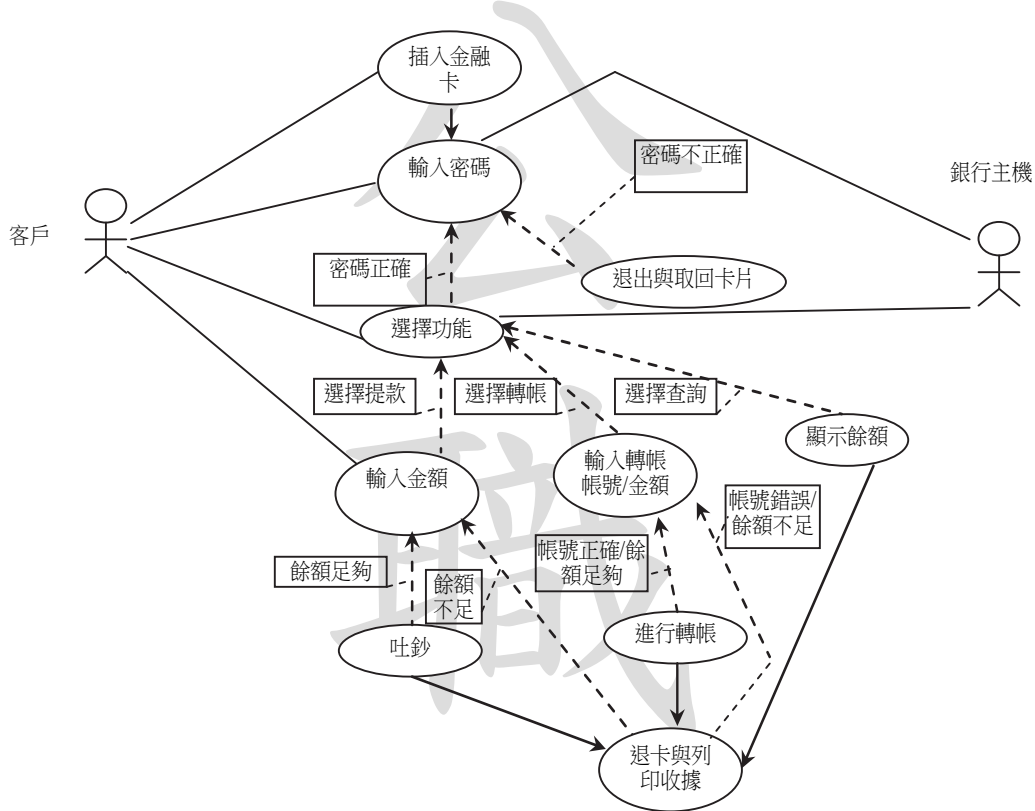


106 年特種考試地方政府公務人員考試試題

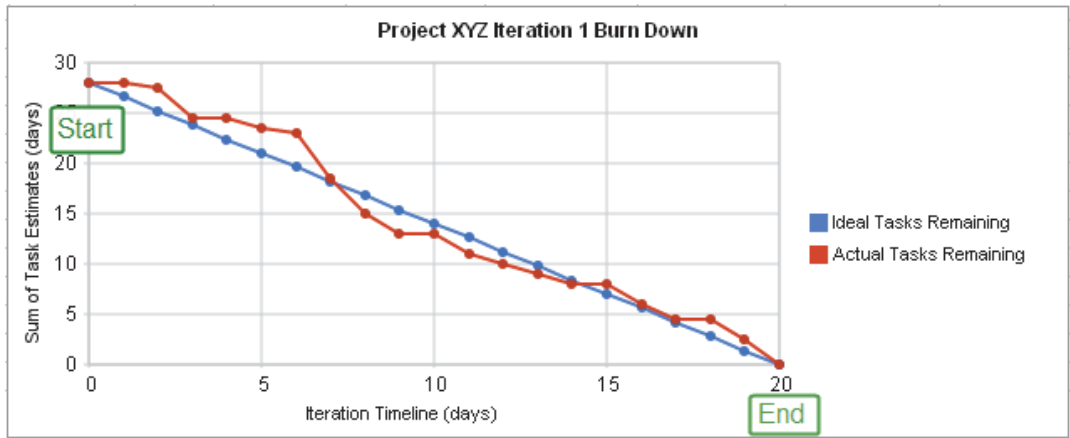
等 別：三等考試
類 科：資訊處理
科 目：系統分析與設計

一、請從自動提款機的觀點看提款、查詢餘額和轉帳等相關流程，試述三個代表這三項功能的使用案例，並建立自動提款機系統的使用案例圖。參與者有顧客以及銀行主機。[106地特]6
擬答：



二、請舉一個Burn-Down Chart (燃盡圖) 的例子並詳述之，其作用為何？[106地特]2
擬答：

- (一)燃盡圖 (burn down chart) 是用於體現剩餘工作量的工作圖表，由橫軸 (X) 和縱軸 (Y) 組成，橫軸表示時間，縱軸表示工作量。這種圖表可以直觀的預測何時工作將全部完成，常用於軟體開發中的敏捷軟體開發方式，也可以用於其他類型的工作流程監控。
- (二)如下圖所示，可在燃盡圖中繪製兩條線，一條表示期望的工作進度，另一條記錄實際的工作進度，依據工作切割的工作要點，完成一個就減去一個，以此來衡量工作距離全部完成的剩餘時間。當實際工作曲線低於期望值時，則表示工作可能提前完成，相反的情況則可能會延期。如果每次繪製的圖標，實際進度曲線都在期望值下方，則表示計劃做的過於保守，可以適當縮短；相反的情況則表示計劃過於激進，應當適當延長。也可以通過多次的記錄統計，了解工作團隊的工作效率是否有一定的提升，找出提高效率的辦法。



(ref: <https://zh.wikipedia.org/wiki/%E7%87%83%E5%B0%BD%E5%9B%BE>)

三、請寫出user story的樣板，並舉一個範例詳述之。[106地特]3

擬答：

(一)User Story是指在軟體開發和專案管理中用日常語言或商務用語寫成的句子。這個句子反應終端用戶或系統用戶捕捉到的關於一個用戶在其工作職責的範圍內做的或需要做的事務。在敏捷開發方法中用來定義系統需要提供的功能和實現需求管理。以客戶或使用者的觀點撰寫下有價值的功能（functionality/feature），代表（represent）客戶的需求，不包含在開發時才會確定的實做細節。

(二)User story最常見的樣板如下：

"As a <role>, I want <capability> so that <receive benefit>"，也就是

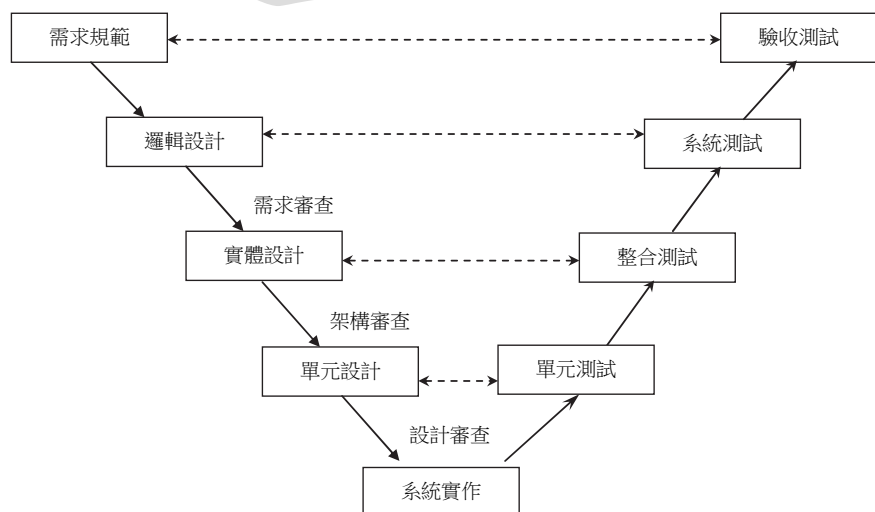
身為一個{{特定角色}}，我希望能有{{特定功能}}以能讓我{{得到某種價值}}

例如：身為一個{{學生}}，我希望能有{{只有我能查到自己的成績}}，這樣才能{{知道我的分數又不用讓其他人知道}}

四、請說明單元測試、整合測試、系統測試與使用者驗收測試主要是根據那些個別對應的程序或文件來擬定測試計畫。[106地特]5

擬答：

如下圖之軟體測試V模型，左半部表示軟體開發生命週期各階段的靜態分析，右半部則表示動態測試的各階段，因此動態測試的各階段即以對應的系統發展階段所得之文件來擬定測試計畫，分述如下：



(一)單元測試 (unit testing)

由軟體開發人員親自對軟體內部結構執行最小規模的測試。受測程式可以是單獨或僅提供某項功能的程式。此類測試的目的在確保其功能可單獨且正確地運作，每個單元必須能獨立測試而不受其他元件的影響，是屬於所有測試策略中最低層次的測試。主要使用白箱測試方法，並配合黑箱測試方法，來設計測試案例，以檢驗每個單獨的模組是否正確的執行其預期功能。

(二)整合測試 (integration testing)

是指在元件通過單元測試之後，開始整合單元模組，以檢查元件之間的一致性。若僅部分元件完成單元測試，其餘尚未發展完畢，則可先整合已完成單元測試的元件並進行整合測試，此時開發人員亦應加入整合測試工作。整合測試的目的在於找出系統整合時的錯誤。雖然每個單元模組都通過個別的單元測試，但組合後的各模組未必能順利執行，因此須檢驗此組合過程與系統架構是否正確，通常用黑箱測試進行此種檢驗。最佳的整合測試是先使用由上而下的整合測試，並在確定較具關鍵性的綜合模組後，再進行由下而上的整合測試。

(三)系統測試 (system testing)

是整個資訊系統的測試(System Testing)，檢驗系統中各子系統之間的整合，與整個系統軟硬體功能及執行績效是否符合需求，主要是由硬體工程師與軟體測試人員進行系統測試。測試環境應儘量與實際環境相似，而測試案例的設計方法以黑箱測試技術為主，並參考系統的需求、規格與效能要求，來進行設計。此階段軟體測試工程師的任務為：1. 分析軟體內部介面問題、參與系統測試規劃與設計，以確保充分測試軟體；2. 以模擬資料測試軟體系統介面的錯誤，並設計錯誤處理路徑，測試來自其他硬體系統或外部軟體的錯誤資訊。

(四)驗收測試 (Acceptance Testing)

透過軟體系統操作者來進行測試，並使用實際完成的軟體系統為測試標的，與實際系統的操作數據來進行測試，而非使用模擬的測試資料與系統。根據需求分析規格中所定義的驗收準則來檢驗軟體的功能與效能是否滿足顧客的需求，一般採用黑箱測試方法進行。驗收測試根據其測試的方式與測試環境的不同，又可以進一步細分為以下兩種測試。1. 阿法(α)測試：由軟體開發人員指導使用者在系統開發所在地進行操作，開發者可同時記錄錯誤和使用時所產生的問題。2. 貝塔(β)測試：在終端用戶處由使用者自行進行測試，開發者並不在現場，使用者須自行記錄問題並回復給開發者。

