軟體工程概念與文件撰寫簡介

July 2022

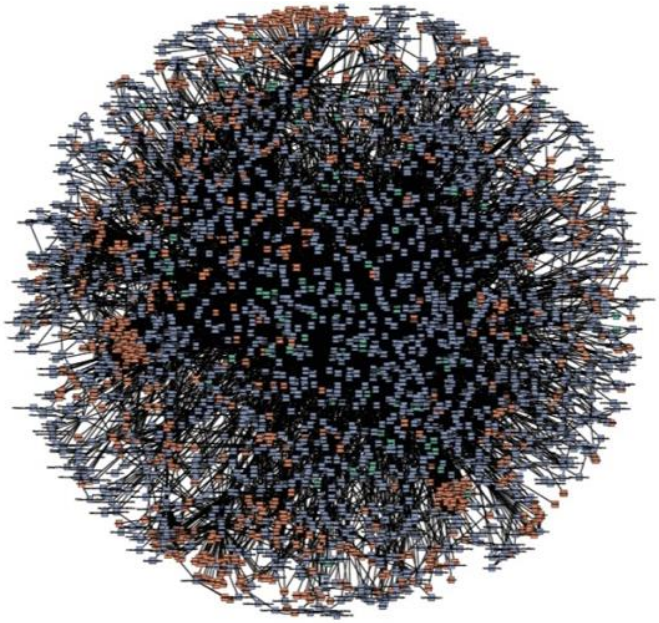
范姜永益

輔仁大學資訊工程學系教授
輔仁大學資訊中心主任
yyfanj@csie.fju.edu.tw

軟體工程概念

軟體的本質問題:
複雜性(Complexity)

軟體系統之複雜程度，往往隨著程式的大小及軟體元件個數以非線性的方式，甚至是等比級數的方式遞增。

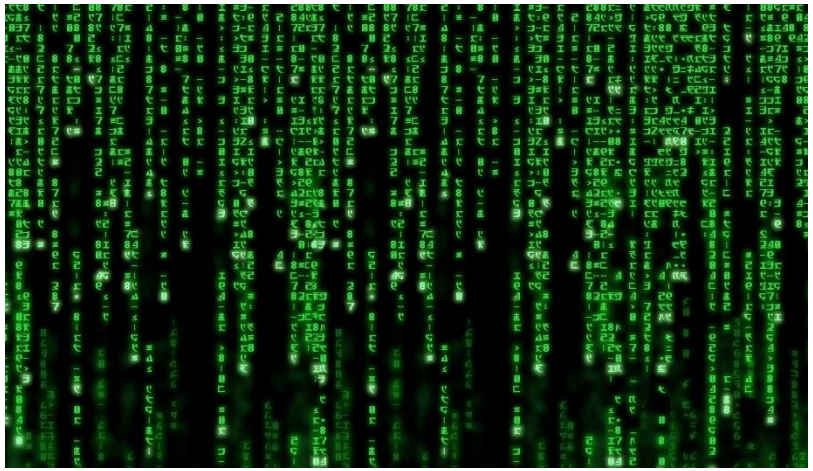


軟體的本質問題:
一致性(Conformity)

在大型的協作環境下發展軟體系統，介面跟介面間、模組跟模組間、系統跟系統間的介接，便都存有一致性的問題需要解決。

軟體的本質問題:
隱藏性(Invisibility)

軟體本身是看不到、摸不著的，導致需求容易存有誤解、疏忽的地方不容易被發現，大大地妨礙了彼此的溝通。



軟體的本質問題:
易變性(Changeability)

為了滿足客戶的需求，一套成功的軟體系統，從開發到完成、從產品交付到營運維護，隨時都有可能要做變更。

真實案例1

- 電腦錯亂存戶A三千多萬，一銀只討回二十萬 (2003)
  - 男子九十二年七月到第一商銀繳交增資卡循環信用貸款利息二千六百元，因櫃台疏失在電腦上多打了一個「1」，造成電腦程式錯亂。
  - 他在兩個月內於自己帳戶提領378次、共三千三百六十三萬餘元，被告等八人雖遭判刑，但銀行只討回廿餘萬元。

https://www.ptt.cc/bbs/Bank_Service/M.1183195444.A.9C7.html

真實案例2

- 廣東銀行出錯多發工資，三百人爭相提款 (2008)
  - 銀行第一次轉賬時，電腦提示交易失敗，之後銀行又轉帳一次，但其實第一次轉賬是成功的。
  - 由於銀行出錯，該廠超過八成員工多發了雙倍甚至四倍工資，共有三百多名員工多發工資。

http://www.epochtimes.com/b5/8/2/28/n2026834.htm

真實案例3

- Instagram大當機無法刷新！ 全球網友崩潰哀嚎(2019.01)
  - https://udn.com/news/story/7086/3620234
- 全台ATM、台彩大當機 中信銀找到原因：非駭客 (2018.10)
  - https://www.chinatimes.com/realtimenews/20181019004717-260410
- YouTube全球大當機！網頁一開變「天書」網友哭：世界末日 (2018.10)
  - https://m.ctee.com.tw/livenews/kj/20181017001515-260412
- 又出包！臉書坦言軟體錯誤 千萬用戶私人貼文被迫公開
  - https://newtalk.tw/news/view/2018-06-08/127146

2019年2月 Ptt 網友的案例 http://bit.ly/2H7yAoa

Project:親戚(親戚的朋友)委託建置
一套系統(含前後端)

前情提要: (1) 系統規模要求比當初談的更大更多
(2) 建置時程2個月
(3) 沒簽約、同時有另一個團隊競爭
(4) 三位同學一起建置 (為求時效)
(5)能力程度是 **同學1**=>**同學2**>>**原PO**

2019年2月 Ptt 網友的案例

過程

(1)每兩周與員工開小會、
月底與親戚和老闆開大會並demo

(2)案子分割好大家各做各的最後整合

(3)每次的小會，**同學1**不出席。
轉述一下開會內容他就能開發了
轉述他都表示講得不好聽不懂
詢問進度他都說目前這樣不用擔心

問:按鈕的功能能不能先開發他都表示他有他的規劃

2019年2月 Ptt 網友的案例

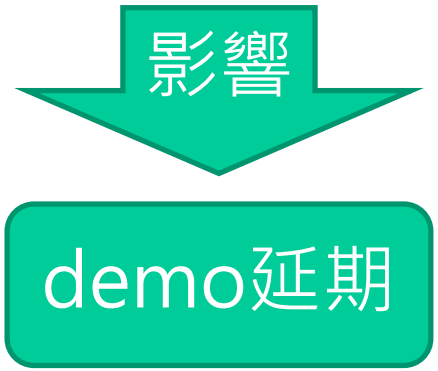
問題出在哪裡?如何改善?

團隊缺乏強制規範與紀律(discipline)

2019年2月 Ptt 網友的案例

過程

(4) 時間很快地來到月底demo前幾天
同學1更改/刪除DB裡的每個table欄位沒有通知所有人
導致同學2和**原PO**所做的系統整個崩潰
也無法在demo前修正完



2019年2月 Ptt 網友的案例

為何自行更動DB裡的table欄位?

「這樣改我才可以開發而且整個系統會變得很有效率

也說有把更改的內容寫成文件幫助我們了解更改哪裡」

2019年2月 Ptt 網友的案例

問題出在哪裡?如何改善?

團隊缺乏強制規範與紀律(discipline)

版本控制(各種更動都要記錄，並能讓所有人知道)

2019年2月 Ptt 網友的案例

過程

(5) 經過上次的經驗，**同學1**終於來開會

「怎麼我的工作量變大了而且你轉述的差好多」

「你做的好爛喔我直接砍掉、我做系統都是全部想好才會開始動工、而且我可以全部一個人來」

「跟**原PO**反映說不要一直催他，他做得出來」

「2個禮拜的例會先不要看demo，因為有作業及報告」

2019年2月 Ptt 網友的案例

過程

(6) 時間又來到第二次demo前一天

「**同學1**反映壓力很大做不完、希望**原PO**接手」

「**原PO**接手發現進度停留在第一次demo的狀態」

影響

Demo再次延期

2019年2月 Ptt 網友的案例

問題出在哪裡?如何改善?

團隊缺乏強制規範與紀律(discipline)

版本控制(各種更動都要記錄，並能讓所有人知道)

系統開發透明度(Visibility)、工程化方法

2019年2月 Ptt 網友的案例

結果

公司的人來表示跟我們同時間開發的競爭團隊已經做完了

影響

原PO沒有拿到案子

2019年2月 Ptt 網友的案例

問題出在哪裡?如何改善?

團隊缺乏強制規範與紀律(discipline)

版本控制(各種更動都要記錄，並能讓所有人知道)

系統開發透明度(Visibility)、工程化方法

建立團隊的重要性、專案管理、專案負責人、團隊溝通模式

風險管理 (系統風險、團隊風險、專案風險)

我們需要一些**方法**和**工具**來協助我們！

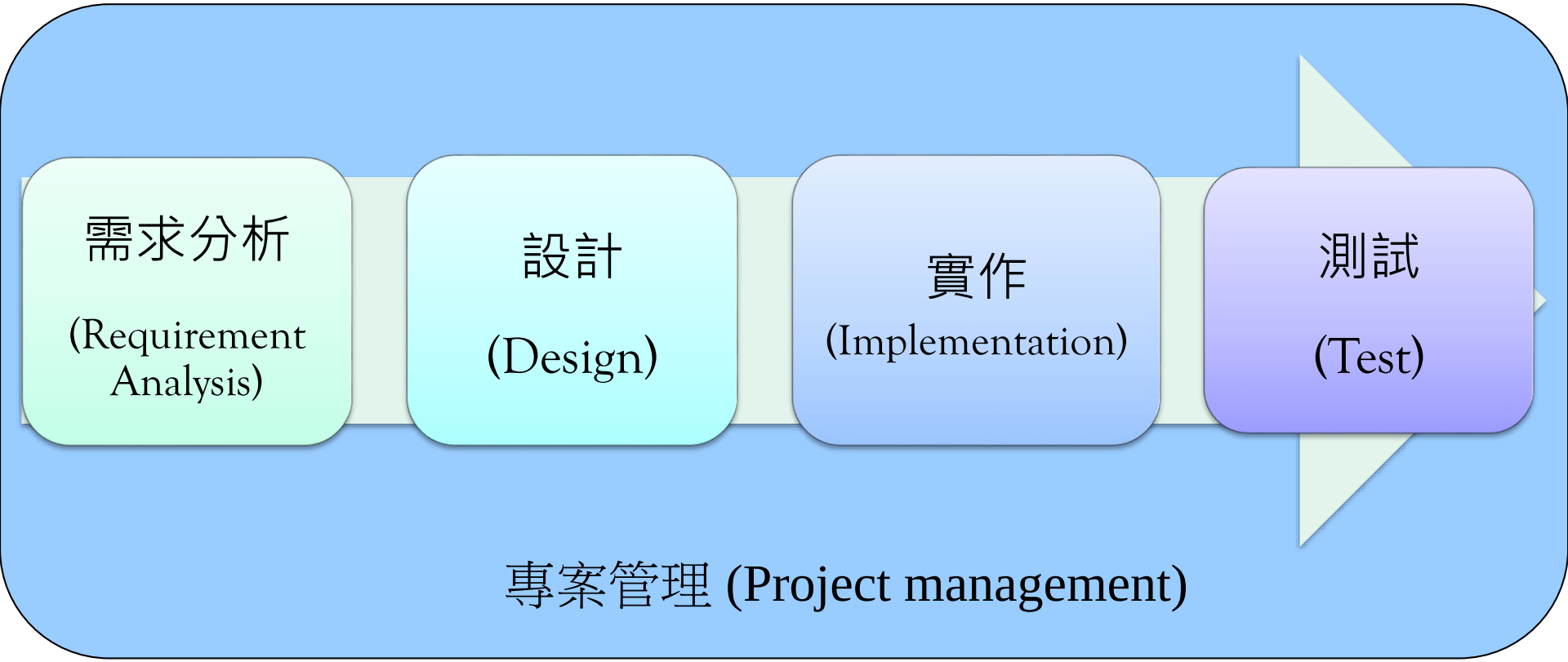
什麼是軟體工程?

「將**系統化的**、**規範的**、**可度量的**方法用於軟體的開發、執行和維護的過程，即將“工程化”應用於軟體開發中。」

---from Wiki

比較簡單的說法：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。

典型軟體開發流程



需求分析(Requirement Analysis)

- 了解客戶的需求、分析系統的可行性、分析需求的一致性，以及正確性等。
- 重點是"**What**"。
- 通常會撰寫**需求文件 (SRS)**。

你可以先訂出使用案例

- 透過**情境(scenario)**思考，站在使用者操作系統的角度，思考系統該具備怎樣的功能，進而引領需求的分析。
- 先把**故事(Scenario)**說出來！

接著訂出功能需求

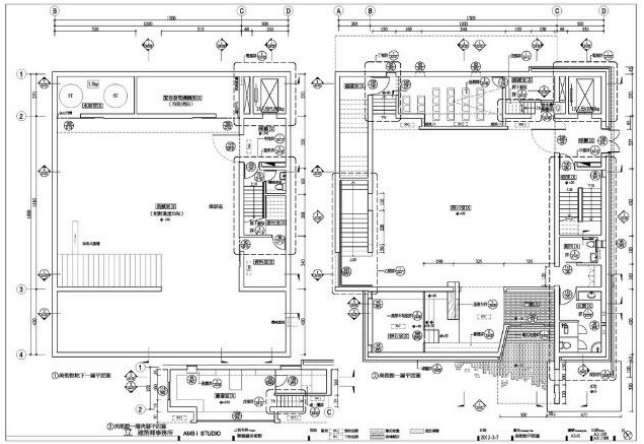
- 具體提出系統應該提供的服務項目。
- 系統是否具備這些功能需求是非常明確的。

再訂出非功能需求

- 強調對於系統品質的要求與限制，或者是說系統應該具備的特性，例如，**可靠度**、**安全性**等品質性的要求。

設計(Design)

- 將需求轉換為系統的重要過程。
- 包含架構設計、模組間的介面設計、資料庫設計、演算法設計與資料結構設計等。
- 重點是"**How**"。
- 有時會考慮日後的"Change"。
- 通常會撰寫**設計文件 (SDD)**。

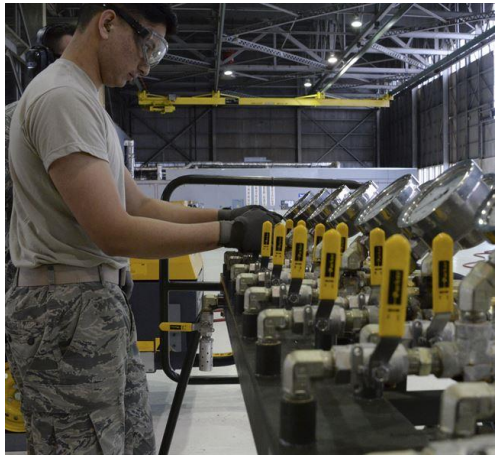


實作(Implementation)

- 透過程式語言將所設計的內容轉化為可執行的軟體。
- 是一般人認為軟體工程師唯一的工作。

測試(Test)

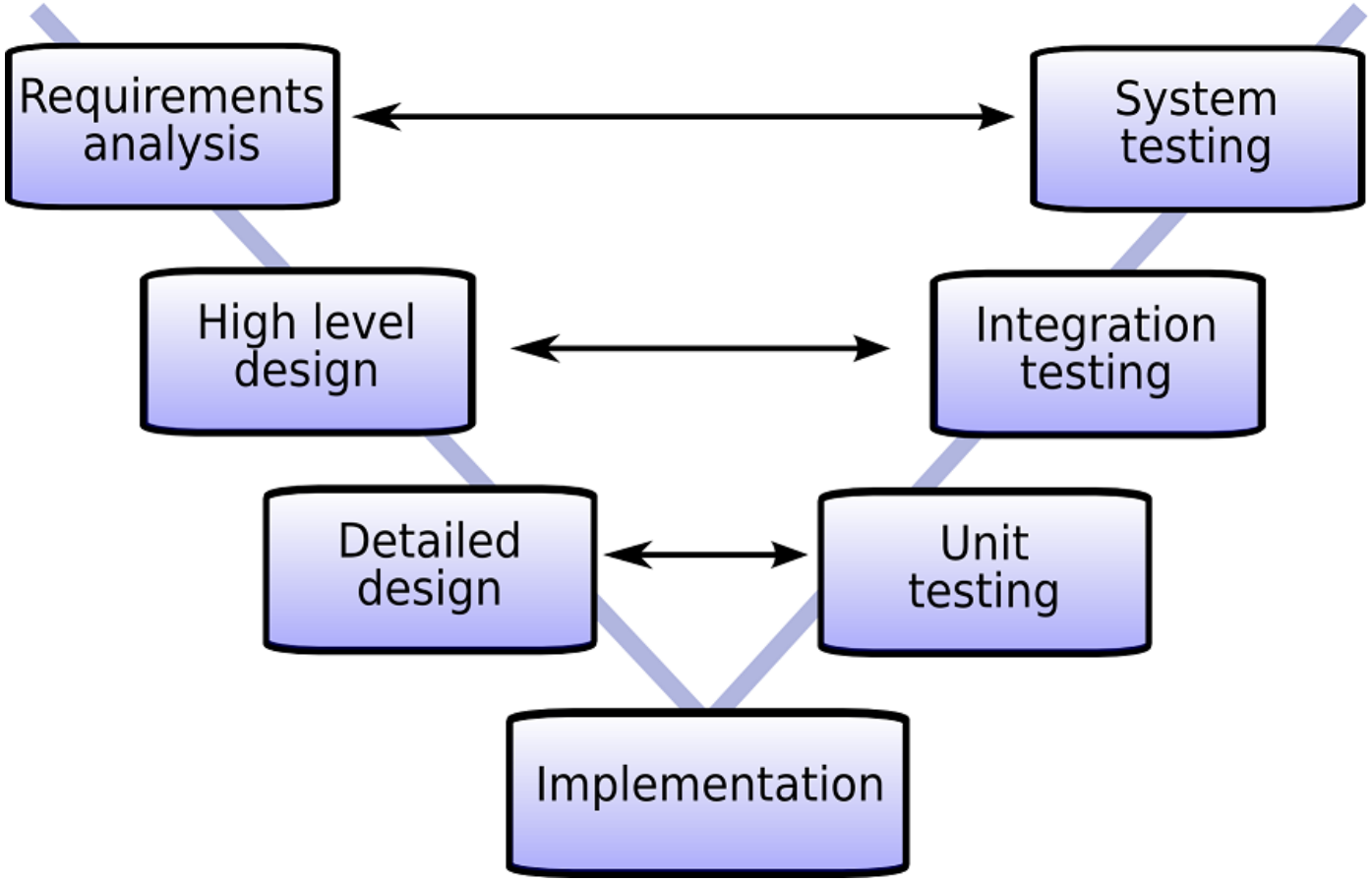
- 測試是對實作活動階段所產出的程式碼模組進行檢測，以檢驗其功能是否正確、效能是否符合要求等。
- **測試案例(Test Case)**的設計是測試流程的重點。



測試可以做到

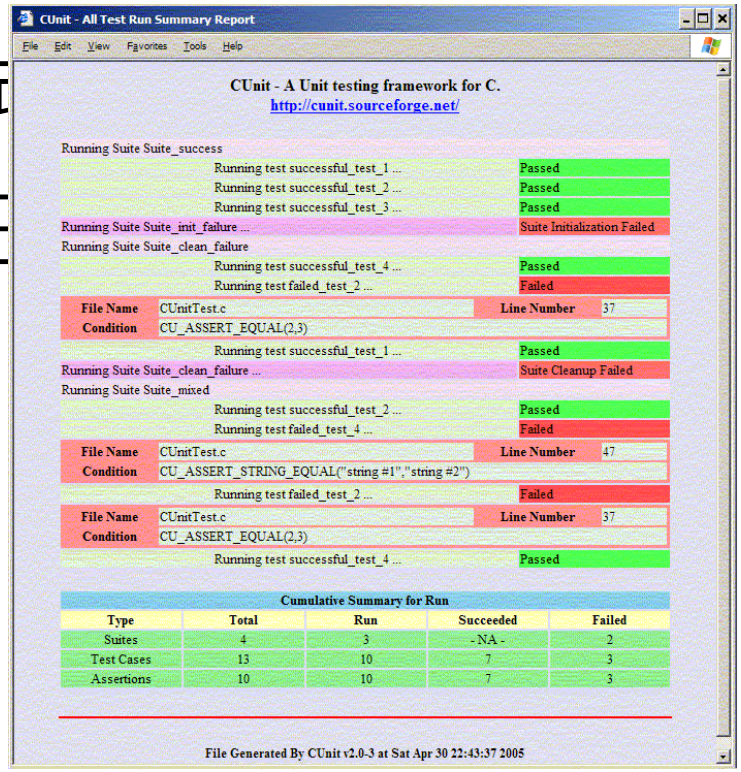
- 發現程式的錯誤(bug)
- 評估程式的品質(quality)
  - 與需求或規格一致(conformance to requirements)
  - 符合使用的目的(fitness for purpose)
  - 功能、效能、安全性、使用者經驗

軟體測試V Model



單元測試

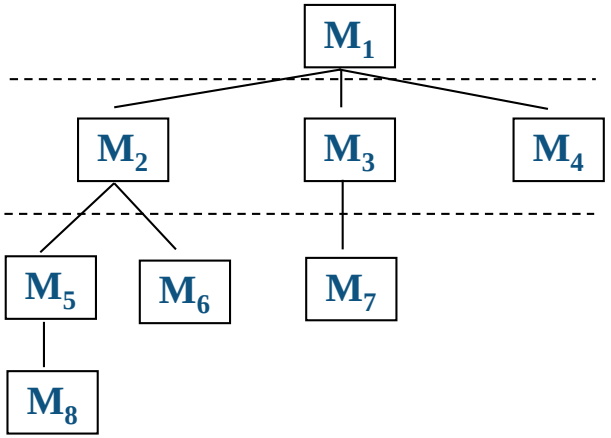
- 以程式中最小的邏輯單元為標的，撰寫測試程式，來驗證程式
- 忽略單元測試可能造成日的bug。

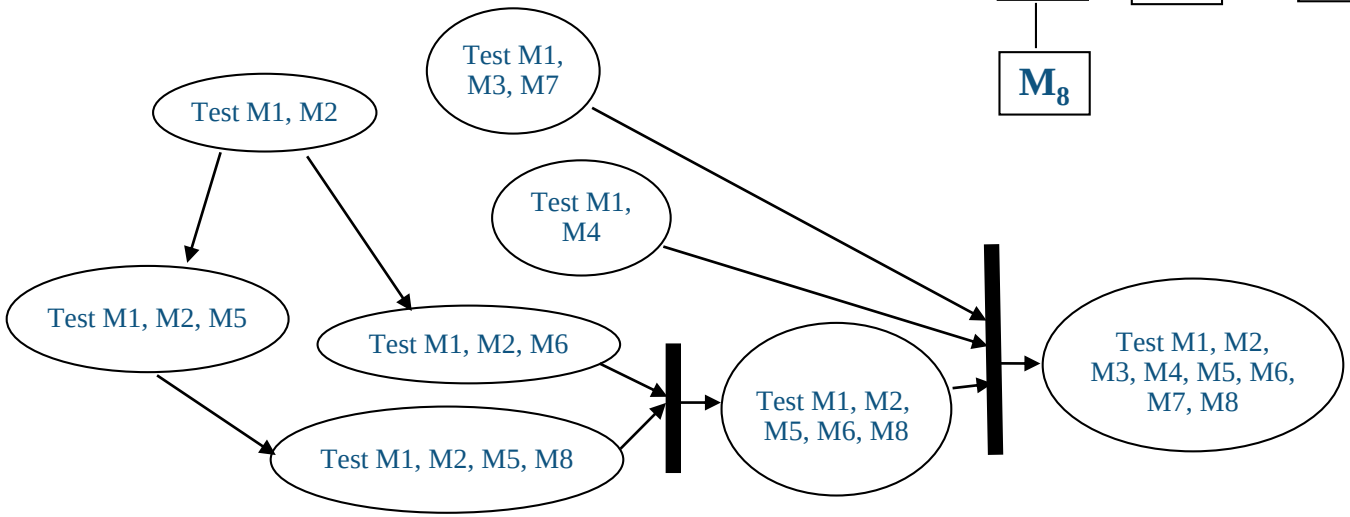



```
1  @Test
2  public void simpleAdd() {
3      Money m12CHF= new Money(12, "NTD");
4      Money m14CHF= new Money(14, "NTD");
5      Money expected= new Money(26, "NTD");
6      Money result= m12CHF.add(m14CHF);
7      assertTrue(expected.equals(result));
8  }
```

整合測試

- 應在整合過程中就逐步進行測試。





SELAB
FU JEN CATHOLIC UNIVERSITY

系統測試

- 是對整個系統的測試，將軟體與硬體看作一個整體，檢驗它是否有不符合系統需求之處。
- 包含**功能測試**、**安全測試**、**壓力測試**等。
- **通常由獨立的測試人員測試**，而非開發者自行測試。

測試工具

- 你可以運用測試工具來加速測試的進行:
  - Selenium (SideeX)
  - Robot Framework

- Top 10 Automation Testing Tools: https://dzone.com/articles/best-automation-testing-tools-for-2018

專案與軟體專案

- 「專案(project)」是指：「針對某一項獨特與唯一的任務，採取一系列的行動來完成既定目標與交付任務的工作。」
- 「軟體專案(software project)」就是以開發或維護軟體系統為目標 的專案
  - 大學專題就是一個典型的軟體專案
  - 通常有多人參與，需團隊合作(teamwork)
- 專案管理通常包含專案規劃與進度追蹤。

專案管理方法案例

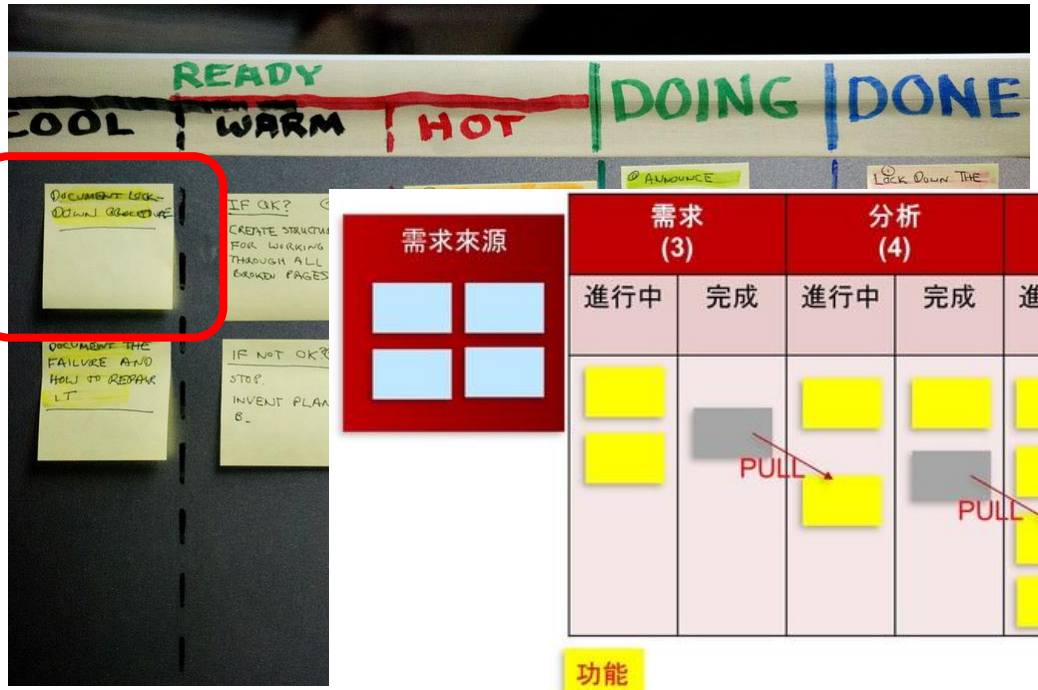
Kanban (看板) & Trello

https://lifehacker.com/productivity-101-how-to-use-personal-kanban-to-visuali-1687948640

Kanban Board (看板)

如何有效把現有流程視覺化，並且改善他!

Card



SELAB
FU JEN CATHOLIC UNIVERSITY

Trello

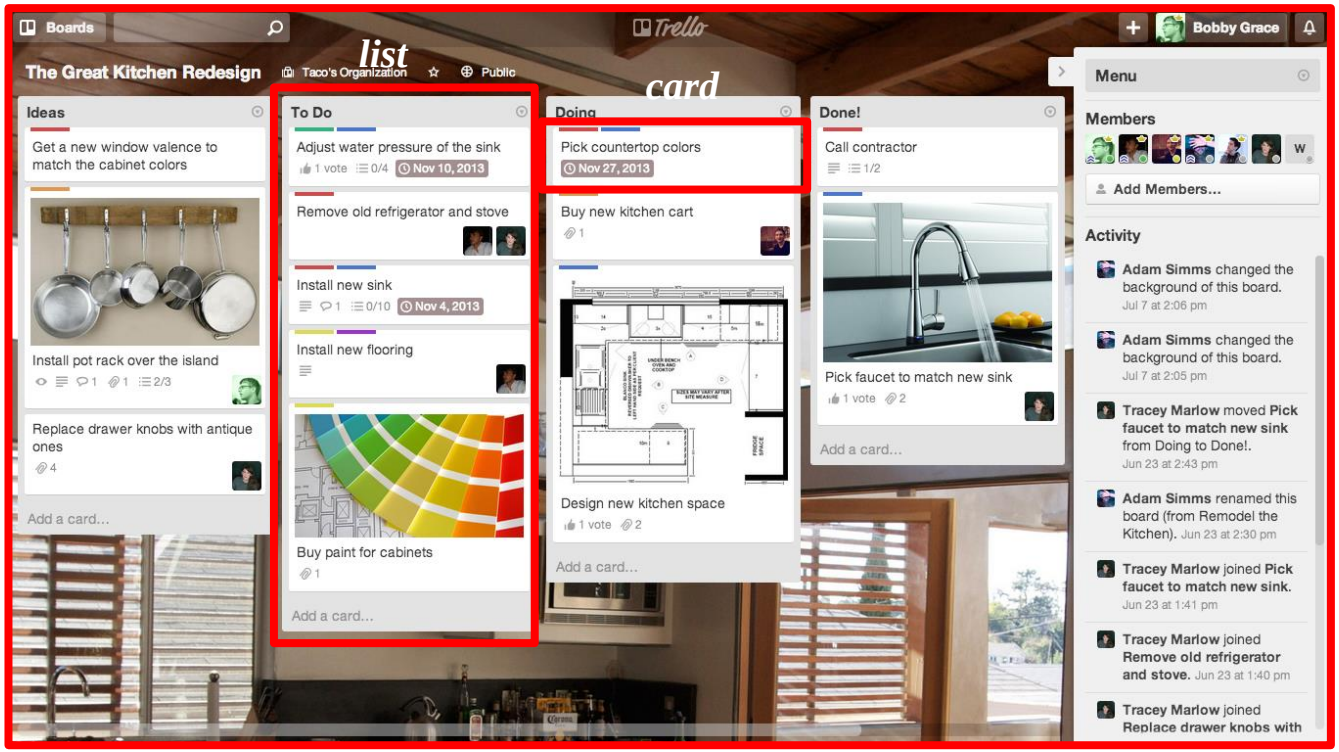
a) 不用錢
b) 適合個人使用
c) 跨平台
d) 簡單易用

- Trello 實現了看板方法
- 如何使用Trello?
  - https://trello.com/b/3ZTiEXNt/welcome-board

- 我們可運用Trello進行專案管理
  - 分工
  - 追蹤進度

Board, List, and Card

board



SELAB
FU JEN CATHOLIC UNIVERSITY

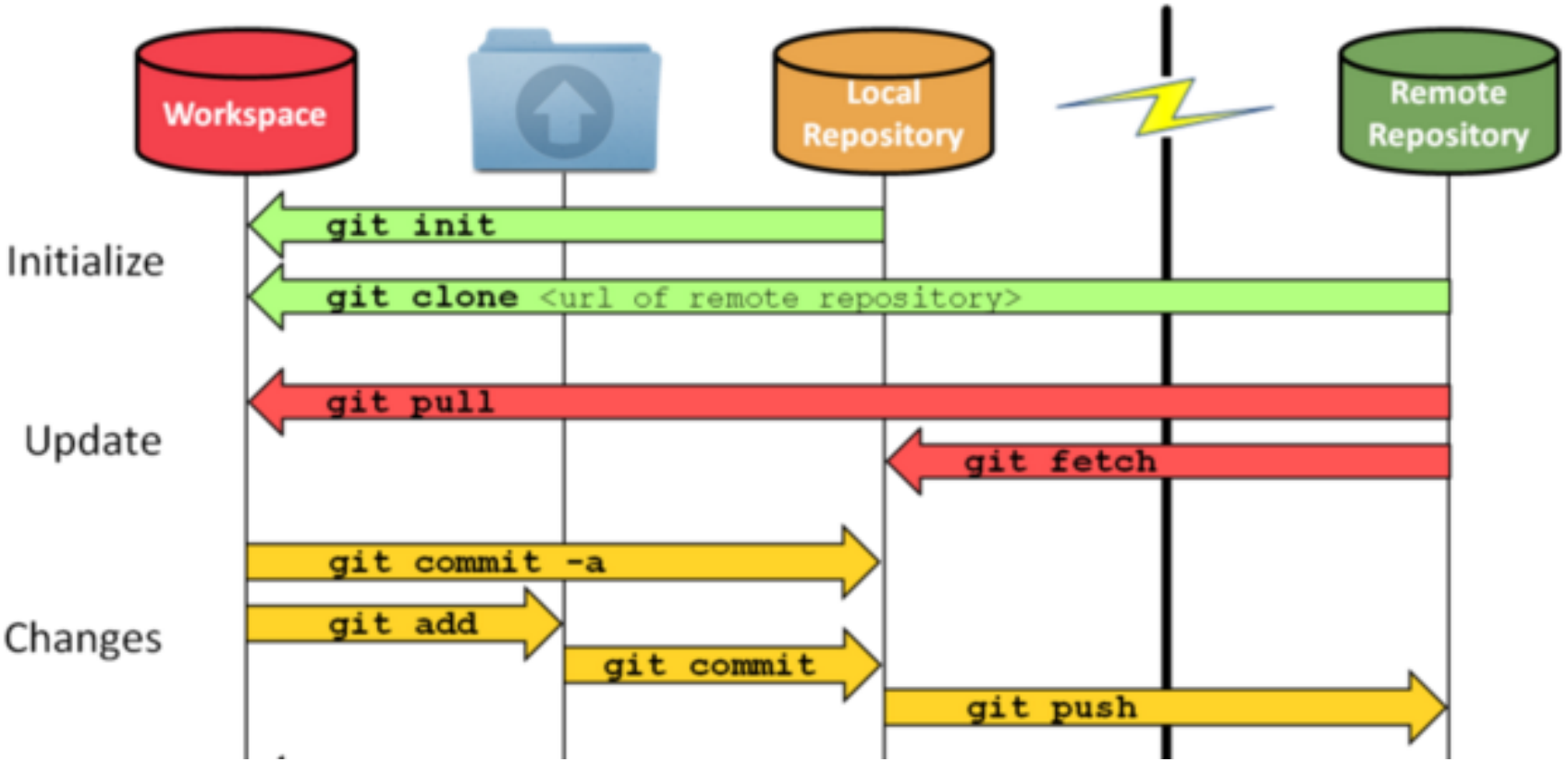
版本控制方法案例

Git

甚麼是Git

- 目前最為主流的分散式**版本控管**軟體
- 每個使用者都有一份完整的儲存庫
- 不需要伺服器端的支援就可以運作
- 提交版本都僅提交到本地的儲存庫
- 因為是在本地提交變更，不會有任何權限限制
- Github是基於Git的原始碼託管服務(hosting service)

Git運作流程



Git基本指令與概念

- **git pull**: 遠端儲存庫有更新，要拉至本地端
- **git add**: 選擇欲加入本次版本的變更檔案
- **git commit**: 將本次版本進行提交(至本地端)，建立檔案快照
- **git push**: 從本地儲存庫上傳檔案至遠端

Example 1

Roader: 高速公路與快速道路即時路況社群及語音導引 APP

創新動機1

- 在國道上收聽警廣，卻始終聽不到前方路況
- 廣播資訊囊括全國各式道路，難以過濾所需資訊
- 行進間想撥專線詢問路況或回報，卻影響行車安全

創新動機2

- 出發前想查看路順不順
- 塞車時想了解原因
- 長途旅程時，不知道該怎麼走最快

Roader系統目標

適地性即時路況

路況回報社群

語音互動與導引

操作概念

路況事件回報

上高架後小寶感受到強勁的陣風，於是輕觸螢幕觸發 Roader 接收語音，小寶說「風很大」，Roader 收到後說「謝謝您的回報！」

路況事件連署

回報之後 Roader 將資訊推播給在高架端上前後 3 公里的小華進行連署，小華的 Roader 詢問：「附近駕駛回報此處風大，是否同意？」小華回答「是」，Roader 在完成連署之後將此狀況擴大散播，讓未上高架的人可收到此消息。

接近路況事件提醒

小寶因為擔心風大發生意外，於是在五股轉接道(32K)開下高架，小寶下轉接道後看到前方壅塞，Roader 馬上告知小寶：「700 公尺前車道發生車禍事故，前方回堵 300 公尺。」

使用案例

*Use case No.	Roader-UC-010
*Use case name	回報事件
Summary	在駕駛或副駕駛模式中回報新事件
*Actors	使用者
Pre-Conditions	1. 登入狀態 2. 駕駛或副駕駛模式中
*Description	Actor Actions： 駕駛模式： 1.點擊螢幕 3.回答事故語句 副駕駛模式： 1.點擊回報 ICON 3.選擇欲回報種類 5.填寫資料送出
	System Responses： 2.觸發語音接收 4.確認或者取消 5.謝謝您的回報! 6.將此回報事件納入待連署事件中 2.顯示所有可回報種類 4.根據種類跳出視窗請使用者回報詳細資訊 6.謝謝您的回報!

功能需求

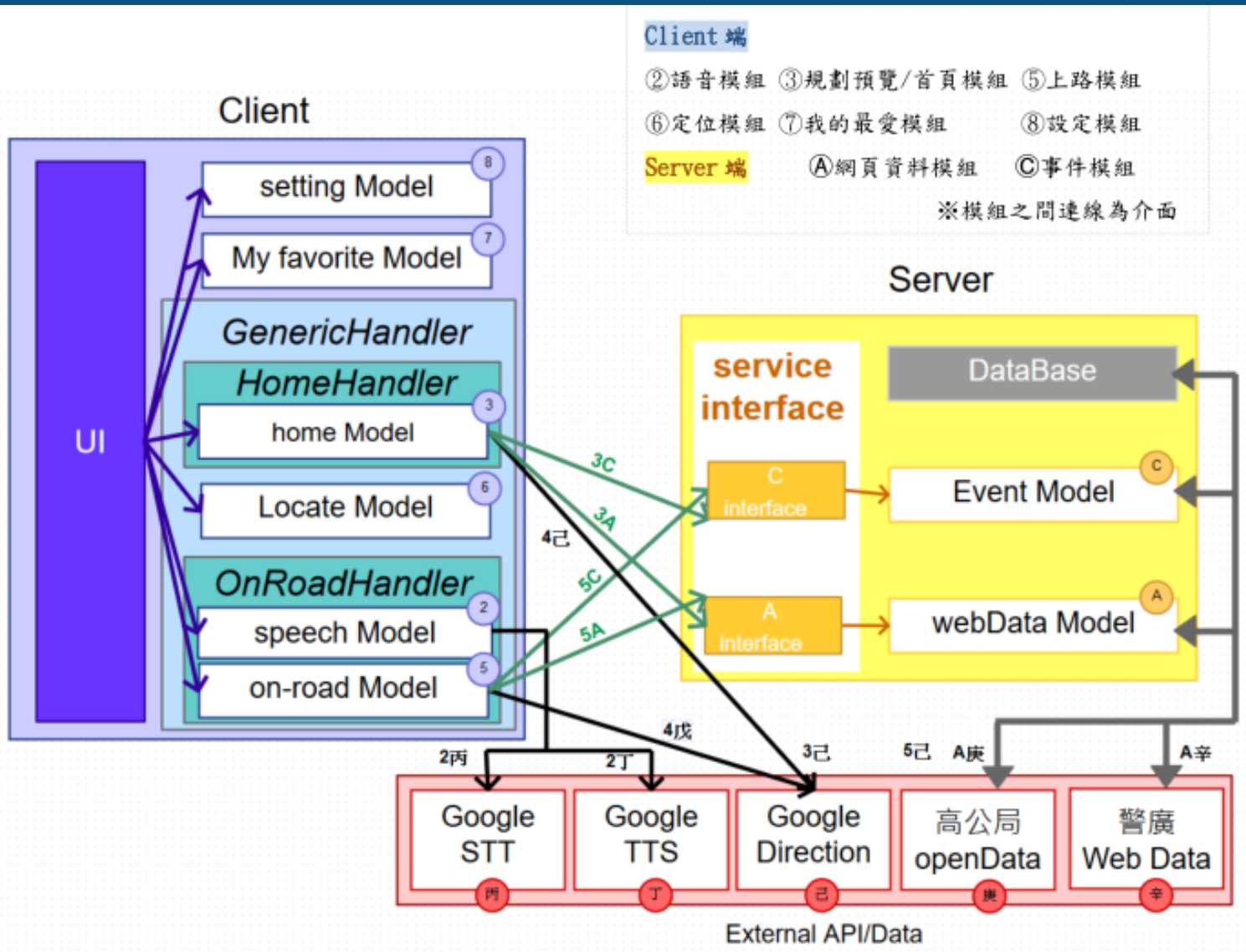
Roader-FR-DM008	語音報告	規劃Roader：語音倒數正在接近的路上事件、告知固定式測速照相位置、車速/最高低速限提醒、接近目的交流道提醒。
		自由Roader：語音倒數正在接近的路上事件、告知固定式測速照相位置、車速/最高低速限提醒。
Roader-FR-DM009	事件連署	當使用者的位置進入其他使用者回報路況事件的範圍，即出現連署畫面，讓使用者進行連署，同時也播放語音，告知使用者可以使用語音指令進行連署，一定時間過後會自動離開該畫面。
Roader-FR-DM010	回報事件	使用者可使用語音指令回報事件(事件分級只有一層) 。
Roader-FR-AM011	切換至副駕駛模式	整個畫面向右(手指向左滑)，即可切換至副駕駛模式。
Roader-FR-AM012	自動結束	離開終點交流道之後依照設定決定是否自動結束Roader。

非功能需求

Roader-NF-001	路徑計算的結果應在3秒內完成。
Roader-NF-002	Server端應容許1000位使用者同時存取資源。
Roader-NF-003	安全的使用環境： 由於在道路上使用手機釀成意外的事件層出不窮，需達到道路上使用最少化，需擁有駕駛語音模式，在道路上時不讓駕駛使用者進行太繁瑣的動作。
Roader-NF-004	不給予不必要資訊： Roader需將路況資訊經過篩選後，根據使用者的位置給予相對應的路況資訊，讓使用者隨時獲得身旁最新資訊。
Roader-NF-005	動態提供最佳路徑： 在駕駛上路後只要接近系統交流道時，Roader會自動依據當時路況動態判斷是否有更快速到達目的地的路徑。

系統架構

軟體架構


Client 端
②語音模組 ③規劃預覽/首頁模組 ⑤上路模組
⑥定位模組 ⑦我的最愛模組 ⑧設定模組
Server 端 Ⓐ網頁資料模組 Ⓒ事件模組
※模組之間連線為介面
Client
UI
setting Model
My favorite Model
GenericHandler
HomeHandler
home Model
Locate Model
OnRoadHandler
speech Model
on-road Model
Server
service interface
DataBase
C interface
Event Model
A interface
webData Model
3C
3A
4己
5C
5A
4戊
2丙
2丁
3己
5己
A庚
A辛
Google STT
Google TTS
Google Direction
高公局 openData
警廣 Web Data
External API/Data

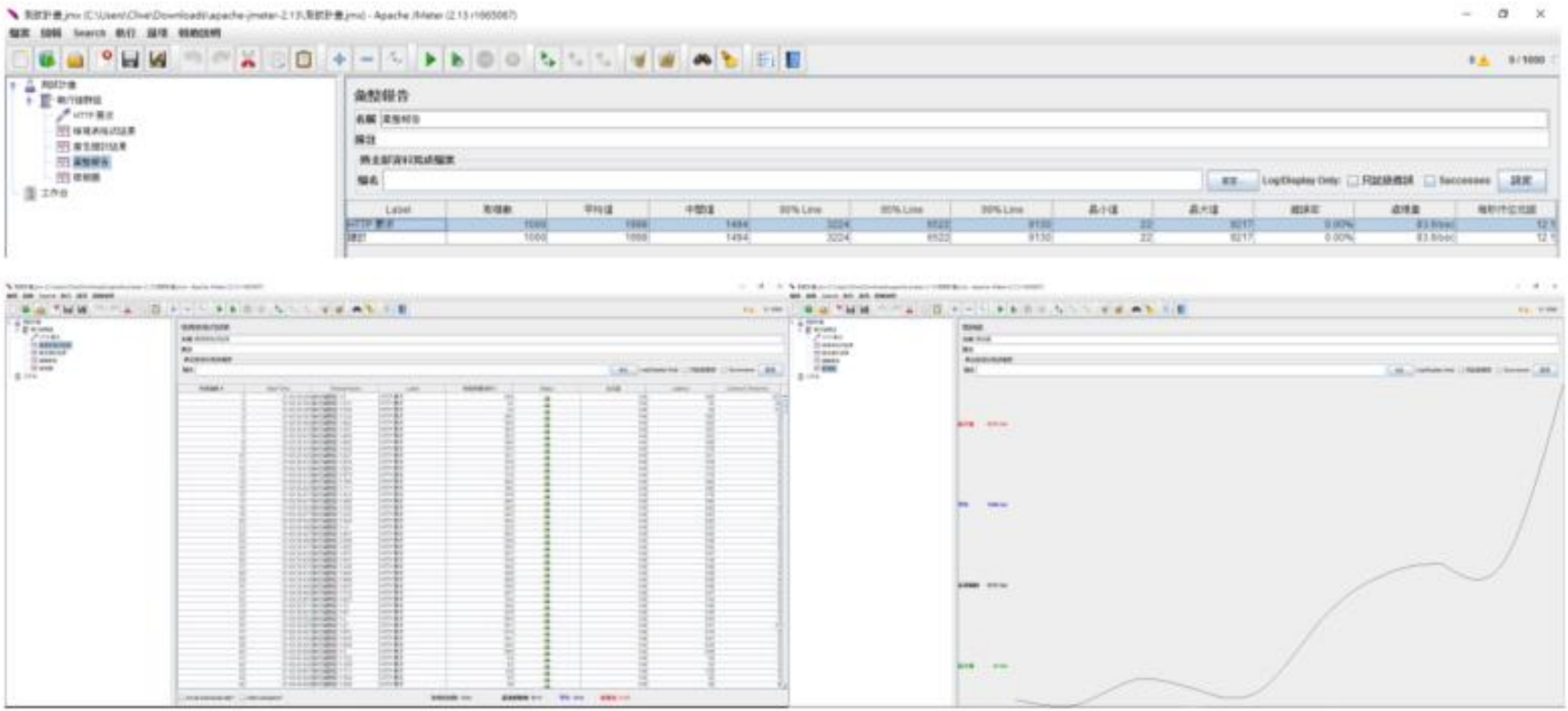
測試案例與測試結果

ID	Roader-TC-003
Name	設定路線提供一或多條參考路徑
Tested target	測試計算一至多條可能路線，並依照路程時間排序
Reference	Roader-FR-PM003
Severity	High
Instructions	1.點擊設定路線 2.選擇基隆端(國道一號-0.6K)為起點 3.選擇旗山端(國道十號 33.8K)為終點
Expected result	顯示一至多條路線並依時間排序給予選擇

Roader-TC-030	Fail	Google Direction 有機率行經一般道路，造成路線判斷錯誤
Roader-TC-031	Pass	
Roader-TC-032	Fail	未能精確倒數
RATE	81.25 %	

效能測試結果

採用 JMeter 對 Server 端做 1 秒同時 1000 次的 HTTP 要求



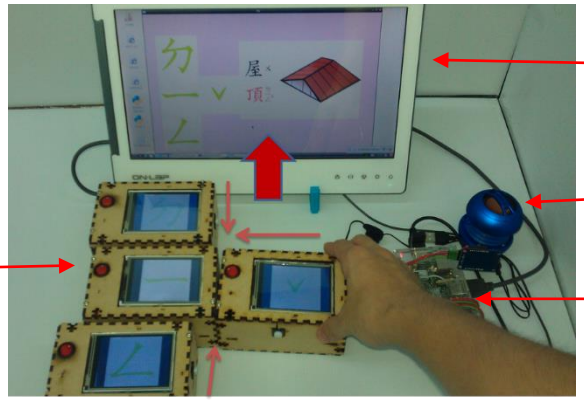
測試結果：失敗率 0.00%，平均回應時間 1998 微秒

Example 2

數位互動式積木系統

需求描述1

- 數位互動式積木系統
  - 系統功能：供兒童使用之注音符號學習系統。

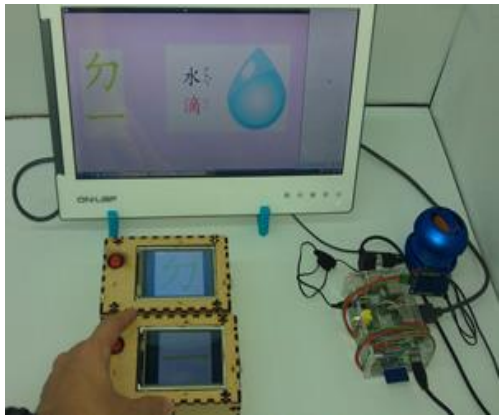


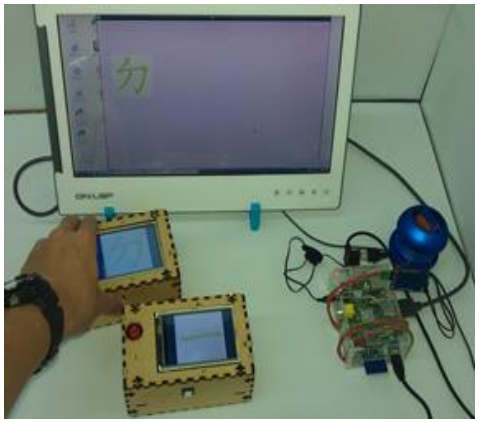
需求描述2

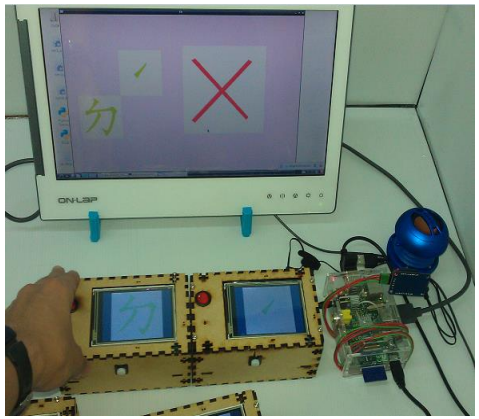
- 操作方式
  - ★ 兒童將數位積木向鄰近數位積木靠攏，觸發感應開關，將積木螢幕顯示注音符號和四方感應開關接觸狀態，傳回主機。
  - ★ 主機接收注音符號與積木開關資訊，運算式否符合注音符號聲母、韻母擺放方式。
  - ★ 主機將積木注音符號顯示於主機螢幕上左方，相關拼音語詞顯示螢幕右方。

需求描述3

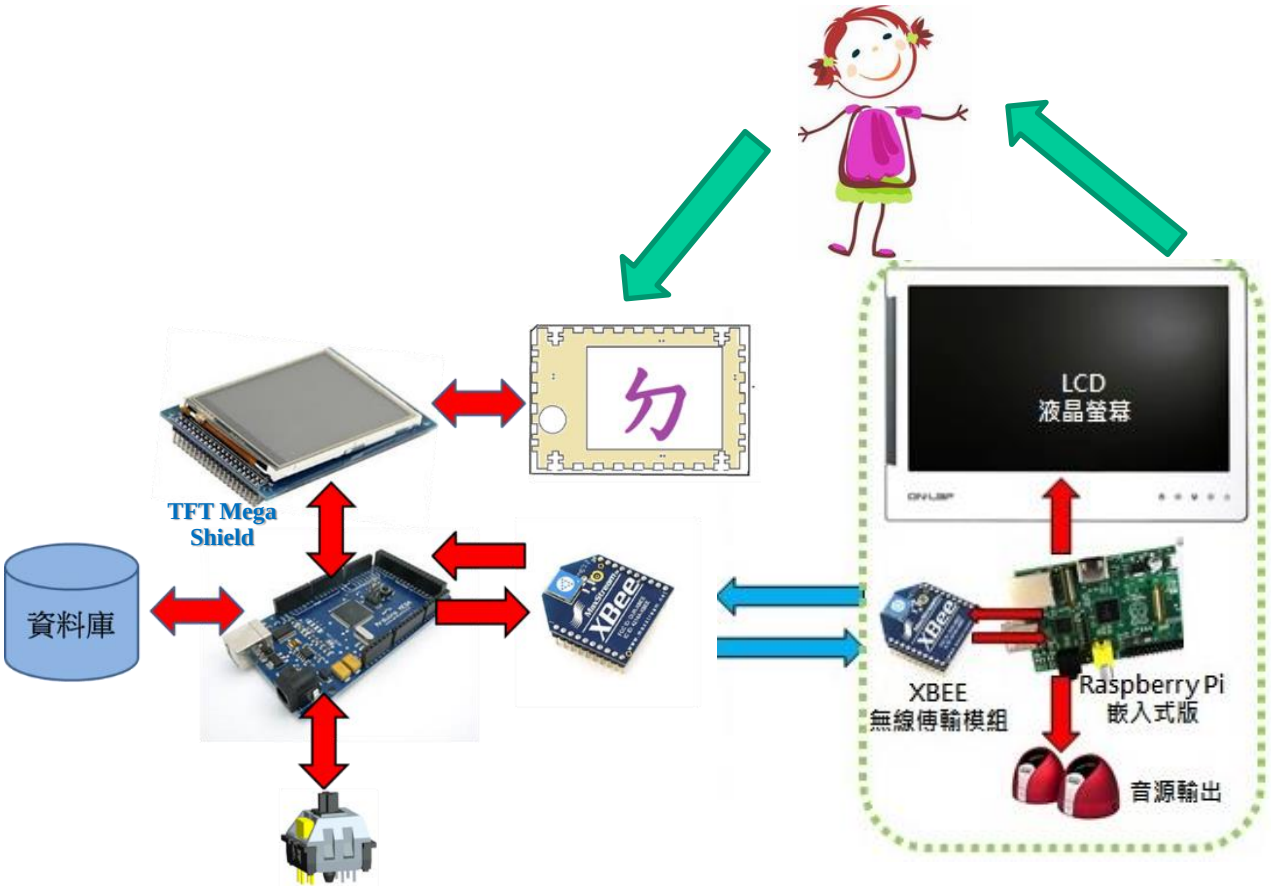
- 操作方式
  - ★ (左)聲母"ㄉ"、韻母”ㄧ”開關觸發
  - ★ (中)挪移積木，僅聲母"ㄉ"開關觸發
  - ★ (右)錯誤顯示，聲母"ㄉ"、聲符"ˊ"開關觸發







系統架構

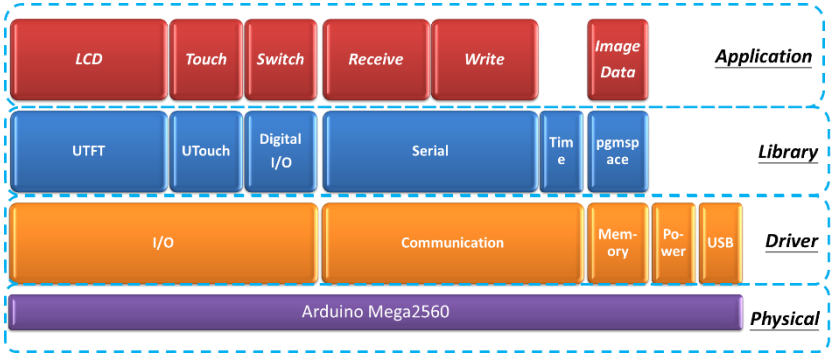


i-Block數位積木

Raspberry Pi多媒體系統

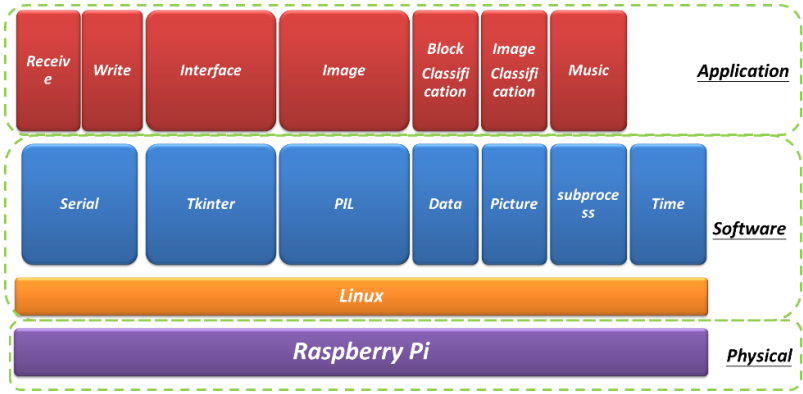
軟體架構設計 - 數位積木

- 分硬體(Physical)、驅動(Driver)、C語言函式庫(Library)、應用(Application)四層。
- 程式撰寫在函式庫與應用層，有積木通訊規則與互動規則。
- 影像傳輸(LCD)、觸控感測(Touch)、感應開關(Switch)、無線資料接收(Receive)、無線資料傳送(Write)、影像資料庫(Image Data)。



軟體架構設計 - Raspberry Pi

- 分硬體(Physical)、軟體(Software)、應用(Application)三層。
- 程式設計於軟體及應用層上，應用層以Python撰寫。
- 軟體模組分七區塊：無線接收(Receive)、無線傳送(Write)、使用者界面(Interface)、圖像輸出(Image)、積木資料分類(Block classification)、圖像資料分類(Image classification)、音訊輸出(Music)。



測試

Arduino UNO與觸控螢幕連線測試
觸控螢幕畫線、寫字、按鈕測試
觸控螢幕圖片點陣圖顯示測試
觸控螢幕與Arduino底層函式庫修正測試
觸控螢幕圖片其他格式圖片顯示測試
Arduino UNO 控制紅外線收發模組測試
接觸式按鈕測試
Arduino UNO圖像最大化限制測試
Arduino MEGA圖像最大化限制測試
樹梅派GPIO腳位、RS232通訊系統
樹梅派廣播最大容忍值測試
電子積木測試
電子積木BUG修正測試
互動系統穩定度調整測試
軟硬體系統整合測試

再來聽一下軟體工程為甚麼重要

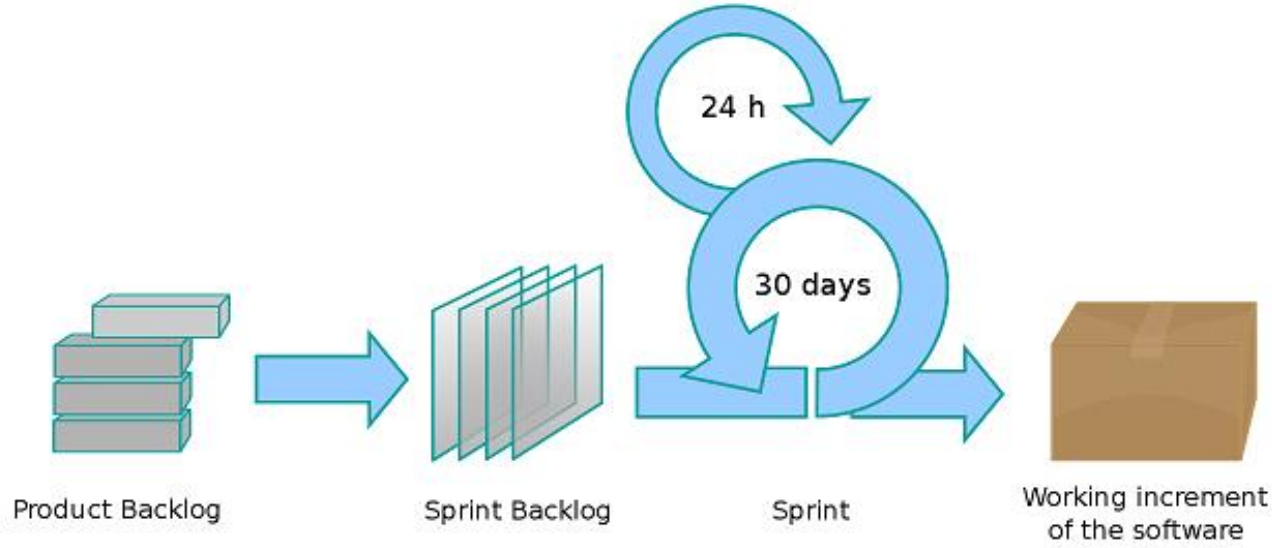
https://www.youtube.com/watch?v=R3NzTt0BTWE

最後談一下敏捷開發

- 更多的溝通，不是按表操課
- 多階段開發，不用一次到位

敏捷宣言(Agile Manifesto, Utah, USA, 2001)

- 個人與互動　重於　流程與工具
- 可用的軟體　重於　詳盡的文件
- 與客戶合作　重於　合約協商
- 回應變化　重於　遵循計劃



THANKS!

Any questions?

You can find me at:
yyfanj@csie.fju.edu.tw

SELAB
FU JEN CATHOLIC UNIVERSITY