ruby, o tecnologia specifica dello strumento)". Viene inoltre specificato come questa scelta sia tipicamente dipendente dalla scelta dello strumento di esecuzione dei test.

Perciò, C è la sola risposta corretta.

Domanda #13 – ITA (Punti: 3)

Sezione

3.2 Progettazione di una TAA

Obiettivo di Apprendimento

ALTA-E-3.2.4 (K4) Analizzare i fattori di implementazione, uso e i requisiti di manutenzione per una data TAS

Domanda

Considerate il seguente approccio per implementare test di accettazione automatizzati relativi alle user story di un progetto Agile mirato a sviluppare un sistema software. Tali test vengono scritti all'interno di opportuni 'feature file' come sequenze che consistono di tre passi: GIVEN-WHEN-THEN. In particolare:

- Il passo GIVEN consente di specificare lo stato in cui si deve trovare inizialmente il sistema
- Il passo WHEN consente di specificare un'azione svolta sul sistema
- Il passo THEN consente di specificare un risultato atteso fornito dal sistema

Tali passi vengono specificati dal punto di vista del comportamento del sistema usando il linguaggio naturale e poi vengono resi eseguibili associandoli, tramite delle espressioni regolari, al corrispondente codice dell'automazione (scritto in un dato linguaggio di programmazione) che li implementa interagendo col sistema. In questo modo, i test nei 'feature file' possono essere scritti da esperti di dominio (ad esempio, dei Test Analyst) senza competenze di programmazione, in modo separato dal codice dell'automazione che invece viene scritto da figure dotate di tali competenze tecniche. Sulla base delle sole informazioni fornite, questo può essere visto come un esempio di un approccio all'automazione dei test di tipo:

- A. descrittivo simile a quello keyword-driven
- B. prescrittivo simile a quello di scripting strutturato
- C. descrittivo simile a quello data-driven
- D. prescrittivo simile a quello di scripting lineare

Risposta corretta

A

Soluzione

L'approccio presentato è quello che viene spesso utilizzato nel contesto della pratica di sviluppo Agile nota come BDD (Behavior-Driven Development), seppur non debba essere considerato necessariamente limitato soltanto ad essa. Infatti, tale pratica usa i test di accettazione specificati (che vengono indicati come 'scenari' o 'specifiche', ma di fatto rappresentano test di accettazione 'non convenzionali' che semplicemente seguono un pattern GIVEN-WHEN-THEN, rispetto a quelli 'convenzionali' che seguono un pattern ARRANGE-ACT-ASSERT) per guidare lo sviluppo del codice (approccio test-first) e rappresenta infatti un'estensione della pratica di sviluppo Agile nota come TDD (Test-Driven Development) che si concentra, di fatto, solo sui test di unità ed è una pratica che coinvolge solo gli sviluppatori (mentre BDD coinvolge l'intero team). L'approccio presentato può essere comunque utilizzato anche al di fuori di BDD, ovvero come parte di un tradizionale approccio test-last all'implementazione di automatizzati, adottato dopo che sia stato sviluppato il codice del SUT. Si tratta di un approccio descrittivo simile a quello keyword-driven. Nell'approccio keyword-driven esiste una parte "business-facing" che consente di descrivere i test usando un metalinguaggio (business readable) non eseguibile sul SUT che è quello delle keyword e una parte "technology-

facing” che usa un linguaggio eseguibile sul SUT: in questo caso, il mapping tra queste due parti avviene attraverso i cosiddetti ‘keyword handler’. Nello scenario descritto, esiste ancora una parte “business-facing” che consente di descrivere i test usando come metalinguaggio (business readable) quello naturale per specificare i passi e una parte “technology-facing” che usa un linguaggio eseguibile sul SUT. Il mapping tra queste due parti come descritto nel seguente estratto dello scenario: “Tali passi vengono specificati dal punto di vista del comportamento del sistema usando il linguaggio naturale e poi vengono resi eseguibili associandoli, tramite delle espressioni regolari, al corrispondente codice dell’automazione (scritto in un dato linguaggio di programmazione) che li implementa interagendo col sistema”. Questo è quello che avviene in tool come “Cucumber” che mappano i passi (“step”) nelle cosiddette “step definitions” del codice dell’automazione come indicato. Nel caso di Cucumber, il metalinguaggio si basa su Gherkin.

Quindi, A è corretta, mentre B e D sono sbagliate. C è sbagliata perché l’approccio data-driven è prescrittivo, non descrittivo.

Perciò, A è la sola risposta corretta.

Domanda #14 – ITA (Punti: 3)

Sezione

3.2 Progettazione di una TAA

Obiettivo di Apprendimento

ALTA-E-3.2.4 (K4) Analizzare i fattori di implementazione, uso e i requisiti di manutenzione per una data TAS

Domanda

In qualità di TAE, state valutando l'approccio di scripting più adatto da utilizzare per l'automazione di test funzionali a livello di GUI su applicazioni desktop. Uno degli obiettivi più importanti è ridurre il più possibile sia lo sforzo richiesto per la manutenzione dei test automatizzati che lo sforzo richiesto per l'aggiunta di nuovi test automatizzati. La TAS comprende uno strumento che offre una funzionalità di cattura/riesecuzione (capture/playback), in grado di riconoscere tutti gli oggetti delle GUI coinvolte. Lo strumento usa un linguaggio di scripting che consente di costruire librerie di test contenenti script di supporto richiamabili dagli script di test che implementano i test automatizzati. I test saranno automatizzati da tester con significative competenze di programmazione. È già stata esclusa la possibilità di predisporre un framework keyword-driven, in quanto considerato troppo costoso rispetto alle caratteristiche (dimensione, stabilità, etc.) delle applicazioni sotto test e dei test da automatizzare. Sulla base delle sole informazioni fornite, quale dei seguenti approcci di scripting reputate essere il più adatto in questo scenario?

- A. Scripting lineare
- B. Scripting strutturato
- C. Scripting event-driven
- D. Scripting process-driven

Risposta corretta

B

Soluzione

Lo scenario specifica quanto segue: “È già stata esclusa la possibilità di predisporre un framework keyword-driven in quanto considerato troppo costoso rispetto alle caratteristiche (dimensione, stabilità, etc.) delle applicazioni sotto test e dei test da automatizzare”. Un approccio di scripting process-driven è di fatto costruito ‘on top’ sull'approccio di testing keyword-driven, come descritto in [a] §3.2.2 (si veda “Scripting Process-driven”). Se è troppo costoso predisporre un framework keyword-driven a maggiore ragione lo è predisporre un framework process-driven. Quindi, D è sbagliata.

Lo scenario specifica anche quanto segue: “Uno degli obiettivi più importanti è ridurre il più possibile sia lo sforzo richiesto per la manutenzione dei test automatizzati che lo sforzo richiesto per l'aggiunta di nuovi test automatizzati”. L'approccio di scripting strutturato consente di raggiungere tale obiettivo molto meglio rispetto all'approccio di scripting lineare. L'opportunità di sfruttare questo approccio è supportata anche da quanto di seguito specificato: “Lo strumento usa un linguaggio di scripting che consente di costruire librerie di test contenenti script di supporto richiamabili dagli script di test che implementano i test automatizzati” (si veda “Scripting strutturato” in [a] §3.2.2). La scelta dell'approccio di scripting strutturato è anche ulteriormente rafforzata dal seguente estratto: “I test saranno automatizzati da tester con significative competenze di programmazione”. Quindi, B è corretta, mentre A è sbagliata.

C è sbagliata in quanto non esiste alcun approccio di scripting indicato come 'event-driven'.
Perciò, B è la sola risposta corretta.

Domanda #15 – ITA (Punti: 2)

Sezione

3.3 Sviluppo di una TAS

Obiettivo di Apprendimento

ALTA-E-3.3.1 (K3) Applicare le componenti di una generica TAA (gTAA) per costruire una TAA per un obiettivo specifico

Domanda

Selenium è un framework di web browser automation frequentemente usato per l'automazione di test su applicazioni web a livello di GUI. Tale framework è costituito da una suite di vari software che include Selenium WebDriver (SWD). SWD offre delle interfacce che consentono di effettuare chiamate a vari browser (Firefox, Chrome, Opera, etc.) usando il supporto nativo per l'automazione di ciascun browser. I test automatizzati per un'applicazione web a livello di GUI possono quindi utilizzare tali interfacce. Si consideri l'uso della gTAA per definire la TAA relativa ad una TAS (mirata ad eseguire test automatizzati per applicazioni web a vari livelli incluso quello di GUI) che includa SWD. Sulla base delle sole informazioni fornite, quale delle seguenti risposte descrive meglio come inquadrare SWD all'interno della gTAA?

- A. Componente del layer di generazione del testing
- B. Componente del layer di definizione del testing
- C. Componente del layer di esecuzione del testing
- D. Componente del layer di adattamento del testing

Risposta corretta

D

Soluzione

Il layer di adattamento del testing facilita la separazione del SUT (e delle sue interfacce) dai test automatizzati (e delle suite relative con cui questi test sono organizzati) da eseguire su tale SUT. In questo caso il SUT è rappresentato da un'applicazione web e l'interfacciamento con tale SUT avviene a livello di GUI tramite l'interazione col browser. SWD può essere quindi inquadrato come un componente del layer di adattamento del testing, dato che fornisce un modo programmatico per accedere al SUT tramite l'interfaccia del browser.

Quindi, D è corretta, mentre A, B e C sono sbagliate.

Perciò, D è la sola risposta corretta.

Domanda #16 – ITA (Punti: 2)

Sezione

3.3 Sviluppo di una TAS

Obiettivo di Apprendimento

ALTA-E-3.3.1 (K3) Applicare le componenti di una generica TAA (gTAA) per costruire una TAA per un obiettivo specifico

Domanda

Si consideri l'uso della gTAA per definire la TAA relativa ad una TAS. La TAS dovrà consentire di generare automaticamente dei test case a partire da descrizioni formali del comportamento del SUT rappresentate da macchine a stati finiti. Tale generazione, che sarà esclusivamente automatica, produrrà dei test case astratti che dovranno essere resi concreti per poi poter essere eseguiti sul SUT. La TAS dovrà occuparsi infatti non solo della generazione automatica dei test case, ma anche della loro successiva esecuzione sul SUT. Sulla base delle sole informazioni fornite, quale delle seguenti risposte descrive correttamente un componente della gTAA che andrebbe incluso in tale TAS?

- A. Il componente 'Manual Test' nel layer di generazione del testing
- B. Il componente 'Test Execution' nel layer di definizione del testing
- C. Il componente 'API' nel layer di adattamento del testing
- D. Il componente 'Test Models' nel layer di generazione del testing

Risposta corretta

D

Soluzione

D è corretta, mentre A è sbagliata. Infatti, nello scenario viene specificato che la generazione dei test case avverrà esclusivamente in modo automatico (ovvero non è da prevedere il componente 'Manual Test' che si occupa della generazione manuale dei test case) a partire da delle macchine a stati finiti che descrivono in modo formale il comportamento (ovvero dei modelli): è quindi certamente da prevedere il componente 'Test Models' che si occupa della generazione automatica di test case (astratti) a partire da dei modelli del SUT.

B è sbagliata. Infatti, il componente 'Test Execution' andrebbe incluso nella TAS, ma esso è nel layer di esecuzione del testing della gTAA, non nel layer di definizione del testing.

C è sbagliata. Nello scenario non vi è alcun riferimento alle interfacce (API, GUI, CLI, etc.) del SUT da utilizzare. Quindi non è possibile determinare, sulla base delle sole informazioni fornite, quali componenti del layer di adattamento del testing (ovvero gli adattatori) includere.

Perciò, D è la sola risposta corretta

Domanda #17 – ITA (Punti: 1)

Sezione

3.3 Sviluppo di una TAS

Obiettivo di Apprendimento

ALTA-E-3.3.2 (K2) Spiegare i fattori da considerare quando si identifica la riusabilità dei componenti

Domanda

Quale delle seguenti affermazioni non è vera?

- A. Per una TAS che adotta una TAA basata sulla gTAA, le procedure di test sono esempi di artefatti che possono essere riutilizzati
- B. È possibile riutilizzare gli artefatti di una TAS solamente se tale TAS adotta una TAA basata sulla gTAA
- C. La capacità di riutilizzare gli artefatti di una TAS viene determinata principalmente dalla TAA, ma può essere migliorata durante il ciclo di vita della TAS
- D. Per una TAS che adotta una TAA basata sulla gTAA, i dati di test sono esempi di artefatti che possono essere riutilizzati

Risposta corretta

B

Soluzione

A e D sono sbagliate perché sono affermazioni vere. Infatti, [a] §3.3.4 specifica che sia le procedure di test che i dati di test sono esempi di tipici artefatti riutilizzabili di una TAS (derivanti dall'adozione di una TAA basata sulla gTAA).

C è sbagliata perché è un'affermazione vera. Infatti, [a] §3.3.4 dice: "È importante notare che mentre progettare per il riuso riguarda principalmente la TAA, la manutenzione e il miglioramento del riuso rappresentano un impegno in tutto il ciclo di vita della TAS".

B è corretta perché è un'affermazione falsa. Benché la gTAA sia un'architettura astratta di riferimento utile per supportare il riutilizzo dei componenti da essa descritti in una TAA concreta basata su di essa, il riutilizzo (riuso) degli artefatti di una TAS non è ovviamente vincolato al fatto di utilizzare la gTAA per produrre la TAA.

Perciò, B è la sola risposta corretta.

4 Capitolo 4

Domanda #18 – ITA (Punti: 2)

Sezione

4.1 Selezione di un Approccio di Automazione del Testing e Pianificazione del Deployment/Rollout

Obiettivo di Apprendimento

ALTA-E-4.1.1 (K3) Applicare le linee guida a supporto di efficaci attività di pilota e deployment di uno strumento di test

Domanda

Considerate un progetto pilota di automazione del testing basato su un business case che mira a ridurre del 60% il tempo di esecuzione per una certa percentuale di test di regressione manuali di una suite di test usata per la manutenzione di un SUT. Si è scelto un progetto non critico per l'organizzazione rispetto al quale svolgere tale pilota e il management che ha approvato tale scelta non ha consapevolezza sulle difficoltà legate all'introduzione dell'automazione del testing, così come delle sue limitazioni. Le persone che lavoreranno sul pilota saranno inoltre coinvolte, in parallelo, su altri progetti approvati dal management.

Sulla base delle sole informazioni fornite, quale delle seguenti affermazioni reputate essere più probabilmente vera in questo scenario?

- A. Il management non dovrebbe essere coinvolto nel progetto pilota per via della sua mancanza di consapevolezza sull'automazione del testing
- B. La scelta di svolgere il pilota per un progetto non critico per l'organizzazione è irragionevole rispetto alle linee guida in genere raccomandate
- C. Il management dovrebbe supportare il coinvolgimento delle persone indicate nel progetto pilota, compatibilmente con le esigenze degli altri progetti
- D. Il business case non dovrebbe indirizzare la riduzione del tempo di esecuzione dei test di regressione manuali

Risposta corretta

C

Soluzione

A è sbagliata. Infatti, al contrario, il management dovrebbe essere coinvolto sia nello svolgimento che nella valutazione del progetto pilota anche per aiutarlo ad aumentare la propria consapevolezza su vantaggi, svantaggi, obiettivi e limitazioni dell'automazione del testing.

B è sbagliata. Le linee guida per un pilota raccomandano di non scegliere un progetto critico per l'organizzazione. Infatti, [a] §4.1.1 dice: "Non selezionare un progetto critico. Quando il deployment della TAS causa ritardi, ciò non deve avere impatti importanti su progetti critici. Il deployment della TAS costerà tempo all'inizio. Il team di progetto ne deve essere consapevole".

C è corretta. Infatti, [a] §4.1.1 dice: "Il pilota dovrebbe essere trattato come un normale progetto di sviluppo: fare un piano, riservare budget e risorse, effettuare report degli avanzamenti, definire milestone, ecc. Un ulteriore punto di attenzione è assicurarsi che le persone che lavoreranno sull'uso della TAS possano spendere abbastanza sforzo anche quando altri progetti richiederanno risorse per le loro attività. È importante avere l'appoggio del management, in modo particolare sulle risorse condivise. Queste persone probabilmente non potranno lavorare a tempo pieno sul pilota".

D è sbagliata. Infatti, la riduzione del tempo di esecuzione dei test, pur non essendo l'unico aspetto rilevante dell'automazione del testing, rimane un aspetto fondamentale da indirizzare in un progetto

pilota. Non si può quindi affermare che, sulla base delle sole informazioni fornite, il business case che guida il progetto pilota non debba indirizzare tale aspetto.
Perciò, C è la sola risposta corretta.

Domanda #19 – ITA (Punti: 2)

Sezione

4.1 Selezione di un Approccio di Automazione del Testing e Pianificazione del Deployment/Rollout

Obiettivo di Apprendimento

ALTA-E-4.1.1 (K3) Applicare le linee guida a supporto di efficaci attività di pilota e deployment di uno strumento di test

Domanda

Un progetto pilota di automazione del testing basato su un business case è durato 40gg come da pianificazione e si è completato con successo. La relativa TAS, che prevede l'utilizzo di uno strumento commerciale di automazione del testing (noto come TA-COM), è in fase di deployment all'interno della vostra organizzazione. Il deployment della TAS adotta una modalità di rollout incrementale che prevede il coinvolgimento di un team alla volta nell'ambito di uno specifico progetto. Al termine di ognuno di questi progetti vengono condotte delle retrospettive con i singoli team che se ne sono occupati. Sulla base delle sole informazioni fornite, quale delle seguenti affermazioni è più probabilmente vera?

- A. Adottare una modalità di rollout non-incrementale (big-bang) in fase di deployment avrebbe consentito di risparmiare sui costi per l'acquisto delle licenze di TA-COM
- B. Il progetto pilota, pur essendosi completato con successo, ha avuto una durata irragionevolmente eccessiva
- C. Durante il deployment, determinare quanti e quali componenti della TAS sono stati usati dai singoli team nei vari progetti può aiutare a valutare l'efficacia della TAS
- D. Andrebbe svolta una sola retrospettiva finale al completamento di tutte le attività di deployment che coinvolga assieme tutti i singoli team che hanno svolto i progetti

Risposta corretta

C

Soluzione

A è sbagliata. Infatti, [a] §4.1.2, nel definire i fattori di successo per il deployment, specifica quanto segue: “Un rollout incrementale: effettuare il rollout al resto dell'organizzazione a passi, per incrementi successivi. In questo modo, il supporto ai nuovi utenti avviene a “ondate” piuttosto che tutto in una volta. Ciò consente all'uso della TAS di crescere passo per passo. Eventuali colli di bottiglia possono essere identificati e risolti prima che divengano problemi reali. Ulteriori licenze possono essere aggiunte solo quando necessarie”.

B è sbagliata. Sulla base delle sole informazioni fornite, non è possibile affermare che la durata sia irragionevole (e.g., non viene specificato il business case di riferimento, quanti test debbano essere automatizzati, di che tipologia, etc.). Peraltro, 40 giorni sembra una durata del tutto ragionevole per un progetto pilota di automazione del testing.

C è corretta. Infatti, l'efficacia della TAS (che va vista in riferimento ai suoi obiettivi) può essere valutata considerando anche quanto i vari componenti di cui essa è costituita siano effettivamente utilizzati dai singoli team nei vari progetti svolti durante il deployment (si veda anche [a] §4.1.2).

D è sbagliata perché non esiste alcuna linea guida che indichi quanto specificato nella risposta D. Quanto specificato, peraltro, non sembra compatibile con una modalità di rollout incrementale. Perciò, C è la sola risposta corretta.

Domanda #20 – ITA (Punti: 3)

Sezione

4.2 Valutazione del Rischio e Strategie di Mitigazione

Obiettivo di Apprendimento

ALTA-E-4.2.1 (K4) Analizzare i rischi di deployment, identificare i problemi tecnici che potrebbero portare al fallimento di un progetto di automazione del testing e prevedere le strategie di mitigazione

Domanda

Per svolgere test automatizzati a livello di GUI su un'applicazione web di e-commerce viene adottato un framework keyword-driven. Tali test sono di due tipi: test di tipo funzionale e test mirati a verificare la corretta interazione con i controlli della GUI. Sono previsti numerosi test funzionali per verificare il corretto aggiornamento dei magazzini dei prodotti a fronte degli ordini (ordini che possono essere effettuati solo da utenti registrati). Le librerie di keyword forniscono keyword di basso livello che consentono di interagire con tutti i controlli della GUI. Il framework consente inoltre di definire e usare delle “user” keyword direttamente all'interno dei file in cui vengono implementati i test automatizzati (“test definition file”): la visibilità di tali keyword sarà limitata ai file in cui sono definite. I test automatizzati vengono condotti da un team di test indipendente che svolge anche test manuali. L'ambiente di test utilizzato è uno solo ed è dotato di un singolo database completamente accessibile sia dai tester che si occupano dei test manuali che dai tester che si occupano dei test automatizzati. Test manuali e automatizzati vengono spesso eseguiti in parallelo, non essendo previste finestre temporali per la loro esecuzione separata. L'implementazione dei test automatizzati prevede il coinvolgimento di due tipi di tester: Test Analyst e Technical Test Analyst. Sulla base delle sole informazioni fornite, quale tra le seguenti potrebbe essere più probabilmente una possibile causa di fallimento dei test automatizzati?

- A. La mancanza di isolamento nel database tra i dati usati dai test automatizzati e i dati usati dai test manuali
- B. La mancanza keyword di più alto livello all'interno delle librerie a supporto dell'implementazione dei test automatizzati di tipo funzionale
- C. La possibilità di definire e usare le “user” keyword, dato che esse non vengono aggiunte alle librerie di keyword
- D. Il coinvolgimento di Test Analyst e di Technical Test Analyst nell'implementazione dei test automatizzati

Risposta corretta

A

Soluzione

A è corretta. Si consideri, come esempio, il test funzionale citato nel seguente estratto della domanda: “Particolarmente numerosi sono i test funzionali che verificano il corretto aggiornamento dei magazzini dei prodotti a fronte degli ordini. Gli ordini possono essere effettuati solo da utenti registrati”. Tali test automatizzati avranno come risultato atteso una certa giacenza in magazzino, determinata come la giacenza iniziale a cui viene sottratta la quantità del prodotto (di una data

tipologia) acquistata tramite un ordine. Se durante l'esecuzione di questo test automatizzato la giacenza in magazzino viene modificata da un test manuale, tale test automatizzato fallirà. Sarebbe necessario isolare i prodotti e/o i magazzini usati dai test automatizzati da quelli usati dai test manuali per evitare questo tipo di interferenza. Simili problematiche di interferenza si potrebbero verificare a causa dell'utilizzo degli stessi account utente e di altre tipologie di dati. La mancanza di isolamento dei dati del database che possono essere usati come dati di test sia dai test automatizzati che dai test manuali può quindi portare ad interferenze che possono fare fallire i test automatizzati.

B è sbagliata. Le keyword disponibili nelle librerie sono orientate all'implementazione dei test automatizzati mirati a verificare la corretta interazione con i vari controlli della GUI. I test automatizzati di tipo funzionale richiederebbero delle keyword di livello più alto. Questo è certamente un problema perché porterebbe a scrivere test automatizzati molto lunghi che presenterebbero problemi di manutenzione. Tuttavia, in sé questa non rappresenta una probabile causa del fallimento dei test automatizzati.

C è sbagliata. La possibilità di definire (e usare) "user" keyword, offre semplicemente una maggiore flessibilità nell'utilizzo del framework citato, senza rappresentare in sé una probabile causa del fallimento dei test automatizzati che fanno utilizzo di questa tipologia di keyword.

D è sbagliata. Tale coinvolgimento non solo non rappresenta una probabile causa del fallimento dei test automatizzati, ma è anche raccomandato. Infatti, il framework-driven enfatizza e supporta il coinvolgimento di entrambe le figure professionali indicate attraverso un'efficace ed efficiente separazione dello sforzo di automazione corrispondente alle loro competenze (si veda [a] §3.2.2 in "Testing Keyword-driven").

Perciò, A è la sola risposta corretta.

Domanda #21 – ITA (Punti: 3)

Sezione

4.2 Valutazione del Rischio e Strategie di Mitigazione

Obiettivo di Apprendimento

ALTA-E-4.2.1 (K4) Analizzare i rischi di deployment, identificare i problemi tecnici che potrebbero portare al fallimento di un progetto di automazione del testing e prevedere le strategie di mitigazione

Domanda

Degli script di test automatizzati a livello di GUI per un'applicazione web sono organizzati in una suite di test. Tali script prevedono un'interazione diretta col codice HTML delle pagine web e sono implementati come segue: essi prima localizzano gli elementi di interesse nella pagina tramite i corrispondenti identificativi ("locators" statici), poi eseguono delle azioni su di essi, ed infine verificano la presenza di un dato testo. L'esecuzione della suite produce un file di log che nel suo nome riporta anche il timestamp relativo al termine di tale esecuzione.

Sulla base delle sole informazioni fornite, quale tra le seguenti potrebbe essere più probabilmente un problema tecnico legato a questa soluzione?

- A. Eccessiva astrazione nella progettazione degli script di test automatizzati che rende complesso comprendere cosa succede durante l'esecuzione
- B. Difficoltà crescente nella gestione delle tabelle dati usate dagli script di test automatizzati al crescere del numero di script
- C. L'indisponibilità dei log prodotti da esecuzioni precedenti all'ultima effettuata a causa della continua sovrascrittura dello stesso file di log
- D. Ridotta manutenibilità degli script di test automatizzati a causa della presenza di istanze di codice ripetuto tra tali script

Risposta corretta

D

Soluzione

A è sbagliata perché non esiste alcun livello di astrazione rilevante dato che gli script di test automatizzati interagiscono direttamente col codice HTML delle pagine web.

B è sbagliata perché non si fa nessun riferimento esplicito all'utilizzo di tabelle dati da parte degli script di test automatizzati.

C è sbagliata perché il timestamp è un'informazione univoca.

D è corretta. Infatti, la presenza di istanze di codice ripetuto e la fragilità alle modifiche della GUI (web UI) sono due tipiche problematiche legate alla soluzione adottata (spesso affrontate attraverso il pattern noto come "Page Object Model").

Perciò, D è la sola risposta corretta.

Domanda #22 – ITA (Punti: 1)

Sezione

4.3 Manutenzione dell'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-4.3.1 (K2) Comprendere quali fattori supportano e influenzano la manutenibilità di una TAS

Domanda

Quale dei seguenti fattori influenza maggiormente la facilità con la quale è possibile sostituire i componenti di una TAS al fine di supportare la manutenibilità di tale TAS?

- A. Aggiornamento del materiale formativo della TAS
- B. Grado di modularizzazione della TAS
- C. Uso consistente degli standard usati per i nomi degli artefatti della TAS
- D. Documentazione aggiornata sull'utilizzo della TAS

Risposta corretta

B

Soluzione

Tutte le risposte si riferiscono a fattori (descritti in [a] §4.3.2) che influenzano la manutenibilità di una TAS. Tuttavia, la modularità è certamente il fattore che più influenza la facilità con la quale è possibile sostituire i componenti di una TAS al fine di supportarne la manutenibilità (come indicato in [a] §4.3.2 che specifica: “La TAS deve essere modulare, così che le sue parti possano essere facilmente sostituite”).

Quindi, B è corretta, mentre A, C e D sono sbagliate.

Perciò, B è la sola risposta corretta.

5 Capitolo 5

Domanda #23 – ITA (Punti: 1)

Sezione

5.1 Selezione di Metriche per la TAS

Obiettivo di Apprendimento

ALTA-E-5.1.1 (K2) Classificare le metriche che possono essere usate per monitorare la strategia dell'automazione del testing e la sua efficacia

Domanda

Considerate i seguenti esempi di metriche relativi a progetti di automazione del testing:

- I. In media, lo sforzo per automatizzare un test di una suite di test manuali di regressione è pari a 3.2 volte lo sforzo per eseguire manualmente tale test
- II. In un team di test, il numero di test automator che dedica il 100% del proprio sforzo all'automazione dei test è pari a 4
- III. In media, 3 test automatizzati di una suite di test di regressione falliscono a causa dello stesso difetto del SUT

Quale delle seguenti classificazioni di queste metriche per una TAS è vera?

- A. I. è una metrica interna della TAS; II. e III. sono metriche esterne delle TAS
- B. I. è una metrica interna della TAS; II. è una metrica esterna della TAS; III. non è una metrica della TAS
- C. I. non è una metrica della TAS; II. è una metrica interna della TAS; III. è una metrica esterna della TAS
- D. I. e III. sono metriche esterne della TAS; II. non è una metrica della TAS

Risposta corretta

D

Soluzione

I. è una metrica esterna della TAS che si riferisce allo sforzo per realizzare i test automatizzati tramite la TAS (si veda [a] §5.1).

II. non è né una metrica interna della TAS, né una metrica esterna della TAS (eventualmente è una metrica rilevante per la gestione del progetto di automazione del testing, ma non è una metrica della TAS).

III. è una metrica esterna della TAS che si riferisce al rapporto tra fallimenti e difetti (si veda [a] §5.1). Quindi, A, B e C sono sbagliate, mentre D è corretta.

Perciò, D è la sola risposta corretta.

Domanda #24 – ITA (Punti: 1)

Sezione

5.1 Selezione di Metriche per la TAS

Obiettivo di Apprendimento

ALTA-E-5.1.1 (K2) Classificare le metriche che possono essere usate per monitorare la strategia dell'automazione del testing e la sua efficacia

Domanda

Quale delle seguenti è una metrica esterna della TAS che può essere espressa come fattore dell'EMTE (Equivalent Manual Test Effort)?

- A. Lo sforzo medio speso per aggiornare i test automatizzati di una suite di test eseguita dalla TAS a fronte di modifiche apportate al SUT
- B. Lo sforzo medio speso per correggere i difetti del SUT trovati grazie all'esecuzione di test automatizzati di una suite di test eseguita dalla TAS
- C. Lo sforzo totale di formazione speso per presentare ai test automator di un team di test tutte le feature offerte da una nuova TAS
- D. Il rapporto medio tra i commenti e le istruzioni eseguibili presenti nel codice dei test automatizzati di una suite di test eseguita dalla TAS

Risposta corretta

A

Soluzione

A è corretta in quanto si riferisce allo sforzo di manutenzione dei test automatizzati, che è una metrica esterna della TAS spesso espressa come media e come fattore dell'EMTE (si veda [a] §5.1).

B e C sono sbagliate perché non si riferiscono a metriche della TAS.

D è sbagliata perché non si riferisce ad una metrica esterna della TAS, ma ad una interna (e in ogni caso non avrebbe senso esprimere tale metrica come fattore dell'EMTE).

Perciò, A è la sola risposta corretta.

Domanda #25 – ITA (Punti: 2)

Sezione

5.2 Implementazione delle Misure

Obiettivo di Apprendimento

ALTA-E-5.2.1 (K3) Implementare metodi di raccolta di metriche per supportare requisiti tecnici e di gestione. Spiegare come le misure dell'automazione del testing possono essere implementate

Domanda

In qualità di TAE, dovete determinare come misurare lo sforzo richiesto per svolgere la manutenzione di una suite di test automatizzati di regressione a fronte di una nuova release di un SUT. Di particolare interesse sarà tracciare l'andamento nel tempo di tale metrica con riferimento alle varie release del SUT. Quale delle seguenti affermazioni non è vera?

- A. Lo sforzo indicato è correlato alla percentuale di test automatizzati della suite che richiedono manutenzione
- B. Lo sforzo indicato può essere determinato automaticamente sulla base dei risultati di esecuzione della suite
- C. Lo sforzo indicato può essere espresso come fattore dell'EMTE (Equivalent Manual Test Effort)
- D. Lo sforzo indicato è correlato al numero di test automatizzati della suite che richiedono manutenzione

Risposta corretta

B

Soluzione

A e D sono sbagliate perché sono affermazioni vere. Infatti, [a] §5.1 (si veda "Sforzo per mantenere i test automatizzati"): "Una metrica correlata è il numero o la percentuale dei test che richiedono lavoro di manutenzione".

B è corretta perché è un'affermazione falsa. Lo sforzo indicato non può essere determinato sulla base dei risultati di esecuzione della suite (tra l'altro neppure manualmente).

C è sbagliata perché è un'affermazione vera. Infatti, [a] §5.1 (si veda "Sforzo per mantenere i test automatizzati") specifica che lo sforzo indicato può essere espresso anche come fattore dell'EMTE.

Perciò, B è la sola risposta corretta.

Domanda #26 – ITA (Punti: 2)

Sezione

5.2 Implementazione delle Misure

Obiettivo di Apprendimento

ALTA-E-5.2.1 (K3) Implementare metodi di raccolta di metriche per supportare requisiti tecnici e di gestione. Spiegare come le misure dell'automazione del testing possono essere implementate

Domanda

Si consideri la build relativa alla versione X di un dato software e una suite di test automatizzati. Tutti i test della suite vengono eseguiti con successo su tale build senza dar luogo ad alcun falso negativo. Vi è stato chiesto di valutare quale sia l'efficacia di tale suite nell'identificare regressioni introdotte da modifiche apportate alla build indicata. Sulla base delle sole informazioni fornite, quali tra le seguenti metriche reputate essere la più rilevante per contribuire ad effettuare tale valutazione?

- A. Il numero totale di difetti rilevati eseguendo la suite su N build prodotte a partire dalla versione X inserendo un singolo difetto noto in ciascuna di queste build
- B. Il numero totale di test falliti ottenuti eseguendo la suite su N build prodotte a partire dalla versione X inserendo un singolo difetto noto in ciascuna di queste build
- C. Il rapporto tra numero totale di test falliti e numero totale di difetti rilevati eseguendo la suite su N build prodotte a partire dalla versione X inserendo un singolo difetto noto in ciascuna di queste build
- D. Il numero medio di difetti presente nel codice di automazione dei test automatizzati della suite ogni cento LOC (Lines Of Code)

Risposta corretta

A

Soluzione

A è corretta, mentre B e C sono sbagliate. Si consideri il seguente estratto dalle risposte A, B e C: “eseguendo la suite su N build prodotte a partire dalla versione X inserendo un singolo difetto noto in ciascuna di queste build”. Tale estratto fa riferimento ad un approccio basato sul cosiddetto “mutation testing” che prevede di inserire dei difetti (“fault seeding”) nella build originale, generando alcune varianti (“mutazioni”) di tale build, con lo scopo di valutare la capacità di rilevare tali difetti tramite l'esecuzione di test (in generale di regressione). Dato che un test automatizzato della suite potrebbe fallire senza rilevare alcun difetto (falso positivo), oppure più test automatizzati della suite potrebbero fallire a causa di uno stesso difetto, le metriche che determinano il numero di fallimenti e il rapporto tra fallimenti e difetti (pur utili per vari motivi ed infatti citate entrambe in [a] §5.1 tra le metriche esterne di una TAS) non sono così rilevanti come la metrica che determina il numero di difetti rilevati al fine di contribuire a valutare l'efficacia di tale suite di test nell'identificare regressioni introdotte da modifiche (in questo caso con riferimento specifico a quelle apportate alla build relativa alla versione X).

D è sbagliata: essa si riferisce alla densità dei difetti presenti nel codice dell'automazione, che è certamente una metrica rilevante per l'automazione (citata infatti in [a] §5.1 tra le metriche interne di una TAS). Certamente la presenza di tali difetti può impattare la capacità dei test automatizzati di rilevare eventuali regressioni nella suite, ma tale metrica si limita ad indicare la densità di tali difetti del codice senza consentire di fare alcuna valutazione diretta sulla capacità della suite nell'identificare regressioni introdotte da modifiche.

Perciò, A è la sola risposta corretta.

Domanda #27 – ITA (Punti: 3)

Sezione

5.3 Logging della TAS e del SUT

Obiettivo di Apprendimento

ALTA-E-5.3.1 (K4) Analizzare il logging dei test sia dei dati del SUT che della TAS

Domanda

Una suite di test automatizzati viene eseguita da una TAS su un ambiente di test che il SUT condivide con altre applicazioni (sulle quali la TAS non ha controllo). Tutti gli ultimi sei run della suite non si sono mai completati e il test in corrispondenza del quale l'esecuzione si è interrotta è sempre stato diverso. In qualità di TAE, ipotizzate che il problema si manifesti lato SUT a causa di errori di "out of memory". Attualmente, i dati registrati nel file di log generato dall'esecuzione non consentono di risalire alla causa. Tale file, in formato HTML, viene generato tramite il meccanismo di seguito descritto. Durante l'esecuzione vengono registrati per ogni keyword, in alcuni file intermedi, tutti i dati di logging attualmente previsti. Tali dati vengono poi inseriti nel file di log finale solo per le keyword che falliscono, mentre per quelle eseguite con successo, viene registrato solo un sottoinsieme (completamente configurabile) di tali dati. Nel file di log viene anche registrato il timestamp di inizio e fine esecuzione di ogni test della suite.

Sulla base delle sole informazioni fornite, quale delle seguenti azioni (supponendo che la TAS consenta di effettuare tutte) intraprendereste per prima per cercare di rilevare la causa del problema di interruzione dell'esecuzione della suite?

- A. Registrare nel file di log tutti i dati di logging previsti non solo per le keyword che falliscono ma anche per quelle che vengono eseguite con successo
- B. Registrare nel file di log anche la quantità di memoria disponibile per il SUT in corrispondenza dell'inizio e della fine di ogni test della suite
- C. Suddividere il file di log generato in parti più piccole caricate in file esterni che vengono caricati nel browser in modo trasparente quando necessario
- D. Registrare nel file di log i nomi delle applicazioni in esecuzione sull'ambiente di test in corrispondenza dell'inizio e della fine di ogni test della suite

Risposta corretta

B

Soluzione

B è corretta, mentre D è sbagliata. Le cause del problema potrebbero essere molteplici. Ad esempio, il problema potrebbe essere dovuto all'utilizzo di dati casuali (casuali proprio perché il test della suite in corrispondenza del quale si verifica l'interruzione sembra essere sempre diverso sulla base degli ultimi sei run) forniti in input ai test: il SUT potrebbe non essere in grado di gestire correttamente questi dati e ciò potrebbe provocare qualche errore/eccezione che porta all'interruzione (abort) dell'esecuzione. Tuttavia, in base a quanto specificato dalla domanda, voi ipotizzate che la causa del problema possa essere la mancanza di memoria a disposizione del SUT (ovvero SUT che va in "out of memory") ed è quindi innanzitutto opportuno investigare questa possibilità (peraltro concreta, dato che la domanda specifica anche che il SUT condivide l'ambiente di test con altre applicazioni sulla quali la TAS non ha controllo). Sempre come specificato dalla domanda, attualmente i dati registrati nel file di log generato dall'esecuzione non consentono di risalire alla causa del problema. L'aggiunta,

in corrispondenza dell'inizio e fine di ogni test della suite, del dato relativo alla quantità di memoria disponibile sul SUT, potrebbe aiutare a confermare o meno l'ipotesi fatta sulla causa del problema. L'aggiunta, in corrispondenza dell'inizio e fine di ogni test della suite, delle applicazioni in esecuzione sull'ambiente di test è molto meno rilevante in quanto non sapremmo quanta memoria queste stanno consumando e soprattutto quanto memoria è ancora disponibile.

A è sbagliata. Registrare nel file di log finale in modo incondizionato (il meccanismo descritto nella domanda è spesso indicato come "logging condizionale") i dati di logging previsti per tutte le keyword, potrebbe aiutare (al netto del fatto che ciò introduce serie problematiche di gestione della dimensione dei file di log). Potrebbe infatti darsi che la causa del problema non sia l'indisponibilità di memoria per il SUT e che, in ogni caso, tale causa possa essere rilevata anche guardando con maggior dettaglio i dati relativi alle keyword che vengono eseguite con successo. Tuttavia, sulla base delle sole informazioni fornite, per non andare a caso, bisogna prima concentrarsi sull'ipotesi relativa all'indisponibilità di memoria e quindi la prima azione da contemplare è quella descritta nella risposta B.

C è sbagliata. L'azione descritta è certamente utile per risolvere quelle situazioni in cui il file di log in formato HTML è troppo grande e non riesce ad essere fruito in modo adeguato dal browser (e.g., in RobotFramework esiste l'opzione --splitlog). Tuttavia, non è certamente la prima azione da prevedere per cercare di rilevare la causa del problema (inoltre, allo stato attuale, non sappiamo quali siano le dimensioni dei file di log).

Perciò, B è la sola risposta corretta.

Domanda #28 – ITA (Punti: 2)

Sezione

5.4 Reportistica dell'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-5.4.1 (K2) Spiegare come un report dell'esecuzione dei test è costruito e pubblicato

Domanda

Il report di esecuzione di una suite di test automatizzati dovrebbe:

- A. indicare, per ogni test della suite, i timestamp di inizio, di fine e di ogni passo intermedio eseguito
- B. indicare sia il SUT che l'ambiente di test nel quale i test della suite sono stati eseguiti
- C. indicare più dettagli possibile sui fallimenti rilevati durante l'esecuzione dei test della suite per consentire agli sviluppatori di riprodurli
- D. essere generato da un componente dedicato nella TAS concepito a livello di TAA nel layer di definizione del testing

Risposta corretta

B

Soluzione

A e C sono sbagliate perché si riferiscono ad informazioni che dovrebbero essere fornite dai log di esecuzione, non dal report di esecuzione.

B è corretta. Infatti, sia il SUT che l'ambiente di test indicati sono informazioni che dovrebbero essere contenute all'interno del report di esecuzione di tale suite (si veda [a] §5.4).

D è sbagliata. Infatti, il report di esecuzione indicato dovrebbe essere generato da un componente dedicato nella TAS concepito a livello di TAA nel layer di esecuzione del testing, non di definizione del testing.

Perciò, B è la sola risposta corretta.

6 Capitolo 6

Domanda #29 – ITA (Punti: 2)

Sezione

6.1 Criteri per l'Automazione

Obiettivo di Apprendimento

ALTA-E-6.1.1 (K3) Applicare criteri per determinare l'idoneità dei test all'automazione

Domanda

State attualmente lavorando come TAE all'interno di un'organizzazione. La seguente tabella si riferisce ai test di quattro suite di test manuali di regressione (RTS1, RTS2, RTS3, RTS4) eseguite a livello di web UI per delle applicazioni web all'interno della vostra organizzazione (ogni suite di test viene usata per testare un'applicazione web diversa sviluppata con un framework di web UI diverso):

Suite di Test	F	C	UI
RTS1	4	2	Y
RTS2	4	3	Y
RTS3	1	2	Y
RTS4	3	3	N

I valori nella colonna F indicano la frequenza d'uso dei test appartenenti ad una suite di test in una scala crescente a 5 valori (1, 2, 3, 4, 5), in cui quindi il valore di frequenza minima è pari a 1 e quello di frequenza massima è pari a 5.

I valori nella colonna C indicano la complessità richiesta per automatizzare i test appartenenti ad una suite di test in una scala crescente a 3 valori (1, 2, 3), in cui quindi la complessità minima è pari a 1 e la complessità massima è pari a 3.

I valori nella colonna UI indicano la disponibilità, all'interno dell'organizzazione, di strumenti di automazione compatibili con i framework di web UI dell'applicazione web sulla quale vengono eseguiti i test appartenenti ad una suite di test ("Y" = disponibile, "N" = non disponibile). La vostra organizzazione dispone anche di uno strumento di automazione per implementare i test automatizzati a livello di backend per la sola applicazione web testata dalla suite RTS4.

Sulla base delle sole informazioni fornite, quale suite di test repute essere la più adatta ad essere automatizzata?

- A. RTS1
- B. RTS2
- C. RTS3
- D. RTS4

Risposta corretta

A

Soluzione

Non viene fornita alcuna informazione né sull'architettura dell'applicazione web testata a livello di UI da RTS4, né sull'obiettivo dei test di tale suite. Quindi, sulla base delle sole informazioni fornite, non possiamo determinare se l'eventuale automazione dei test di RTS4 a livello di backend possa dare valore (e.g., magari si tratta di test che pur essendo eseguiti manualmente a livello di UI hanno come

obiettivo quello di testare la business logic dell'applicazione e sulla base dell'architettura applicativa potrebbe anche aver senso implementarli a livello di backend). Quindi, D è sbagliata.

Tra le altre tre suite, RTS1 (sulla base di quanto indicato in [a] §6.1):

- è più adatta ad essere automatizzata di RTS2 in quanto, pur avendo lo stesso valore di F, ha un valore inferiore di C
- è più adatta ad essere automatizzata di RTS3 in quanto, pur avendo lo stesso valore di C, ha un valore superiore di F

Quindi, A è corretta, mentre B e C sono sbagliate.

Perciò, A è la sola risposta corretta.

Domanda #30 – ITA (Punti: 1)

Sezione

6.1 Criteri per l'Automazione

Obiettivo di Apprendimento

ALTA-E-6.1.2 (K2) Comprendere i fattori in gioco nella transizione dal testing manuale a quello automatizzato

Domanda

La transizione dal testing manuale al testing automatizzato:

- A. è raccomandata anche per un sistema prossimo al ritiro che sarà sostituito da un nuovo sistema, a prescindere dalle funzionalità implementate e dalle architetture dei due sistemi coinvolti
- B. dovrebbe essere supportata da un progetto pilota che si concentri sull'automazione dei test tecnicamente più complessi, in modo che il successo di tale progetto assicuri la fattibilità dell'automazione per tutti i test
- C. richiede generalmente che sviluppatori e tester lavorino molto più a stretto contatto rispetto a quanto avviene in ambiti nei quali il testing venga svolto soltanto manualmente
- D. non può essere svolta da team separati dai team di progetto, ma solamente dai team di progetto che al proprio interno devono includere sia tester con competenze tecniche che tester con competenze di dominio

Risposta corretta

C

Soluzione

A è sbagliata. Infatti, [a] §6.1 (si veda “Idoneità dell'automazione per la fase del ciclo di vita del prodotto software”) dice: “L'automazione non è raccomandata per un sistema vicino alla fine del suo ciclo di vita poiché sarà di poco valore intraprendere un'iniziativa con una vita così breve. Comunque, per sistemi che stanno per essere riprogettati usando un'architettura differente, ma preservando le funzionalità esistenti, un ambiente di test automatizzato che definisca i dati sarà ugualmente utile nei vecchi e nei nuovi sistemi. In questo caso, il riuso dei dati di test sarebbe possibile e si renderebbe necessario ricodificare l'ambiente automatizzato per ottenere la compatibilità con la nuova architettura”.

B è sbagliata. Infatti, [a] §6.1 (si veda “Ambito dello sforzo di automazione del testing”) dice: “In aggiunta, deve essere data priorità ai test che sono critici per l'organizzazione, poiché questi mostreranno all'inizio il valore più grande. Comunque, all'interno di questo contesto, è importante che come parte dello sforzo per il pilota, si eviti di automatizzare i test tecnicamente più sfidanti. Altrimenti, sarà impiegato troppo sforzo per cercare di sviluppare l'automazione e ci saranno troppi pochi risultati da mostrare. Come regola generale, identificare i test che condividono

caratteristiche/funzionalità con gran parte dell'applicazione fornirà lo slancio per mantenere vivo lo sforzo di automazione”.

C è corretta. Infatti, [a] §6.1 (si veda “Cooperazione fra sviluppatori e TAE”) dice: “Il successo dell'automazione del testing richiede il coinvolgimento del team di sviluppo del software e dei tester. Sviluppatori e tester dovranno lavorare molto più a stretto contatto per l'automazione del testing in modo tale che gli sviluppatori possano fornire supporto personale e informazioni tecniche sui loro metodi e strumenti di sviluppo”.

D è sbagliata: “non può” rende l'affermazione falsa. Infatti, [a] §6.1 (si veda “Sforzo parallelo”) dice: “Come parte delle attività di transizione, molte organizzazioni creano un team parallelo per iniziare il processo di automazione degli script manuali di test esistenti. I nuovi script automatizzati sono poi incorporati nel processo di testing, rimpiazzando gli script manuali”.

Perciò, C è la sola risposta corretta.

Domanda #31 – ITA (Punti: 1)

Sezione

6.2 Identificare i passi necessari per implementare l'Automazione per il Testing di Regressione

Obiettivo di Apprendimento

ALTA-E-6.2.1 (K2) Spiegare i fattori da considerare nell'implementazione del testing di regressione automatizzato

Domanda

Si consideri una suite di test di regressione automatizzati che viene normalmente lanciata su un singolo SUT. Sulla base delle sole informazioni fornite, quale tra le seguenti situazioni reputate essere la più adatta per poter sfruttare un'esecuzione concorrente su più istanze dello stesso SUT (tutte aventi la stessa versione) dei test della suite al fine di ridurre il tempo di esecuzione complessivo di tale suite?

- A. I test della suite sono molti, lenti da eseguire e indipendenti tra loro
- B. I test della suite sono veloci da eseguire e implementati con un approccio keyword-driven
- C. I test della suite sono molti, dipendenti tra loro e implementati con un approccio data-driven
- D. I test sono lenti da eseguire e le post-condizioni di un test della suite coincidono con le precondizioni del test successivo

Risposta corretta

A

Soluzione

Il prerequisito principale per poter effettuare un'esecuzione concorrente dei test della suite su più SUT che consenta di ridurre il tempo di esecuzione della suite è che i test della suite siano indipendenti tra loro (si veda [a] §6.2).

A è corretta perché è l'unica che fa riferimento all'indipendenza dei test della suite.

B è sbagliata perché non fa alcun riferimento all'indipendenza dei test della suite. Inoltre, fa riferimento ad una situazione che è comunque meno adatta di A, in quanto indica che i test della suite sono veloci da eseguire. L'uso dell'approccio keyword-driven non è rilevante.

C è sbagliata perché indica esplicitamente che i test della suite sono dipendenti tra loro. L'uso dell'approccio data-driven non è rilevante.

D è sbagliata perché indica un vincolo che rende i test della suite dipendenti tra loro.

Perciò, A è la sola risposta corretta.

Domanda #32 – ITA (Punti: 1)

Sezione

6.3 Fattori da considerare quando si implementa l'Automazione per il Testing di Nuove Funzionalità

Obiettivo di Apprendimento

ALTA-E-6.3.1 (K2) Spiegare i fattori da considerare nell'implementazione dell'automazione per il testing di nuove funzionalità

Domanda

Quale delle seguenti affermazioni non è vera?

- A. Nei framework keyword-driven, l'automazione di test relativi a nuove funzionalità del SUT potrebbe richiedere l'aggiunta e/o la modifica di nuove keyword
- B. A fronte di modifiche apportate alla TAS per supportare l'automazione di test relativi a nuove funzionalità del SUT, dovrebbero essere eseguiti dei test di regressione sia funzionali che non-funzionali sulla TAS
- C. L'automazione di test relativi a nuove funzionalità del SUT potrebbe trarre vantaggio da valutazioni di testabilità condotte durante la progettazione di tali funzionalità
- D. In progetti Agile, i test relativi a nuove funzionalità del SUT dovrebbero essere esclusivamente manuali mentre i test di regressione del SUT dovrebbero essere automatizzati il più possibile

Risposta corretta

D

Soluzione

A è sbagliata perché è un'affermazione vera. Infatti, [a] §6.3 dice: "Per esempio, se si usa un approccio al testing keyword-driven, può essere necessario sviluppare keyword aggiuntive o modificare/espandere le keyword esistenti per adattare alle nuove funzionalità".

B è sbagliata perché è un'affermazione vera. Infatti, [a] §6.3 dice: "I cambiamenti alla TAS devono essere valutati rispetto ai componenti esistenti del testware in modo tale che i cambiamenti o le aggiunte siano pienamente documentati e non influenzino il comportamento (o le prestazioni) delle funzionalità esistenti della TAS".

C è sbagliata perché è un'affermazione vera. Infatti, [a] §6.3 dice: "Quando si introduce una nuova funzionalità, i test engineer hanno l'opportunità di assicurare che la nuova funzionalità sia testabile. Durante la fase di progettazione, si deve tener conto del testing pianificando di fornire le interfacce di test che possano essere usate dai linguaggi di scripting o dallo strumento di automazione del testing per verificare la nuova funzionalità".

D è corretta perché è un'affermazione falsa. In progetti Agile, il testing manuale, pur rimanendo di indubbio valore (di questo si tiene infatti conto nel quadrante del testing Q3, descritto nella tassonomia affermata in ambito Agile come 'quadranti del testing'), rimane comunque circoscritto. I test di nuove funzionalità sono di fatto i test mirati a coprire i criteri di accettazione delle user story. Tali test (descritti dal quadrante Q2 dei 'quadranti del testing') dovrebbero essere automatizzati il più possibile. Pratiche di sviluppo fondamentali come BDD e ATDD mirano addirittura a fare in modo che tali test vengano automatizzati guidando lo sviluppo. Si consideri poi che i test di nuove funzionalità sviluppati nell'ambito di un'iterazione possono poi essere incorporati nei test di regressione che verranno eseguiti nelle iterazioni successive. Ricordiamo che tali test di regressione devono essere

automatizzati il più possibile proprio per gestire il rischio di regressione che è particolarmente rilevante in tali progetti (per via delle frequenti modifiche al codice).
Perciò, D è la sola risposta corretta.

Domanda #33 – ITA (Punti: 1)

Sezione

6.4 Fattori da considerare quando si implementa l'Automazione del Testing Confermativo

Obiettivo di Apprendimento

ALTA-E-6.4.1 (K2) Spiegare i fattori da considerare nell'implementazione del testing confermativo automatizzato

Domanda

L'automazione di un test confermativo associato ad un dato difetto:

- A. mira a supportare gli sviluppatori ad effettuare più velocemente le attività di localizzazione di tale difetto all'interno del codice e di correzione
- B. richiede uno sforzo di implementazione che è tanto maggiore quante più volte tale difetto è stato riaperto
- C. richiede l'utilizzo di una tecnica di scripting data-driven per l'implementazione e viene svolta a livello di API
- D. può aiutare a rilevare più facilmente il rimanifestarsi di tale difetto nelle release successive a quella in cui è avvenuta la conferma della correzione

Risposta corretta

D

Soluzione

A è sbagliata perché si riferisce ad attività tipiche del debugging e l'automazione di un test confermativo non mira certo a supportare tali attività. Come descritto nella descrizione sottostante per la risposta D, l'automazione di un test confermativo può aiutare a ridurre il tempo di esecuzione e supportare nel rilevamento del difetto associato qualora questo si ripresentasse nelle release successive, per via, per esempio, di problemi di gestione della configurazione (configuration management).

B è sbagliata perché non esiste alcuna relazione nota che legghi lo sforzo di automazione di un test confermativo al numero di riaperture del difetto associato.

C è sbagliata perché i fattori che determinano che interfaccia del SUT utilizzare per l'automazione e la tecnica di scripting per implementare tale automazione dipendono da diversi fattori. L'automazione a livello di API e l'utilizzo di una tecnica di scripting data-driven non sono certamente, di per sé, caratteristiche dell'automazione del testing confermativo.

D è corretta. Infatti, [a] §6.4.1 dice: "I difetti hanno la capacità di riprodursi nelle release successive (ciò può indicare un problema di gestione della configurazione) e quindi i test confermativi sono candidati primari per l'automazione. L'uso dell'automazione aiuterà a ridurre il tempo di esecuzione per il testing confermativo. Il testing confermativo può essere aggiunto all'attuale insieme dei test di regressione automatizzata e completarlo".

Perciò, D è la sola risposta corretta.

7 Capitolo 7

Domanda #34 – ITA (Punti: 2)

Sezione

7.1 Verificare i Componenti dell'Ambiente di Test Automatizzato

Obiettivo di Apprendimento

ALTA-E-7.1.1 (K3) Verificare la correttezza di un ambiente di test automatizzato incluso il setup dello strumento di test

Domanda

In qualità di TAE, state svolgendo tutti i controlli mirati a verificare la correttezza di un ambiente di test automatizzato per una TAS che verrà usata per effettuare test automatizzati a livello di GUI su delle applicazioni web. Sulla base delle sole informazioni fornite, quale deli seguenti non è un controllo relativo all'ambito di verifica di cui vi state occupando?

- A. Controllare la generazione dei log da parte della TAS
- B. Controllare il comportamento funzionale dei componenti del framework
- C. Controllare la connettività verso sistemi/interfacce interni ed esterni
- D. Controllare la verifica delle post-condizioni di tutti i test automatizzati

Risposta corretta

D

Soluzione

A è sbagliata perché si riferisce ad un controllo nell'ambito della verifica della correttezza di un ambiente di test automatizzato. Infatti, [a] §7.1 (si veda “Script di test con esiti di esecuzione positivi (pass) e negativi (fail) noti”) dice: “È importante controllare la corretta generazione dei file di log e delle metriche delle prestazioni come pure setup e teardown automatizzati dei casi/script di test”.

B è sbagliata perché si riferisce ad un controllo nell'ambito della verifica della correttezza di un ambiente di test automatizzato. Infatti, [a] §7.1 (si veda “Testing dei componenti del framework”) dice: “Come ogni altro progetto di sviluppo software, i componenti del framework automatizzato devono essere testati e verificati individualmente. Questo può includere testing funzionale e non-funzionale (prestazioni, utilizzo delle risorse, usabilità, ecc.)”.

C è sbagliata perché si riferisce ad un controllo nell'ambito della verifica della correttezza di un ambiente di test automatizzato. Infatti, [a] §7.1 (si veda “Connettività verso sistemi/interfacce interni ed esterni”) dice: “Una volta che una TAS è installata in un dato ambiente del SUT e prima del suo reale utilizzo sul SUT, devono essere eseguiti una serie di controlli o precondizioni per assicurare che la connettività verso sistemi interni ed esterni, interfacce, ecc., sia disponibile”.

D è corretta perché si riferisce ad un controllo fuori dall'ambito della verifica della correttezza di un ambiente di test automatizzato. Infatti, tale controllo è descritto nell'ambito della verifica di una suite di test automatizzati (si veda [a] §7.2 in “Controllare che siano presenti sufficienti punti di verifica nella suite dei test automatizzati”).

Perciò, D è la sola risposta corretta.

Domanda #35 – ITA (Punti: 2)

Sezione

7.1 Verificare i Componenti dell'Ambiente di Test Automatizzato

Obiettivo di Apprendimento

ALTA-E-7.1.1 (K3) Verificare la correttezza di un ambiente di test automatizzato incluso il setup dello strumento di test

Domanda

Il deployment di una TAS è stato effettuato con successo in un primo ambiente di test automatizzato. Tale TAS, perfettamente funzionante e usata regolarmente per eseguire una suite di test automatizzati, è stata predisposta (ovvero installata e configurata) tramite una procedura manuale ben documentata che prevede di trasferire manualmente i file richiesti (eseguibili, librerie, file di configurazione, ecc.) nelle opportune cartelle all'interno dell'ambiente. La stessa versione della TAS è stata poi predisposta in un secondo ambiente. In qualità di TAE, state svolgendo tutti i controlli necessari atti a verificare la corretta predisposizione della TAS in questo secondo ambiente, nel quale la TAS è stata invece predisposta lanciando script automatizzati che prendono i file da un repository centrale. La TAS dovrà essere predisposta a breve in ulteriori ambienti e bisognerà garantire che essa abbia la stessa versione in tutti questi ambienti a fronte di qualsiasi suo aggiornamento. Sulla base delle sole informazioni fornite, quale delle seguenti affermazioni relative ai controlli che state svolgendo è vera?

- A. Gli script automatizzati che hanno predisposto la TAS oggetto della vostra verifica, non dovrebbero prendere i file da un repository centrale, ma dovrebbero copiarli automaticamente dal primo ambiente al secondo ambiente
- B. Tra gli altri controlli che dovrete effettuare per la TAS oggetto della vostra verifica, è incluso il controllo della presenza di sufficienti punti di verifica nella suite dei test automatizzati lanciata da tale TAS
- C. Gli script automatizzati che hanno predisposto la TAS oggetto della vostra verifica, andrebbero dismessi e bisognerebbe predisporre nuovamente la TAS, dopo averla disinstallata, tramite la procedura manuale usata nel primo ambiente
- D. Tra gli altri controlli che dovrete effettuare per la TAS oggetto della vostra verifica, è incluso il controllo della connettività verso sistemi/interfacce interni ed esterni

Risposta corretta

D

Soluzione

A e C sono sbagliate. [a] §7.1 (si veda "Installazione, setup, configurazione e personalizzazione dello strumento di test") dice "L'installazione automatizzata (o la copia) da un repository centrale ha dei vantaggi. Essa permette di garantire che i test su SUT differenti siano stati eseguiti con la stessa versione della TAS e la stessa configurazione della TAS, dove questa sia appropriata. Aggiornamenti alla TAS possono essere fatti per mezzo del repository. L'uso di repository e il processo per aggiornare la TAS ad una nuova versione devono essere gli stessi degli strumenti di sviluppo standard". Lo scenario in effetti specifica che: "Inoltre, la TAS dovrà inoltre essere predisposta a breve in ulteriori ambienti e bisognerà garantire che in ogni momento la TAS abbia la stessa versione in tutti gli ambienti in cui opererà a fronte di qualsiasi suo aggiornamento".

B è sbagliata perché si riferisce ad un controllo al di fuori dell'ambito della vostra verifica, in quanto atto a verificare la suite di test automatizzati, non la predisposizione dalla TAS (si veda [a] §7.2 in "Controllare che siano presenti sufficienti punti di verifica nella suite dei test automatizzati").

D è corretta perché si riferisce ad un controllo (per il quale non vi è indicazione alcuna che sia già stato effettuato) da effettuare nell'ambito della vostra verifica, come descritto in [a] §7.1 (si veda "Connettività verso sistemi/interfacce interni ed esterni").

Perciò, D è la sola risposta corretta.

Domanda #36 – ITA (Punti: 2)

Sezione

7.2 Verificare la Suite di Test Automatizzati

Obiettivo di Apprendimento

ALTA-E-7.2.1 (K3) Verificare il corretto comportamento per un dato script di test automatizzato e/o una suite di test automatizzati

Domanda

Una TAS è stata predisposta correttamente per funzionare in un ambiente di test automatizzato. Tale predisposizione ha compreso anche installazione, configurazione e setup dello strumento di esecuzione dei test automatizzati. Potete quindi assumere che tutte le verifiche relative siano state svolte con successo. In qualità di TAE, state attualmente conducendo varie esecuzioni di una suite di test automatizzati tramite tale TAS. Al termine di queste esecuzioni osservate che fallisce sempre lo stesso test della suite mentre tutti gli altri passano. Le esecuzioni possono essere considerate ripetibili e affidabili per tutti i test della suite. Una rapida analisi evidenzia che il fallimento del test indicato rappresenta un falso positivo dovuto ad un'errata implementazione del test stesso. Tutti i test della suite sono indipendenti tra loro. La pianificazione di progetto impone tempi molto stretti che non consentono di correggere tale falso positivo e ci si attende che la suite di test dovrà essere rieseguita ancora alcune volte prima che la correzione possa essere implementata. Sulla base delle sole informazioni fornite, quale delle seguenti azioni svolgereste in questo scenario?

- A. Rimuovere temporaneamente dalla suite di test il test che fallisce fino a quando tale test verrà corretto
- B. Rieseguire l'intera suite più lentamente tramite l'introduzione di opportuni ritardi tra i vari test
- C. Rieseguire esclusivamente il test che fallisce in modo separato dall'esecuzione di tale test all'interno della suite
- D. Verificare la connettività della TAS verso i sistemi (interni ed esterni) con cui essa dovrà interagire

Risposta corretta

A

Soluzione

A è corretta. Quando è chiaro che il test ha un problema, e il fallimento è un falso allarme, in pratica capita spesso che, a causa di pressioni sui tempi, il test non venga corretto subito. Nelle esecuzioni successive, questo difetto si potrebbe rimaniifestare e ciò richiederebbe di spendere ulteriore sforzo di analisi del fallimento. È quindi importante correggere questo difetto nel test il prima possibile o rimuoverlo dalla suite fino a quando verrà corretto. In caso contrario, il suo risultato non sarà affidabile ed il fallimento del test comporterà dei costi senza fornire alcun valore aggiunto.

B è sbagliata. La domanda specifica che le esecuzioni della suite di test possono essere considerate ripetibili e affidabili per tutti i test. Sulla base delle sole informazioni fornite, non c'è motivo per ritenere

che introdurre dei ritardi tra i vari test possa influenzare positivamente l'esito di esecuzione (come implicitamente sottinteso dalla domanda) per il test fallito.

C è sbagliata. La domanda specifica, inoltre, che tutti i test della suite sono indipendenti tra loro e che le esecuzioni possono essere considerate ripetibili e affidabili per tutti i test. In tal caso l'esecuzione separata del test indicato da tutti gli altri test della suite non ne cambierebbe l'esito.

D è sbagliata. La domanda specifica che l'esito del test che fallisce rappresenta un falso allarme dovuto ad una implementazione errata del test stesso. Il problema è quindi nel test stesso e l'azione descritta in D non aggiungerebbe alcun valore. Ciò è peraltro confermato dal fatto la domanda specifica anche quanto segue: "Una TAS è stata predisposta correttamente per funzionare in un ambiente di test automatizzato. Tale predisposizione ha compreso anche installazione, configurazione e setup dello strumento di esecuzione dei test automatizzati. Potete quindi assumere che tutte le verifiche relative siano state svolte con successo".

Perciò, A è la sola risposta corretta.

Domanda #37 – ITA (Punti: 2)

Sezione

7.2 Verificare la Suite di Test Automatizzati

Obiettivo di Apprendimento

ALTA-E-7.2.1 (K3) Verificare il corretto comportamento per un dato script di test automatizzato e/o una suite di test automatizzati

Domanda

Supponete che una TAS sia stata predisposta correttamente per funzionare in un ambiente di test automatizzato. Tale predisposizione ha compreso anche installazione, configurazione e setup dello strumento di esecuzione dei test automatizzati. Potete quindi assumere che tutte le verifiche relative siano state svolte con successo. In qualità di TAE, state attualmente svolgendo tutti i controlli mirati a verificare la correttezza di una suite di test automatizzati a livello di API che verranno eseguiti dalla TAS su dei servizi web. Come parte di questi controlli finora vi siete limitati a svolgere con successo i controlli atti a verificare la completezza della suite. Sulla base delle sole informazioni fornite, quale dei seguenti è un controllo, relativo all'ambito di verifica di cui vi state occupando, che dovrete ancora svolgere?

- A. Controllare la presenza dei risultati attesi nei test automatizzati
- B. Controllare la connettività verso sistemi/interfacce interni ed esterni
- C. Controllare la ripetibilità dei test automatizzati
- D. Controllare le prestazioni dei componenti del framework

Risposta corretta

C

Soluzione

A è sbagliata perché si riferisce ad un controllo, nell'ambito della verifica della correttezza di un ambiente di test automatizzato, già svolto. Infatti, la domanda dice "Come parte di questi controlli finora vi siete limitati a svolgere con successo i controlli atti a verificare la completezza della suite" e la risposta A si riferisce proprio ad uno di questi controlli di completezza (si veda [a] §7.2 in "Controllare la suite di test").

B è sbagliata perché si riferisce ad un controllo nell'ambito della verifica della correttezza di un ambiente di test automatizzato. Infatti, [a] §7.1 (si veda "Connettività verso sistemi/interfacce interni ed esterni") dice: "Una volta che una TAS è installata in un dato ambiente del SUT e prima del suo reale utilizzo sul SUT, devono essere eseguiti una serie di controlli o precondizioni per assicurare che la connettività verso sistemi interni ed esterni, interfacce, ecc., sia disponibile".

C è corretta perché si riferisce ad un controllo, nell'ambito della verifica della correttezza di un ambiente di test automatizzato, che non è ancora stato svolto (si veda [a] §7.2 in "Considerare la ripetibilità dei test").

D è sbagliata perché si riferisce ad un controllo nell'ambito della verifica della correttezza di un ambiente di test automatizzato. Infatti, [a] §7.1 (si veda "Testing dei componenti del framework") dice: "Come ogni altro progetto di sviluppo software, i componenti del framework automatizzato devono essere testati e verificati individualmente. Questo può includere testing funzionale e non-funzionale (prestazioni, utilizzo delle risorse, usabilità, ecc.)".

Perciò, C è la sola risposta corretta.

8 Capitolo 8

Domanda #38 – ITA (Punti: 3)

Sezione

8.1 Opzioni per Migliorare l'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-8.1.1 (K4) Analizzare gli aspetti tecnici di una soluzione di automazione del testing rilasciata e fornire le raccomandazioni per il suo miglioramento

Domanda

Considerate una TAS utilizzata per eseguire una suite di test composta da 40 test automatizzati che sono stati implementati con un approccio di scripting strutturato. Tali test sono indipendenti l'uno dall'altro e ognuno di essi deve eseguire azioni sulla stessa immagine K (di grande dimensione) di un database e modificare un singolo record in K. Le feature di setup e teardown fornite dalla TAS sono disponibili sia a livello di singolo test che di suite di test. Attualmente, ogni singolo test invoca la feature di setup (in cui viene fatto il load di K) e quella di teardown (in cui viene fatto l'unload di K). Si assuma che l'adozione di un framework keyword-driven per implementare i test della suite venga considerata troppo costosa. Sulla base delle sole informazioni fornite, quale delle seguenti raccomandazioni fornireste, per migliorare la TAS?

- A. Rimuovere il load e l'unload di K nei setup e teardown a livello di ogni singolo test ed eseguirli nei setup e teardown a livello di suite di test, senza più effettuare alcuna azione, né nel setup né nel teardown, a livello di singolo test. Questo migliorerebbe il tempo di esecuzione della suite, mantenendo l'indipendenza tra i test
- B. Adottare un framework process-driven per implementare i test automatizzati, al posto dell'implementazione attuale basata su un approccio di scripting strutturato, per ridurre lo sforzo di aggiunta di nuovi test automatizzati alla suite. Non è possibile modificare né il setup né il teardown a livello di singolo test perché ciò non consentirebbe di mantenere l'indipendenza tra i test nella suite
- C. Adottare un approccio di scripting lineare per implementare i test automatizzati, al posto dell'implementazione attuale basata su un approccio di scripting strutturato, per ridurre lo sforzo di aggiunta di nuovi test automatizzati alla suite. Non è possibile modificare né il setup né il teardown a livello di singolo test perché ciò non consentirebbe di mantenere l'indipendenza tra i test nella suite
- D. Rimuovere il load e l'unload di K nei setup e teardown a livello di ogni singolo test ed eseguirli nei setup e teardown a livello di suite di test, disfacendo le azioni ("undo") svolte da ogni test su K nel corrispondente teardown a livello di singolo test. Questo migliorerebbe il tempo di esecuzione della suite, mantenendo l'indipendenza tra i test

Risposta corretta

D

Soluzione

D è corretta, mentre A è sbagliata. Dato che K è molto grande e ci sono un discreto numero (40) di test automatizzati (che hanno le stesse caratteristiche in termini di precondizioni e azioni), effettuare il load di K prima dell'esecuzione di ognuno di essi tramite la feature di setup (e l'unload di K dopo l'esecuzione tramite la feature di teardown), porta ad un'esecuzione molto lenta. Con la raccomandazione fornita, il tempo di esecuzione della suite verrebbe significativamente ridotto. Per poter continuare però a garantire l'indipendenza dei test, è necessario disfare le azioni svolte da ogni

singolo test, perché altrimenti un test all'interno della suite si troverebbe un database che non contiene K, ma bensì K modificata dai test precedenti della suite.

B e C sono sbagliate. Innanzitutto, come sopra discusso per le risposte D e A, il fatto che tutti i test automatizzati rimangano indipendenti tra loro non impone che essi effettuino le azioni di setup e teardown a livello di singolo test specificate nello scenario. Inoltre, l'adozione di un framework process-driven (che potenzialmente potrebbe ridurre lo sforzo di aggiunta di nuovi test automatizzati alla suite) non è fattibile, in quanto la domanda specifica quanto segue: "Si assuma che l'adozione di un framework keyword-driven per implementare i test della suite venga considerata troppo costosa". Ricordiamo infatti che un framework process-driven viene costruito "on-top" su uno keyword-driven e risulta ancora più costoso. L'adozione poi di un approccio di scripting lineare addirittura aumenterebbe in modo significativo lo sforzo di aggiunta di nuovi test automatizzati alla suite.

Perciò, D è la sola risposta corretta.

Domanda #39 – ITA (Punti: 3)

Sezione

8.1 Opzioni per Migliorare l'Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-8.1.1 (K4) Analizzare gli aspetti tecnici di una soluzione di automazione del testing rilasciata e fornire le raccomandazioni per il suo miglioramento

Domanda

Considerate una TAS che viene utilizzata per eseguire, su una singola macchina dotata di un processore single-core, una suite di test automatizzati per la UI di un'applicazione web. I test sono scritti manipolando direttamente il codice HTML della pagina web: essi prima localizzano gli elementi di interesse nella pagina tramite i corrispondenti identificativi ("locators" statici), poi eseguono delle azioni su di essi, ed infine verificano la presenza di un dato testo. Inoltre, tali test gestiscono l'attesa necessaria per il caricamento di ogni pagina web da parte del web server, attendendo, in modalità "polling", il verificarsi di una certa condizione sulla pagina fino allo scadere di un opportuno time-out. All'interno della suite, ogni test dipende strettamente dal precedente. Attualmente, il tempo di esecuzione della suite è troppo lento e i test richiedono uno sforzo molto elevato di manutenzione a fronte di modifiche agli identificativi degli elementi delle pagine.

Sulla base delle sole informazioni fornite, quale delle seguenti raccomandazioni fornireste, per migliorare la TAS?

- A. Adottare un meccanismo di attesa per il caricamento di ogni pagina web da parte del web server che preveda di attendere un tempo fissato e definito a priori (potenzialmente diverso per ogni pagina)
- B. Memorizzare gli identificativi di tutti gli elementi HTML necessari in una posizione centralizzata (e.g., un file) e modificare i test in modo che facciano riferimento ad essa per individuare tali elementi
- C. Partizionare i test della suite in più sotto-suite di test separate ed eseguire tali sotto-suite in parallelo su più macchine che devono essere aggiunte a quella attualmente utilizzata
- D. Adottare un approccio basato su scripting lineare per implementare i test della suite di test, dato che tali test sono scritti per la UI di un'applicazione web

Risposta corretta

B

Soluzione

A è sbagliata. Infatti, utilizzare il meccanismo di attesa specificato nella risposta A (attesa statica) renderebbe i test automatizzati più fragili rispetto al meccanismo attualmente implementato (attesa dinamica basata su “polling”).

B è corretta. Infatti, con la raccomandazione fornita nella risposta B, la modifica di un locator avverrebbe in modo centralizzato una sola volta. Ciò eviterebbe di dover identificare quali test automatizzati facciano riferimento a tale locator e di effettuare la modifica in corrispondenza di tutti questi test.

C è sbagliata. Infatti, pur potendo potenzialmente rappresentare un modo per ridurre il tempo di esecuzione della suite, la risposta C si riferisce ad un miglioramento che non è possibile implementare in questo scenario, dato che la domanda specifica quanto segue: “All’interno della suite, ogni test dipende strettamente dal precedente”.

D è sbagliata. Infatti, prima di tutto, non esiste alcun riferimento nello scenario che descriva l’approccio attualmente utilizzato per l’implementazione dei test automatizzati. In secondo luogo, il nesso causale (“dato che tali test sono scritti per la UI di un’applicazione web”) non ha alcun fondamento (si noti inoltre che lo scripting lineare è un approccio più che altro ‘dimostrativo’ che non consente di per sé di gestire la manutenzione dei test in modo ragionevolmente efficace).

Perciò, B è la sola risposta corretta.

Domanda #40 – ITA (Punti: 3)

Sezione

8.2 Pianificare l’Implementazione dei Miglioramenti per l’Automazione del Testing

Obiettivo di Apprendimento

ALTA-E-8.2.1 (K4) Analizzare il testware automatizzato, inclusi i componenti dell’ambiente di test, gli strumenti e le librerie di funzioni a supporto, per capire dove dovrebbero essere fatti consolidamenti e aggiornamenti a seguito di un dato insieme di cambiamenti nell’ambiente di test o del SUT

Domanda

Il deployment di una TAS è stato effettuato circa un anno fa e finora le attività di testing automatico sul SUT tramite tale TAS sono state svolte senza particolari problemi. In corrispondenza delle ultime release del SUT, le esecuzioni notturne di una suite di test automatizzati non si sono mai completate a causa di alcuni servizi/componenti del SUT andati in crash. Con differenti release del SUT, il test della suite in cui si verifica l’interruzione in genere è diverso. In ogni caso, l’esecuzione è ripetibile ed affidabile, ed il test in cui l’esecuzione della suite si interrompe è sempre lo stesso a parità di release del SUT. I test sono automatizzati a livello di GUI e l’approccio di scripting usato è quello strutturato. Le librerie di test attualmente utilizzate sono quattro.

Sulla base delle sole informazioni fornite, quale tra le seguenti azioni reputate essere la più rilevante da proporre per migliorare la TAS?

- A. Implementare un meccanismo, attivabile automaticamente per certe condizioni, per il ripristino del funzionamento corretto del SUT
- B. Adottare un approccio basato su scripting lineare anziché quello attuale di scripting strutturato, dato che i test della suite sono automatizzati a livello di GUI
- C. Modificare i test automatizzati della suite rimuovendo le istanze di codice ripetuto tra i vari test
- D. Adottare un approccio basato su scripting lineare anziché quello attuale di scripting strutturato, dato che i test della suite sono automatizzati a livello di GUI

Risposta corretta

A

Soluzione

A è corretta perché il problema indicato è dovuto al crash di alcuni servizi/componenti del SUT. Il meccanismo sarà triggerato dal fallimento del test indicato, a fronte di opportune condizioni di errore/eccezione nel SUT relative al crash dei servizi/ componenti. Tale miglioramento è descritto anche in [a] §8.1 (si veda “Scripting”).

B e D sono sbagliate. Infatti, non esiste alcun riferimento rilevante nello scenario che consenta di capire quale sia l’approccio di scripting più adeguato e quindi di valutare eventualmente come migliorarlo. In secondo luogo, il nesso causale (“dato che i test della suite sono automatizzati a livello di GUI”) non ha alcun fondamento.

C è sbagliata. Infatti, sulla base delle solte informazioni fornite. il miglioramento proposto dalla risposta C, presente anche in [a] §8.1 (si veda “Scripting”), non è così rilevante come quello proposto dalla risposta A.

Perciò, A è la sola risposta corretta