Test post 26.5 admin

目錄

簡介

問題：

A.

API測試是一種軟體測試，用於驗證應用程式程式設計介面(API)以確保其滿足對功能、可靠性、效能和安全性的期望。它主要關注軟體系統之間的業務邏輯層和資料交換，與使用者介面(UI)無關

這是為了測試文本之間的URL



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

思科的業務行為準則(COBC)反映了我們如何誠信地工作和做出決策。它還提供資源，幫助解決複雜問題，如負責任的人工智慧的使用和利益衝突。

```
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=

請求您遇到問題的幫助。將建立和管理事件記錄，直至成功解決問題。您還將收到有關進度的通知。

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

在黑盒測試技術中，測試人員或QA分析人員僅通過手動提供不同的測試用例，檢查特定模組或特定方法的功能，有時檢查整個應用程式的功能。在這裡，測試人員將給出應用程式的輸入並手動對其進行測試。

如果返回預期輸出，測試人員將進行另一組輸入並將所有結果報告給團隊。如果使用者在測試期間手動提供的輸入失敗，他/她將向開發團隊報告此問題。



測試影片

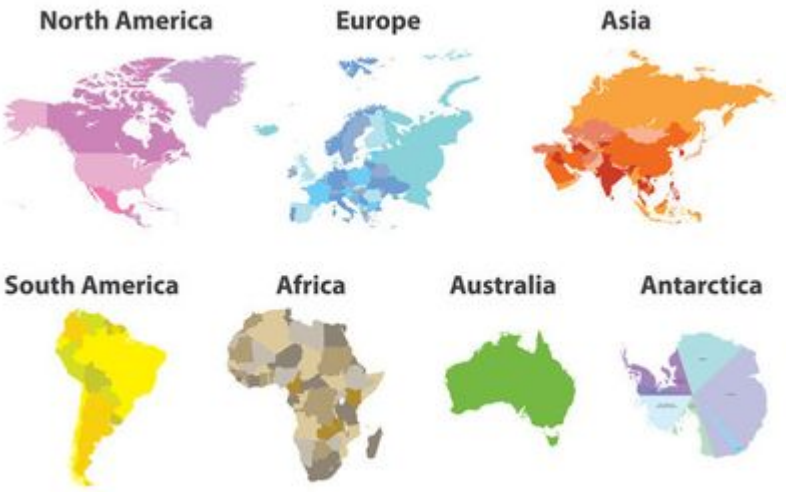
支票	表
	檢查連結

測試表

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

白盒測試

在白盒測試技術中，人員將手動檢查系統的內部結構，如設計、編碼等。在這裡，開發團隊將逐行稽核整個編碼部分以確保代碼的正確性。



CISCO



HERE IS A SAMPLE



如果他/她在代碼中發現任何差異或錯誤，他們將會糾正或修復編碼或設計中的錯誤。在這裡，該過程完全由人工執行，並且由於該檢查代碼或設計由人工手動檢查，該過程是高效的。

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

https://www.google.com/

「bdb developer role」檢查已在One Access中從ART API遷移到Entra ID。請求訪問時，請確保選擇「整合方法：memberOf",因為有兩個權利具有相同的名稱。

API測試的主要方面 ☐

- 消息層通訊:API測試不是使用GUI，而是使用各種HTTP方法(GET、POST、PUT、DELETE)和JSON或XML等資料格式直接與應用程式的端點(URI)互動。
- 早期缺陷檢測:API測試可以在軟體開發生命週期早期執行，甚至可以在構建UI之前執行，使團隊能夠以更低的成本更高效地查詢和修復問題。
- 自動化重點:由於直接互動和一致結構的性質，API測試非常適合自動化，這對於在CI/CD管道中持續測試在現代靈活和DevOps環境中至關重要。
- 全面覆蓋:與單獨的UI測試相比，它提供的測試範圍更廣，包括測試邊緣案例、錯誤處理以及可能難以通過UI訪問的安全漏洞。

API測試的型別 ☐
使用不同型別的API測試來涵蓋應用程式品質的各個方面： ☐

卡塔隆

- 功能測試:驗證API是否正確執行其預期操作，按照指定處理輸入、輸出和狀態代碼。
- 效能測試:評估API在各種負載條件（如峰值流量、壓力）下的速度、穩定性和可擴充性。
- 安全測試:識別漏洞，例如SQL隱碼攻擊、跨站點指令碼(XSS)和損壞的身份驗證/授權，以保護敏感資料。
- 整合測試:確認API與系統或外部服務的不同部分無縫合作。
- 合約測試:確保API遵守約定的合約(如OpenAPI/Swagger或WSDL的規範)，防止服務更新之間的更改中斷。
- 端到端測試:驗證涉及連結在一起的多個API呼叫的整個使用者過程。

- API測試的型別 □
  使用不同型別的API測試來涵蓋應用程式品質的各個方面： □

  卡塔隆
  - 功能測試:驗證API是否正確執行其預期操作，按照指定處理輸入、輸出和狀態代碼。
  - 效能測試:評估API在各種負載條件（如峰值流量、壓力）下的速度、穩定性和可擴充性。
  - 安全測試:識別漏洞，例如SQL隱碼攻擊、跨站點指令碼(XSS)和損壞的身份驗證/授權，以保護敏感資料。
  - 整合測試:確認API與系統或外部服務的不同部分無縫合作。
  - 合約測試:確保API遵守約定的合約(如OpenAPI/Swagger或WSDL的規範)，防止服務更新之間的更改中斷。
  - 端到端測試:驗證涉及連結在一起的多個API呼叫的整個使用者過程。

- 工作方式

  通過可視的無代碼工具，您可以輕鬆建立、擴展和組織跨API、Web UI、資料庫、ESB，甚至是AI輸入系統中常見的MCP伺服器的測試。不需要深厚的技術技能。SOAtest支援120多種協定和消息格式，為您提供了一個統一的框架，用於端到端驗證業務邏輯。

  使用SOAtest，您可以：

  - 建立模擬實際業務交易的基於情景的流，幫助您查詢由特定序列觸發的隱藏錯誤。
  - 以最少的技術專業知識構建測試邏輯、複雜斷言、環路和資料驅動流程。
  - 運行單個測試或完整套件，並附加回歸控制以立即捕獲意外更改。

JavaScript Statements

```
Multiple statements on one line are allowed.
```

JavaScript Statements

Multiple statements on one line are allowed.

JavaScript Statements

Multiple statements on one line are allowed.

JavaScript Statements

Multiple statements on one line are allowed.

JavaScript Statements

```
Multiple statements on one line are allowed.
```

什麼是軟體手動測試？

手動測試是藉助軟體的各種功能和特性來驗證軟體的過程。它以預先設計好的測試集為指導，這些測試集驗證軟體並提供最終結果報告。這種測試需要一定的時間才能完成，因為它是完全通過手動操作完成的。因此，在執行此類測試時，始終存在一定程度的人為錯誤。

每個新軟體在採用自動化之前首先都要經過人工測試。手動驗證整個軟體會花費更多的時間。一旦軟體的所有功能、功能都穩定，工作正常，部分人工測試用例就可以轉化為自動化。首先評估手動測試用例，檢查它們是否可以完全自動化。這種型別的測試不需要使用任何自動化工具來完成整個流程。

廣告

軟體手工測試的特點

下面列出了軟體手動測試的特徵−

- 人工測試完全在人工干預的幫助下進行。
- 探索性測試是手動測試的重要組成部分。在探索性測試中、測試人員無需預先設定任何測試集便可對軟體進行驗證。檢測未預測缺陷，提高客戶滿意度。
- 手動測試是靈活的，因為它允許根據需求的變化和其他測試條件來修改測試案例。
- 手動測試可以從軟體開發生命週期(SDLC)的初始階段開始。
- 有些複雜的測試用例只能手動執行，不需要任何自動化。
- 手動測試對於驗證軟體的使用者介面非常有用。它有助於驗證軟體的顯示、響應和正常設計。

為什麼需要軟體手動測試？

由於下列原因，需要進行軟體手動測試−

- 手動測試可確認軟體沒有任何缺陷，能夠按照要求正常工作，並且穩定性足以在生產環境中部署。
- 通過手動測試，測試人員可以熟悉該軟體，並瞭解該軟體如何與客戶進行響應。這有助於開發有效的測試用例。
- 手動測試可識別並解決軟體中的缺陷。

軟體手動測試步驟

下面列出了軟體手動測試的不同步驟−

步驟1 −第一步涉及通過瀏覽需求和規範文檔、指南等進行需求分析階段。

步驟2 −第二步涉及建立滿足所有要求的測試計畫。

步驟3 −第三步涉及建立涵蓋每項要求的測試案例。

步驟4 −第四步涉及在正確的測試環境中執行測試案例。

步驟5 −第五步涉及分析測試執行結果，並將差異報告為缺陷。

步驟6 −第六步涉及缺陷修復和重新測試。它還包括重新執行失敗的測試用例。

軟體手動測試的型別

下面列出了不同型別的軟體手動測試−

- 黑盒測試−測試技術是指測試人員對軟體內部工作一無所知的測試技術。它主要處理驗證特性和功能是否按照使用者要求正常工作。
- 白盒測試−測試過程包括軟體內部結構的驗證和程式的原始碼。
- 灰盒測試−白盒測試技術是一種既利用黑盒測試原理，又利用白盒測試原理的測試技術。

用於軟體手動測試的工具

下面列出了用於軟體手動測試的不同工具−

- 測試連結
- 布吉拉
- 吉拉
- LoadRunner
- Apache JMeter
- Perfecto

軟體手動測試與自動化測試之間的差異

對軟體手工測試和自動化測試進行了比-

手動測試	自動化測試
該過程需要手動驗證軟體。	該過程是在自動化工具的幫助下對軟體進行驗證。
它涉及手動執行測試用例。	它涉及通過自動化指令碼和工具執行測試案例。
效率較低，需要更多時間才能完成。	它更加有效，完成所需時間更少。
它不能確保百分之百的測試覆蓋率。	它確保了比手動測試更多的測試覆蓋範圍。
它不需要程式設計技能。它只能在瞭解軟體的情況下執行。	它需要程式設計技能。

軟體手動測試的優點

下面列出了軟體手動測試的優-

- 手動測試有助於驗證螢幕上動態變化的元素。
- 人工測試成本低廉，並且不依賴熟練的資源。
- 該手動測試可以由沒有程式設計知識的測試者執行。
- 可以非常快速地採用手動測試，並且適合適應軟體中不可預知的更改。

軟體手動測試的缺點

下面列出了軟體手動測試的缺點-

- 手動測試不是很可靠，為人為錯誤提供了範圍。
- 需要針對不同的模組開發單獨的手動測試用例集，使得可重複使用的範圍非常小。
- 手動測試完全依賴於手動執行的測試。但是，有些測試步驟無法通過手動操作執行。
- 執行手動測試的測試人員應該具有使用軟體的經驗。此外，還不能保證在執行手動測試時，軟體的所有功能都已被覆蓋。
- 手動測試大多非常耗時。

結論

我們對「軟體手動測試」教程所做的全面介紹到此結束。首先介紹了什麼是軟體手動測試，什麼是軟體手動測試的特點，為什麼需要軟體手動測試，軟體手動測試的不同步驟是什麼，軟體手動測試的不同型別是什麼，用於軟體手動測試的不同工具是什麼，軟體手動測試與自動化測試有什麼區別，軟體手動測試有哪些優點，軟體手動測試有哪些缺點。這將讓您深入瞭解「軟體手動測試」。繼續實踐您所學到的知識並探索與軟體測試相關的其他知識，以加深您的瞭解並拓展您的視野，是明智之舉。

什麼是輔助功能測試？

無障礙測試是可用性測試的一個子集，考慮的使用者是擁有各種能力和殘疾的人。此測試的意義在於驗證可用性和可訪問性。

無障礙環境旨在滿足不同能力的人群，例如：

- 視力障礙
- 物理損傷
- 聽覺障礙
- 認知障礙
- 學習障礙

一個好的網路應用應該面向所有群體，而不僅僅侷限於殘疾人。這些包括：

1. 通訊基礎設施較差的使用者
2. 老人和新使用者，他們往往不懂電腦
3. 使用舊系統的使用者（不能運行最新軟體）
4. 使用非標準裝置的使用者
5. 具有受限訪問許可權的使用者

廣告

如何執行輔助功能測試

網站無障礙倡議(WAI)介紹了對網站進行初步和一致性審查的戰略。Web可訪問性計畫(WAI)包括一系列軟體工具，用於幫助進行符合性評估。這些工具包括諸如色盲之類的特定問題，以及執行自動螺旋成形工具的工具。

Web輔助功能測試工具

產品	供應商	URL
AccVerify	HiSoftware	http://www.hisoftware.com
鮑比	觀火	http://www.watchfire.com
WebXM	觀火	http://www.watchfire.com
遞增斜坡	德克	http://www.deque.com
InFocus	SSB技術	http://www.ssbtechnologies.com/

自動化工具在驗收測試中的作用

上述自動輔助功能測試工具非常善於識別需要手動檢查輔助功能的內碼表和行。

1. 檢查站點代碼的語法
2. 搜尋人類列出的已知模式
3. 標識包含可能導致問題的元素的頁面
4. 確定一些實際輔助功能問題
5. 確定一些潛在問題

要解釋自動無障礙環境測試工具的結果，就需要在無障礙環境技術方面具備經驗，同時瞭解技術和可用性問題。

測試採用正式和非正式的方式進行，以提高軟體品質。正式測試完成後，會進行一輪非正式及任意測試。這稱為臨時測試。

什麼是Ad Hoc測試？

即席測試是對軟體進行非正式的缺陷檢測技術。它是以隨機格式進行的，也稱為猴子測試。臨時測試不採用系統化的方法，也沒有任何詳細記錄的測試案例。

即席測試沒有任何文檔、測試場景、案例等。開發人員發現難以修復臨時測試檢測到的缺陷，因為這些測試文檔不存在。此外，某些重大、罕見及意外錯誤僅能通過對軟體進行隨機非正式測試予以識別。它也是一種驗收測試，可以節省建立新測試用例的時間。

一個即席測試的實際例子是假定一個軟體需要在一天內發往客戶端，並且其開發在一天前完成，此時沒有時間建立和執行測試用例，因此測試團隊基於總體產品知識和經驗對整個軟體進行即席測試。

廣告

臨時測試的型別

特定測試的不同型別列示如下−

夥伴測試

在夥伴測試中，在測試過程中至少有兩名成員參與 — 一名開發人員和一名測試人員。一旦開發人員完成了一個元件的實現，他就對該元件進行單元測試。在測試員將一些隨機的任意資料提供給同一元件後檢查結果。如果出現任何錯誤，開發人員會修正這些缺陷。

配對測試

在配對測試中，有兩個測試者參與。其中一組對軟體進行非正式、隨機的驗證，另外一組對測試結果進行記錄。因此，他們兩人成對工作，交流思想、知識，以便正確完成測試。

Ad Hoc測試的特點

即席測試的功能如下−

- 這是一種隨意而非正式的測試方式。
- 任何文檔、測試方案、案例等都不支援。
- 它是在正式測試完成後執行的。
- 它沒有遵循有條不紊或結構化的方法。
- 執行專門測試所需的時間更少。
- 偵測軟體上的錯誤（沒有可用的測試案例）。

臨時測試何時完成？

即席測試在下面列出的方案中完成&minu;

- 可用於測試軟體的時間有限。
- 正式測試已完成。
- 測試用例不可用。

何時未完成即席測試？

在下面列出的場景中不會執行即席測試−

- 如果透過執行測試用例偵測到錯誤，則不會執行偵錯。
- 在beta測試時，不會執行該操作。

Ad Hoc測試的優點

即席測試的優勢列示如下−

- 它不遵循任何流程，因此可在軟體開發生命週期中的任何時間點執行臨時測試。
- 測試團隊可以應用新的測試技術來驗證軟體並發現錯誤，而不只是依賴測試用例。
- 開發人員可以在他正在開發的同一模組上執行臨時測試，並提高其代碼品質。
- 儘管正式測試過程需要大量的時間，但臨時測試可以在短時間內執行。
- 它不需要任何文檔。

Ad Hoc測試的缺點

臨時測試的缺點列示如下−

- 專門測試必須由團隊成員來執行，這些團隊成員必須具備測試經驗和對產品的豐富知識。團隊中任何沒有經驗的成員均無法執行臨時測試。
- 發生錯誤時，因為隨機測試並非由任何規劃所驅動，所以難以複製相同。

臨時測試中應遵循的最佳實踐

在特殊測試中要遵循的最佳實踐列示如下−

- 收集該產品的所有知識。
- 確定軟體容易出現缺陷的元件並排定其優先順序。
- 使用合適的測試工具。

結論

我們對「軟體Ad Hoc測試」教程的全面介紹到此結束。我們首先描述什麼是臨時測試，什麼是臨時測試的型別、特徵、技術、優勢、劣勢、時間和最佳實踐。

這將讓您深入瞭解軟體Ad Hoc測試。繼續實踐您所學到的知識並探索與軟體測試相關的其他知識，以加深您的瞭解並拓展您的視野，是明智之舉。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。