

## 軟體產業與軟體工程

軟體不會受到實體的限制，但很容易變得極度複雜。如蝦皮、報稅軟體、電腦遊戲的軟體做法完全不同，但是都需要軟體工程，而不是同一種工程技術。通常是由團隊合作而非個人來開發，而且在軟體的營運周期中會持續進行維護和優化並創新。

專業的軟體是設計給其他人而不是給開發它的人使用的，必須支援專業軟體的開發，而不是個人的程式設計。為了貼近使用者，軟體所具備的需求可分為三大項-角色-需求、技術-創新、流程-保質，透過團隊合作並了解使用者的需求，開發新的軟體，並確保流程中的使用者需求分析是產品設計的最初端，在設計的過程裡面需要被考量，包括程式本身、讓程式正常運作所需的設定資料。必須要充分瞭解人與系統互動的模式，於創新與優化方面讓使用者有更好的體驗。

## 軟體工程流程與團隊

軟體工程比較正式的說法：將系統化的、規範的、可度量的方法用於軟體的開發、行和維護，即將「工程化」應用於軟體開發中。符合使用者的需求來進行開發，較簡單的說法為軟體工程是提供「將真實世界的需求轉換成 軟體世界的軟體」的各種做法。

軟體開發通常需進行的需求分析了解開發的重點，客戶要什麼站在使用者的角度去設計系統應該要具備的功能，再進一步設計轉換為系統架構，根據設計結構實現，透過程式碼來實作的同時一同測試產品是否符合最初的需求，功能是否正確。而軟體開發可能參與的角色，系統分析師(SA) 發展軟體需求；程式設計師(Programmer, PG)負責撰寫程式；測試工程師(Software Tester, ST) 負責測試軟體是否存在錯誤；UX 設計師(User Experience Designer)主要關注產品的使用者感受；UI 設計師(User Interface Designer)關注 產品的使用者介面佈局(Layout)方式；前端工程師(Front-End Engineer, F2E), 負責選擇、開發和測試網站的使用者面；FAE (Field Application Engineer)負責軟體產品的售前(pre-sale)和售後(postsale)支援工作。

## 敏捷開發方法-Scrum

敏捷宣言(Agile Manifesto, Utah, USA, 2001) 的核心概念為個人與互動>流程與工具 • 可用的軟體>詳盡的文件；與客戶合作>合約協商；回應變化>遵循計劃，敏捷的概念為共同合作產出實際的成果，評估是否有符合目標，並依此去做改善，較實務的方法為根據歷程來學習、為改變(需求、環境)而規劃、擁抱改變 • 觀察並調適，根據 How 和 What 兩個面向， 可分為 Simple 、

Complicated、Complex、Chaotic 四種複雜程度，並以若以迭代 (Iterative) 的方式、遞增式 (Incremental) 地進行開發，較容易逐步實現需求願景，Scrum 參與流程的三種主要角色)：Product Owner (產品擁有者)：願景的產品特性，包括功能與需求；Development Team (開發團隊)：根據需求進行專案開發；Scrum Master：協助團隊自主發揮功能、為團隊排除障礙。Scrum 產出可分為潛在可發布增量 (Potentially Releasable Increment, PRI)、工作待辦清單 (Sprint Backlog) 與產品待辦清單 (Product Backlog)：Product Backlog Item (PBI) 以使用者故事呈現，產品需求特性、價值預估、時程預估、優先序。而工作待辦清單 (Sprint Backlog) 會將 PBI 再分解成工作任務。潛在可發布增量 (Potentially Releasable Increment, PRI) 在 Sprint 所增加的軟體功能，是可以隨時交付給客戶的軟體版本。Scrum 需開的四種會議分為 1. Sprint Planning (衝刺規劃會議) 2. Daily Scrum 3. :Sprint Review 4. Sprint Retrospective，Scrum 角色分為: Product Owner、Development Team (開發團隊)、Scrum Master。

## 程式碼版本控制簡介

Git 主要是拿來做軟體的版本控制與維護。在開發軟體時常常會有需多不同的版本，而這些都需要備份，相同內容只會有一份記錄。在 git 中 git 會保留對於檔案的新增、修改或是刪除等操作的歷史紀錄，它也會記錄是誰在什麼時間對檔案做更動。

常用基本指令	
<b>git init：</b>	創建一個 git 資料夾 (初始化)，裡面有預先配置好的東西
<b>rm -r .git：</b>	刪除 git
<b>rm -rf .git：</b>	刪除 git 包含裡面的檔案
<b>git status：</b>	顯示目前 git 狀態

是否加入控制版本	
<b>untracked：</b>	沒有加入版本控制的檔案
<b>staged：</b>	要加入版本控制的檔案
<b>git add code.js：</b>	將 code.js 加入版本控制
<b>git add .：</b>	將目前資料夾底下所有的檔案都加入版本控制
<b>git rm --cached code.js：</b>	將 code.js 移回 untracked 狀態

歷史紀錄	
<b>git log：</b>	顯示詳細歷史紀錄、commit ID、作者、建立時間、信息
<b>git log -- oneline：</b>	顯示簡易歷史紀錄、commit ID 前七碼、信息

## 軟體測試簡介

軟體測試類型可分為四個象限，系統、整合、單元測試偏向在工程師導向，以自動化來進行；功能驗收測試以使用者從使用的狀態測試，確保使用者在使用是不會有問題的；非功能性驗收測試，可分為壓力測試、跨平台測試、安全測試，系統有沒有辦法正確的運行、讓使用者操作、使用上是安全的；探索性測試指開發出一個軟體，並找出系統中的 BUG，以人下去操作，如跨平台測試使用者的作業系統不一樣，裝置也不同是否可以依照這些不同軟體仍然可以成功的在不同的介面運行，依據先前的經驗找到漏洞，測試人員不斷的探索系統的問題，在更進一步地進行測試，手動操作是仰賴於人的經驗、之前遇到過的案例來進行多面向的測試很難依賴電腦進行自動化的測試，找出問題。

技術導向