Ammin post test 26.5

Sommario

Introduzione

D.

R.

Il test API è un tipo di test software che convalida un'API (Application Programming Interface) per garantire che soddisfi le aspettative in termini di funzionalità, affidabilità, prestazioni e sicurezza. Si concentra principalmente sul livello della logica aziendale e sullo scambio di dati tra i sistemi software, indipendentemente da un'interfaccia utente

Verifica degli URL tra i testi



https://policycentral.cloudapps.cisco.com/cppc/policy-advisor/policies/view-policy/1624

Il Codice di condotta commerciale (COBC) di Cisco riflette il nostro modo di lavorare e di prendere decisioni con integrità. Fornisce inoltre risorse per la risoluzione di problemi complessi, come l'utilizzo responsabile dell'intelligenza artificiale e i conflitti di interessi.

```
function reverseString(str) {
  return str.split("").reverse().join("");
}
```

https://cisco.account.box.com/login

https://cisco.service-

now.com/helpzone?id=sc_cat_item&sys_id=a9860b89dbd9a640cb5772fc0f96191d&u_business_service=

Richiedere assistenza per un problema riscontrato. Verrà creato e gestito un record di evento imprevisto fino alla risoluzione corretta. Verrai inoltre informato dei progressi.

https://www.geeksforgeeks.org/software-testing/software-testing-manual-testing/

Nella tecnica di test Black Box, il tester o l'analista QA controllano solo la funzionalità di un particolare modulo o metodo o talvolta l'intera applicazione fornendo manualmente i diversi test case. In questo caso, il tester fornirà l'input per l'applicazione e lo verificherà manualmente.

Se restituisce l'output previsto, il tester procederà con un altro set di input e segnalerà tutti i risultati al team. Se l'input fornito manualmente dall'utente durante il test ha esito negativo, segnalerà il problema al team di sviluppo.



VIDEO DI PROVA

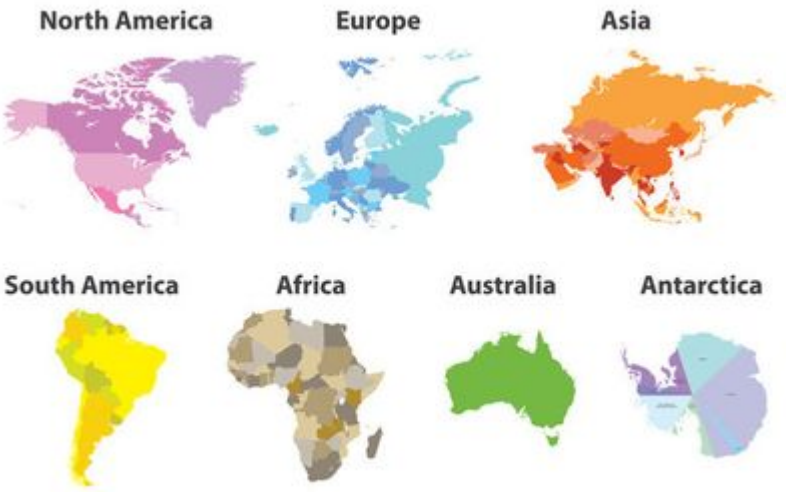
Assegno	Tabella
	Controlla LINK

TABELLA DI PROVA

https://cisco.service-now.com/now/sow/record/incident/507c393193e672502c66ff60ed03d632

Test del white box

In White Box Testing tecnica, la persona controllerà manualmente la struttura interna del sistema come progetti, codifica, ecc. In questo caso, il team di sviluppo esaminerà l'intera parte di codice riga per riga per garantire la correttezza del codice.



CISCO



HERE IS A SAMPLE



Se rileva differenze o errori nel codice, questi correggeranno o correggeranno gli errori nel codice o nelle progettazioni. In questo caso, il processo viene interamente eseguito manualmente e risulta efficiente poiché il codice di controllo o il progetto vengono controllati manualmente dagli esseri umani.

https://en.wikipedia.org/wiki/Manual_testing

https://www.google.com/

La verifica del "ruolo sviluppatore bdb" è stata migrata dall'API ART all'Entra ID in One Access. Quando si richiede l'accesso, selezionare "Metodo di integrazione: memberOf", poiché esistono due diritti con lo stesso nome.

Aspetti chiave del test API ☐

- Comunicazione a livello di messaggio: Anziché utilizzare una GUI, i test API interagiscono direttamente con gli endpoint (URI) dell'applicazione utilizzando vari metodi HTTP (GET, POST, PUT, DELETE) e formati di dati quali JSON o XML.
- Rilevamento Anticipato Difetti: Il test delle API può essere eseguito all'inizio del ciclo di vita dello sviluppo del software, anche prima della creazione dell'interfaccia utente, consentendo ai team di individuare e risolvere i problemi in modo più efficiente e a costi inferiori.
- Attenzione all'automazione: A causa della natura dell'interazione diretta e della struttura coerente, i test API sono particolarmente adatti per l'automazione, che è fondamentale negli ambienti Agile e DevOps moderni per i test continui nelle pipeline CI/CD.
- Copertura completa: Offre una copertura dei test più ampia rispetto ai soli test dell'interfaccia utente, inclusi test di casi limite, gestione degli errori e vulnerabilità della sicurezza che potrebbero essere difficili da accedere tramite l'interfaccia utente.

Tipi di test API ☐
I diversi tipi di test API vengono utilizzati per coprire vari aspetti della qualità di un'applicazione:
☐

Katalon

- Testing funzionale: Verifica che l'API esegua correttamente le operazioni previste, gestendo input, output e codici di stato come specificato.
- Test delle prestazioni: Valuta la velocità, la stabilità e la scalabilità dell'API in varie condizioni di carico (ad esempio, traffico di picco, stress).
- Test di sicurezza: Identifica vulnerabilità quali SQL injection, cross-site scripting (XSS) e autenticazione/autorizzazione interrotta per proteggere i dati sensibili.
- Test di integrazione: Conferma che le diverse parti di un sistema o i servizi esterni con cui l'API interagisce funzionano perfettamente insieme.
- Test del contratto: Assicura che l'API aderisca a un contratto concordato (specifica come OpenAPI/Swagger o WSDL), impedendo modifiche sostanziali tra gli aggiornamenti del servizio.
- Test end-to-end: Convalida gli interi percorsi utente che implicano più chiamate API concatenate.
- Tipi di test API
  I diversi tipi di test API vengono utilizzati per coprire vari aspetti della qualità di un'applicazione:

  Katalon
  - Testing funzionale: Verifica che l'API esegua correttamente le operazioni previste, gestendo input, output e codici di stato come specificato.
  - Test delle prestazioni: Valuta la velocità, la stabilità e la scalabilità dell'API in varie condizioni di carico (ad esempio, traffico di picco, stress).
  - Test di sicurezza: Identifica vulnerabilità quali SQL injection, cross-site scripting (XSS) e autenticazione/autorizzazione interrotta per proteggere i dati sensibili.
  - Test di integrazione: Conferma che le diverse parti di un sistema o i servizi esterni con cui l'API interagisce funzionano perfettamente insieme.
  - Test del contratto: Assicura che l'API aderisca a un contratto concordato (specifica come OpenAPI/Swagger o WSDL), impedendo modifiche sostanziali tra gli aggiornamenti del servizio.
  - Test end-to-end: Convalida gli interi percorsi utente che implicano più chiamate API concatenate.

- ## Come funziona

  Gli strumenti visivi senza codice consentono di creare, estendere e organizzare facilmente i test tra API, interfacce utente Web, database, ESB e anche server MCP comuni nei sistemi basati su AI. Non sono necessarie profonde competenze tecniche. Grazie al supporto di oltre 120 protocolli e formati di messaggio, SOAtest fornisce un framework unificato per la convalida completa della logica aziendale.

  Tramite SOAtest è possibile:

  - Creazione di flussi basati su scenari che simulano le transazioni aziendali reali, consentendo di individuare i bug nascosti generati da sequenze specifiche.
  - Definizione di logica di test, asserzioni complesse, loop e flussi basati su dati con competenze tecniche minime.

- Eseguire singoli test o suite complete e associare controlli di regressione per rilevare immediatamente modifiche impreviste.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Statements</h1>

<p>Multiple statements on one line are allowed.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5; b = 6; c = a + b;

document.getElementById("demo1").innerHTML = c;
</script>

</body>
</html>
```


```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Statements</h1>

<p>Multiple statements on one line are allowed.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5; b = 6; c = a + b;

document.getElementById("demo1").innerHTML = c;
</script>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Statements</h1>

<p>Multiple statements on one line are allowed.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5; b = 6; c = a + b;

document.getElementById("demo1").innerHTML = c;
</script>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Statements</h1>

<p>Multiple statements on one line are allowed.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5; b = 6; c = a + b;

document.getElementById("demo1").innerHTML = c;
</script>

</body>
</html>
```


```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Statements</h1>

<p>Multiple statements on one line are allowed.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5; b = 6; c = a + b;

document.getElementById("demo1").innerHTML = c;
</script>

</body>
</html>
```


Che cos'è il test manuale del software?

Il test manuale è la procedura per verificare il software con l'aiuto delle sue varie caratteristiche e funzionalità. È guidato da una serie preconcetta di test che convalidano il software e fornisce un report finale. Il completamento di questo tipo di test richiede tempo, in quanto viene eseguito completamente attraverso gli sforzi manuali. Pertanto, esiste sempre un ambito di errore umano durante l'esecuzione di questo tipo di test.

Ogni nuovo software viene dapprima testato manualmente prima di adottare l'automazione. La verifica manuale di un software completo richiede più tempo. Una volta che tutte le funzionalità e le caratteristiche del software sono stabili e funzionanti, alcuni dei test case manuali possono essere convertiti in automazione. I test case manuali vengono valutati per primi per verificare se possono essere completamente automatizzati. Questo tipo di test non richiede l'utilizzo di strumenti di automazione per completare l'intero processo.

Annuncio

Caratteristiche dei test manuali del software

Le caratteristiche dei test manuali del software sono elencate di seguito:

- Il test manuale viene eseguito completamente con l'aiuto dell'intervento umano.
- I test esplorativi sono una parte importante dei test manuali. Nei test esplorativi, i tester verificano il software senza alcuna serie predeterminata di test. Rileva i difetti imprevisti e migliora la soddisfazione del cliente.
- La prova manuale è flessibile in quanto consente di modificare i casi di prova in base alle modifiche dei requisiti e ad altre condizioni di prova.
- I test manuali possono essere adottati fin dalle prime fasi del ciclo di vita dello sviluppo del software (SDLC).
- Alcuni dei test case complessi possono essere eseguiti solo manualmente senza alcuna automazione.
- Il test manuale è utile per convalidare l'interfaccia utente del software. Consente di verificare la visualizzazione, i tempi di risposta e la progettazione normale del software.

Perché è necessario eseguire il test manuale del software?

Il test manuale del software è necessario per i motivi elencati di seguito:

- Il test manuale conferma che il software è privo di difetti, funziona correttamente in base ai requisiti ed è sufficientemente stabile da essere implementato in produzione.
- Il test manuale consente ai tester di familiarizzarsi con il software e di comprendere in che modo il software risponde ai clienti. In questo modo è possibile sviluppare test case efficaci.
- Il test manuale identifica e risolve i difetti del software.

Fasi del test manuale del software

Di seguito sono elencate le diverse fasi del test manuale del software:

Fase 1: la prima fase prevede la fase di analisi dei requisiti attraverso l'analisi dei requisiti e delle specifiche, dei documenti, delle guide, ecc.

Fase 2: la seconda fase prevede la creazione di un piano di test che soddisfi tutti i requisiti.

Fase 3 - La terza fase prevede la creazione di test case che coprano ogni requisito.

Fase 4: la quarta fase prevede l'esecuzione dei test case nell'ambiente di test corretto.

Fase 5: la quinta fase prevede l'analisi dei risultati dell'esecuzione dei test e la segnalazione delle discrepanze come difetti.

Passaggio 6 - Il sesto passaggio prevede la correzione dei difetti e il nuovo test. Include inoltre la riesecuzione dei test case non superati.

Tipi di test manuali del software

Di seguito sono elencati i diversi tipi di test manuali del software:

- Black Box Testing: è la tecnica di test in cui il tester non ha alcuna conoscenza del funzionamento interno del software. Si occupa principalmente di verificare se le funzionalità e le caratteristiche funzionano correttamente in base alle esigenze dell'utente.
- White Box Testing: è la procedura di test che include la verifica della struttura interna e del codice sorgente del software.
- Test delle scatole grigie: è la tecnica di test che utilizza sia i principi della scatola nera che le tecniche di test delle scatole bianche.

Strumenti utilizzati per il test manuale del software

Di seguito sono elencati i diversi strumenti utilizzati per il test manuale del software:

- Prova collegamento
- Bugzilla
- Jira
- CaricaCorriere
- JMeter Apache
- Perfecto

Differenze tra il test manuale del software e il test di automazione

Confronto tra test manuali e test di automazione del software:

Testing manuale	Test di automazione
È la procedura per verificare il software con gli sforzi manuali.	È la procedura per verificare il software con l'aiuto degli strumenti di automazione.
Comporta l'esecuzione manuale dei test case.	Prevede l'esecuzione dei test case tramite script di automazione e strumenti.
È meno produttivo e richiede più tempo per il completamento.	È più produttivo e richiede meno tempo per il completamento.
Non garantisce una copertura del 100%.	Garantisce una copertura più ampia rispetto ai test manuali.
Non richiede capacità di programmazione. Può essere eseguito solo con la conoscenza del software.	Richiede capacità di programmazione.

Vantaggi del testing manuale del software

I vantaggi del testing manuale del software sono elencati di seguito:

- Il test manuale consente di verificare gli elementi che cambiano dinamicamente sullo schermo.
- I test manuali sono economici e non si basano su risorse qualificate.
- Le prove manuali possono essere effettuate da collaudatori che non hanno conoscenze di programmazione.
- Il test manuale può essere adottato molto rapidamente ed è adatto per supportare modifiche imprevedibili nel software.

Svantaggi dei test manuali del software

Gli svantaggi dei test manuali del software sono elencati di seguito:

- I test manuali non sono molto affidabili e danno spazio ad errori umani.
- È necessario sviluppare serie separate di test case manuali per moduli diversi, in modo da avere un ambito di riutilizzabilità molto inferiore.
- Il collaudo manuale dipende totalmente dall'esecuzione manuale dei collaudi. Tuttavia, alcune fasi di prova non possono essere eseguite con gli sforzi manuali.
- I tester che eseguono i test manuali devono avere l'esperienza necessaria per utilizzare il software. Inoltre, non vi è alcuna garanzia che tutte le funzioni del software siano state coperte durante l'esecuzione dei test manuali.
- I test manuali sono per lo più un'attività che richiede molto tempo.

Conclusioni

In questo modo si conclude l'esercitazione completa sul testing manuale del software. Dopo aver iniziato con la descrizione di ciò che è il test manuale del software, quali sono le caratteristiche del test manuale del software, perché è necessario il test manuale del software, quali sono i diversi passaggi del test manuale del software, quali sono i diversi tipi di test manuali del software, quali sono i diversi strumenti utilizzati per il test manuale del software, quali sono le differenze tra il manuale del software e il test di automazione, quali sono i vantaggi del test manuale del software e quali sono gli svantaggi del test manuale del software. In questo modo è possibile acquisire una conoscenza approfondita del testing manuale del software. È consigliabile continuare a praticare ciò che si è appreso ed esplorare altri elementi rilevanti per il testing del software per approfondire la comprensione e ampliare gli orizzonti.

Che cos'è Verifica accessibilità?

I test di accessibilità sono un sottoinsieme di test di usabilità in cui gli utenti considerati sono persone con tutte le abilità e disabilità. L'obiettivo di questo test è verificare l'usabilità e l'accessibilità.

L'accessibilità mira a soddisfare le esigenze di persone con capacità diverse, quali:

- Insufficienza visiva
- Insufficienza fisica
- Disturbi uditivi
- Insufficienza cognitiva
- Apprendimento compromesso

Una buona applicazione Web dovrebbe essere adatta a tutte le categorie di persone e NON limitata solo ai disabili. Tra queste:

1. Utenti con infrastrutture di comunicazione inadeguate
2. Persone anziane e nuovi utenti, spesso analfabeti di computer
3. Utenti che utilizzano un sistema obsoleto (NON in grado di eseguire il software più recente)
4. Utenti che utilizzano apparecchiature NON standard
5. Utenti con accesso limitato

Annuncio

Come eseguire il test dell'accessibilità

La Web Accessibility Initiative (WAI) descrive la strategia per le verifiche preliminari e di conformità dei siti Web. La Web Accessibility Initiative (WAI) include un elenco di strumenti software per la valutazione della conformità. Questi strumenti spaziano da problemi specifici come la cecità del colore a strumenti che eseguiranno strumenti di ragnatela automatizzati.

Strumenti di test per l'accessibilità al Web

Prodotto	Fornitore	URL
AccVerify	HiSoftware	http://www.hisoftware.com
Bobby	Watchfire	http://www.watchfire.com
WebXM	Watchfire	http://www.watchfire.com
Scalata	Annulla ordine	http://www.deque.com
InFocus	Tecnologie SSB	http://www.ssbtechnologies.com/

Ruolo degli strumenti automatizzati nei test di accettazione

Gli strumenti di test di accessibilità automatizzati descritti in precedenza sono molto utili per identificare pagine e righe di codice che devono essere controllate manualmente per verificarne l'accessibilità.

1. controllare la sintassi del codice del sito
2. Cerca modelli noti elencati dagli esseri umani
3. identificare le pagine contenenti gli elementi che possono causare problemi
4. identificare alcuni problemi di accessibilità effettivi
5. identificare alcuni potenziali problemi

L'interpretazione dei risultati degli strumenti di test automatizzati di accessibilità richiede esperienza nelle tecniche di accessibilità e una comprensione dei problemi tecnici e di usabilità.

I test vengono eseguiti in modo formale e informale per migliorare la qualità del software. Una

volta completate le verifiche formali, viene eseguita una serie di verifiche informali e arbitrarie. Questo processo è noto come test ad hoc.

Che cos'è il testing ad hoc?

Un test ad hoc è una tecnica di test informale eseguita sul software per individuare i difetti. È condotto in un formato casuale ed è anche noto come test delle scimmie. I test ad hoc non seguono un approccio sistematico e sono privi di casi di test ben documentati.

I test ad hoc non dispongono di documentazione, scenari di test, casi e così via. Gli sviluppatori hanno difficoltà a correggere i difetti rilevati dai test ad hoc a causa della mancanza di questi documenti di test. Inoltre, alcuni bug critici, rari e imprevisti vengono identificati solo effettuando un test informale e casuale del software. Si tratta inoltre di una sorta di test di accettazione che consente di risparmiare tempo nella creazione di nuovi test case.

Un esempio pratico di test ad hoc è il presupposto che un software debba essere consegnato al cliente in un giorno e che il suo sviluppo sia terminato appena un giorno prima, a questo punto non è rimasto tempo per creare ed eseguire i test case in modo che il team di test esegua test ad hoc sull'intero software in base alla conoscenza e all'esperienza generali del prodotto.

Annuncio

Tipi di test ad hoc

Di seguito sono elencati i diversi tipi di test ad hoc:

Test amico

Nei test buddy, vi è il coinvolgimento di almeno due membri durante il processo di test - uno sviluppatore e un tester. Una volta che lo sviluppatore ha completato l'implementazione di un componente, ne esegue il testing di unità. Inserire che il tester invia alcuni dati casuali e arbitrari allo stesso componente ed esamina i risultati. In caso di errori, lo sviluppatore corregge tali difetti.

Prova in coppia

Nei test in coppia, vi è il coinvolgimento di due tester. Uno esegue una verifica informale e casuale del software, mentre l'altro tiene traccia dei risultati dei test. Così entrambi lavorano in coppia e si scambiano idee, conoscenze in modo che il test sia fatto correttamente.

Caratteristiche del testing ad hoc

Di seguito sono elencate le caratteristiche dei test ad hoc:

- Si tratta di un approccio casuale e informale al test.
- Non è supportato da documentazione, scenari di test, casi e così via.
- Viene eseguita dopo il completamento dei test formali.
- Non segue un approccio metodico o strutturato.
- L'esecuzione di test ad hoc richiede meno tempo.
- Vengono rilevati bug nel software in cui non sono disponibili test case.

Quando viene eseguito il test ad hoc?

Il test ad hoc viene eseguito negli scenari elencati di seguito &minu;

- Il tempo disponibile per il test del software è limitato.
- Test formale completato.
- Test case non disponibili.

Quando non viene eseguito il test ad hoc?

I test ad hoc non vengono eseguiti negli scenari elencati di seguito.

- Questa operazione non viene eseguita se vengono rilevati bug eseguendo i test case.
- Al momento del test beta, non viene eseguito.

Vantaggi dei test ad hoc

I vantaggi dei test ad hoc sono elencati di seguito:

- Poiché non aderisce ad alcun processo, è possibile eseguire test ad hoc in qualsiasi momento del ciclo di vita dello sviluppo software.
- Il team addetto ai test può verificare il software e individuare eventuali errori applicando

nuove tecniche di test senza affidarsi solo ai test case.
- Uno sviluppatore può eseguire test ad hoc sullo stesso modulo che sta sviluppando e aumentare la qualità del codice.
- Mentre il processo di test formale richiede molto tempo, i test ad hoc possono essere eseguiti in breve tempo.
- Non richiede alcuna documentazione.

Svantaggi dei test ad hoc

Gli svantaggi dei test ad hoc sono elencati di seguito:

- I test ad hoc devono essere eseguiti da membri del team che abbiano esperienza nel test e una solida conoscenza del prodotto. Qualsiasi membro inesperto del team non può eseguire un test ad hoc.
- Nel caso di un bug, è difficile riprodurre lo stesso risultato in quanto i test ad hoc non sono guidati da alcuna pianificazione.

Procedure ottimali da seguire nei test ad hoc

Di seguito sono elencate le best practice da seguire nei test ad hoc:

- Raccogliere tutte le conoscenze sul prodotto.
- Identificare i componenti del software soggetti a difetti e assegnare loro la priorità.
- Uso di strumenti di prova adeguati.

Conclusioni

In questo modo si conclude l'esercitazione completa sui test ad hoc del software. Dopo aver iniziato con la descrizione dei test ad hoc, quali sono i tipi, le funzionalità, le tecniche, i vantaggi, gli svantaggi, i tempi e le procedure ottimali dei test ad hoc.

In questo modo è possibile acquisire una conoscenza approfondita dei test ad hoc del software. È consigliabile continuare a praticare ciò che si è appreso ed esplorare altri elementi rilevanti per il testing del software per approfondire la comprensione e ampliare gli orizzonti.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).