



Part I

軟體工程簡介

李信杰 副教授

國立成功大學資訊工程學系



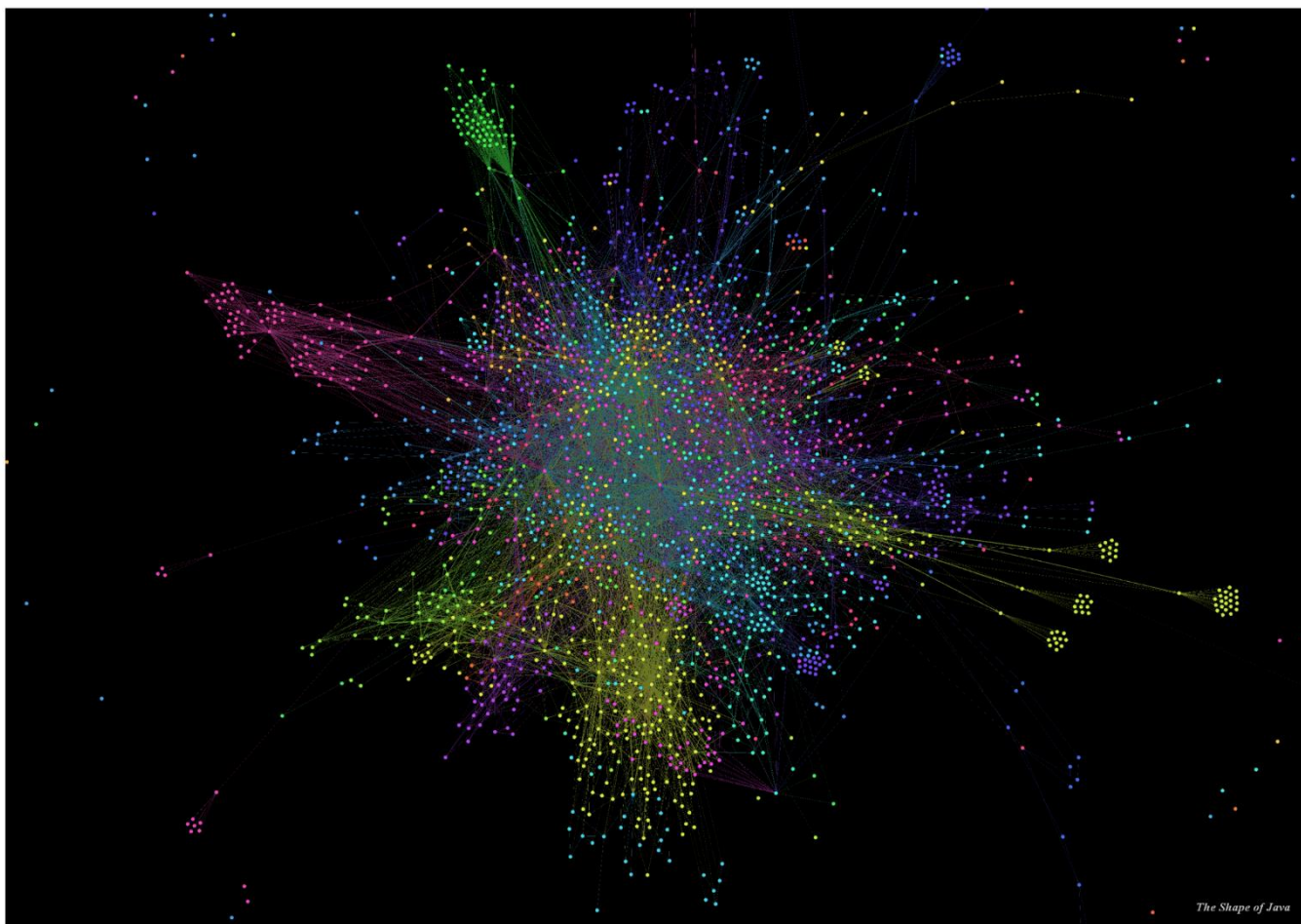
Lab上傳網址

<https://forms.gle/aAo8rwuowvMgESd17>



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。





真實案例

- 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
- 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>



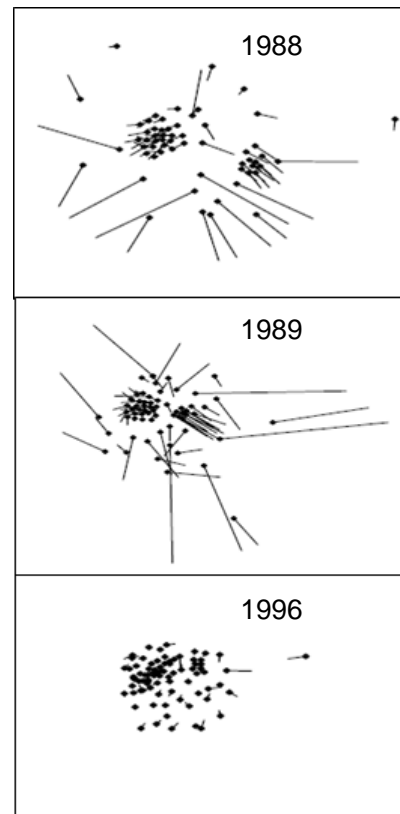
腦力激盪

- ☐ 你覺得什麼樣的程式是複雜的？
- ☐ 複雜度有辦法度量嗎？
- ☐ 遇到複雜的程式你會想怎麼做？
- ☐ 同一份系統需求，由不同的人開發，複雜度會差不多嗎？



軟體衰退問題

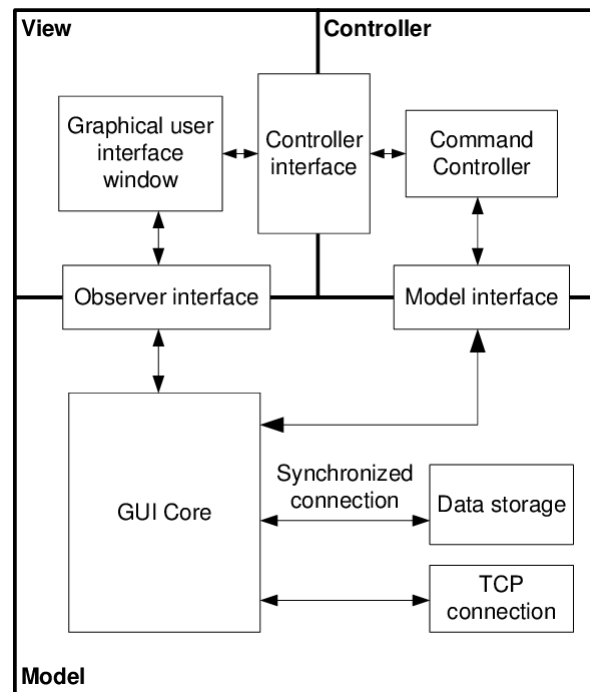
- ❑ 如果程式碼變得越來越難修改，那麼此程式碼已經衰退了。
- ❑ An Example: The Telephone Switches Project
 - Fifteen-year old real-time software system for telephone switches
 - 100,000,000 lines of source code (C/C++) and 100,000,000 lines of header and make files, organized into some 50 major subsystems and 5,000 modules.
 - More than 10,000 software developers have participated.
 - A module within one of the clusters is often changed together with other modules in the cluster but not with other modules
 - Head of each tadpole-like shape corresponds to a module





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。





腦力激盪

- ☐ 生活中你觀察到有哪些一致性？
- ☐ 不一致的結果會造成什麼災難？
- ☐ 整合介面有可能改嗎？



軟體的本質問題: 可見性(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。



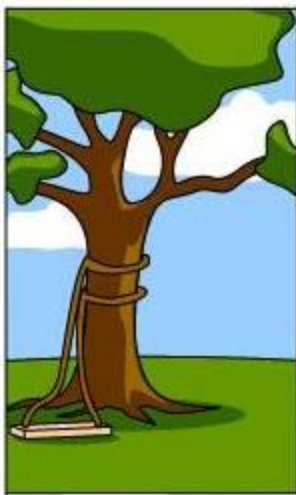
顧客解釋他們想要的



專案主持人對顧客
需求的認知



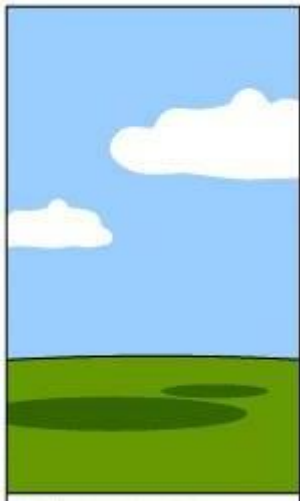
系統分析師所設計的



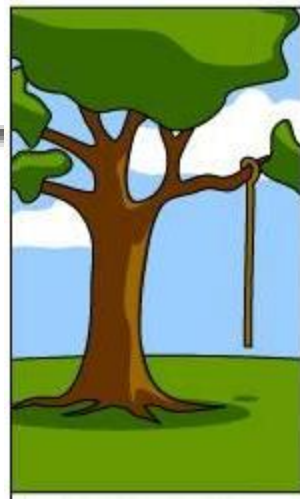
程式設計師所寫出
來的



顧問所描繪的願景



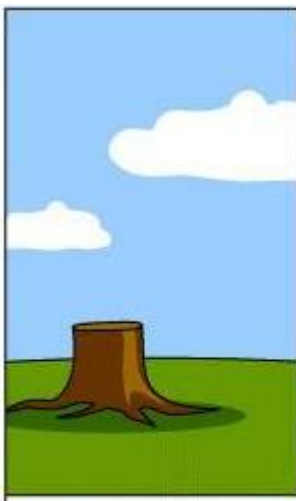
專案的文件



最後交付給顧客的軟體



顧客所付的錢



上線後的技術支援



顧客所真正需要的



腦力激盪

□ 你認為軟體可由哪些面向來進行討論？



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。





腦力激盪

- ☐ 你認為軟體開發過程當中有哪些產出是可能變更的？
- ☐ 變更不好嗎？可以避免變更嗎？



**我們需要一些方法和工
具來協助我們！**

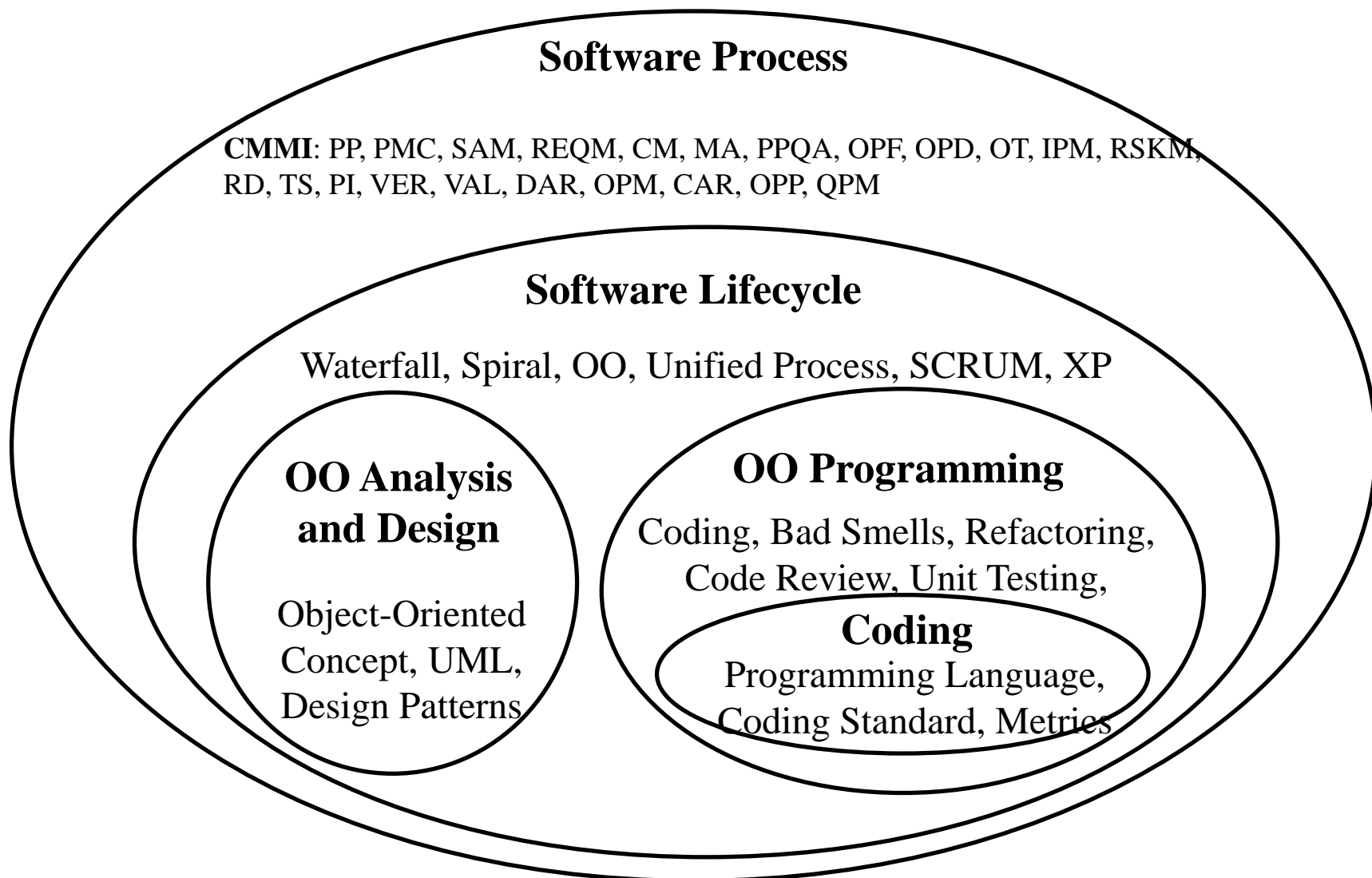


甚麼是軟體工程？

- "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

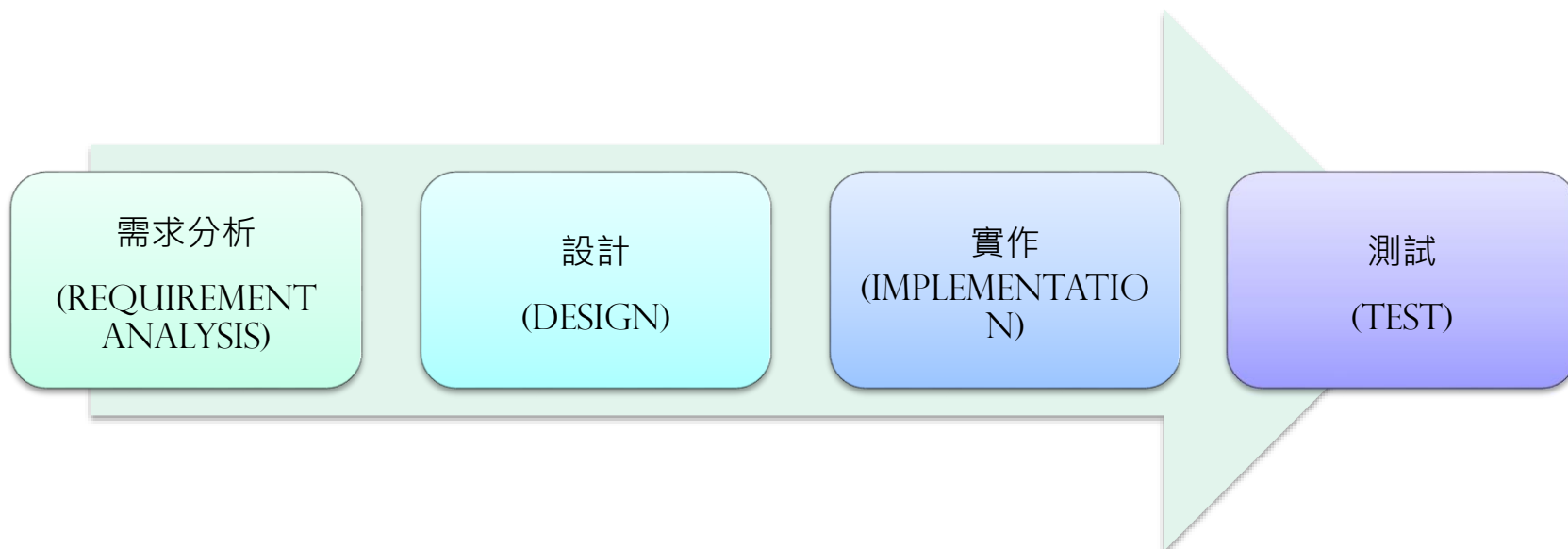


Software Engineering





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) 。



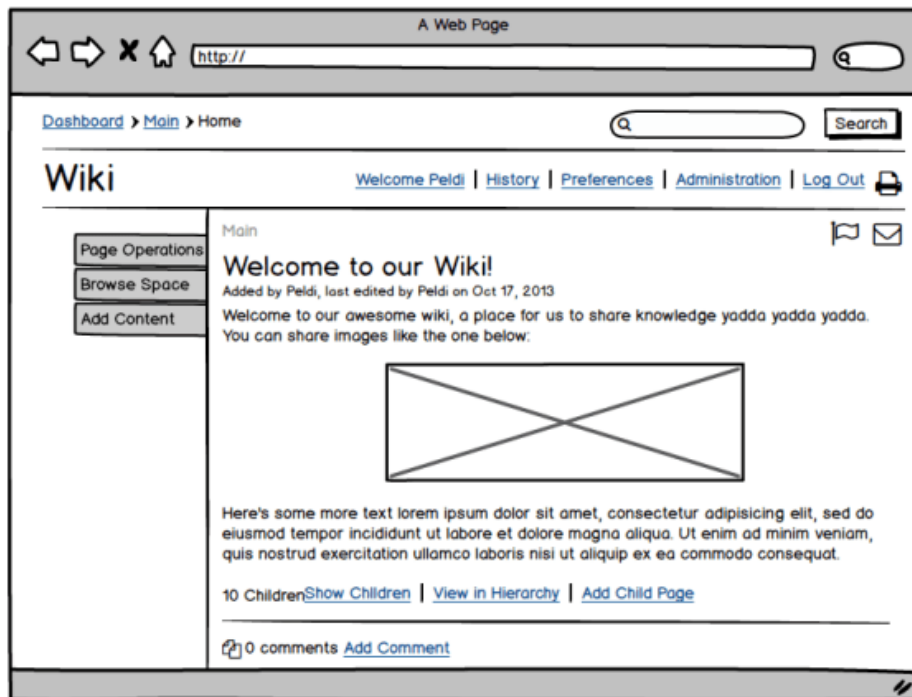
你可以先訂出使用案例

- 透過**情境**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！



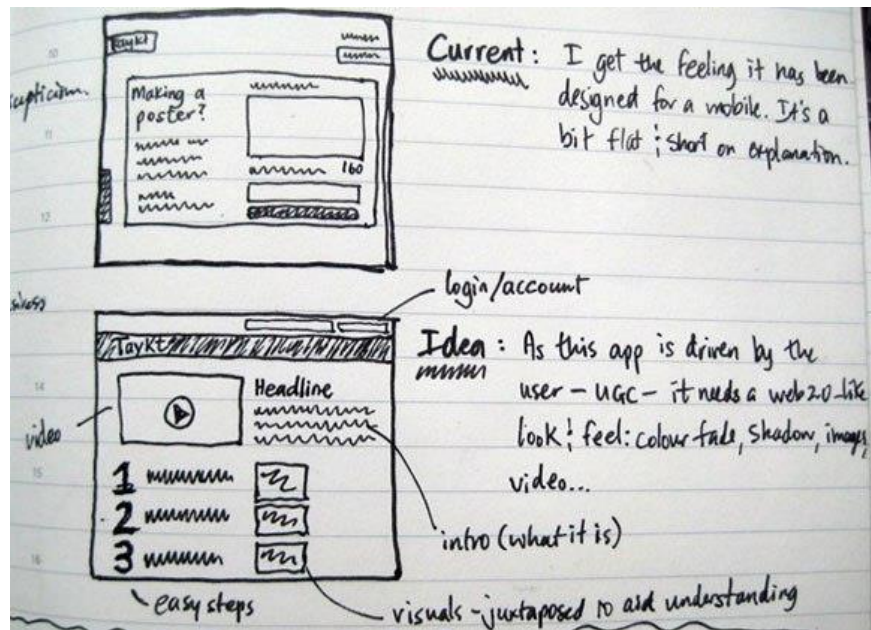


Wireframe



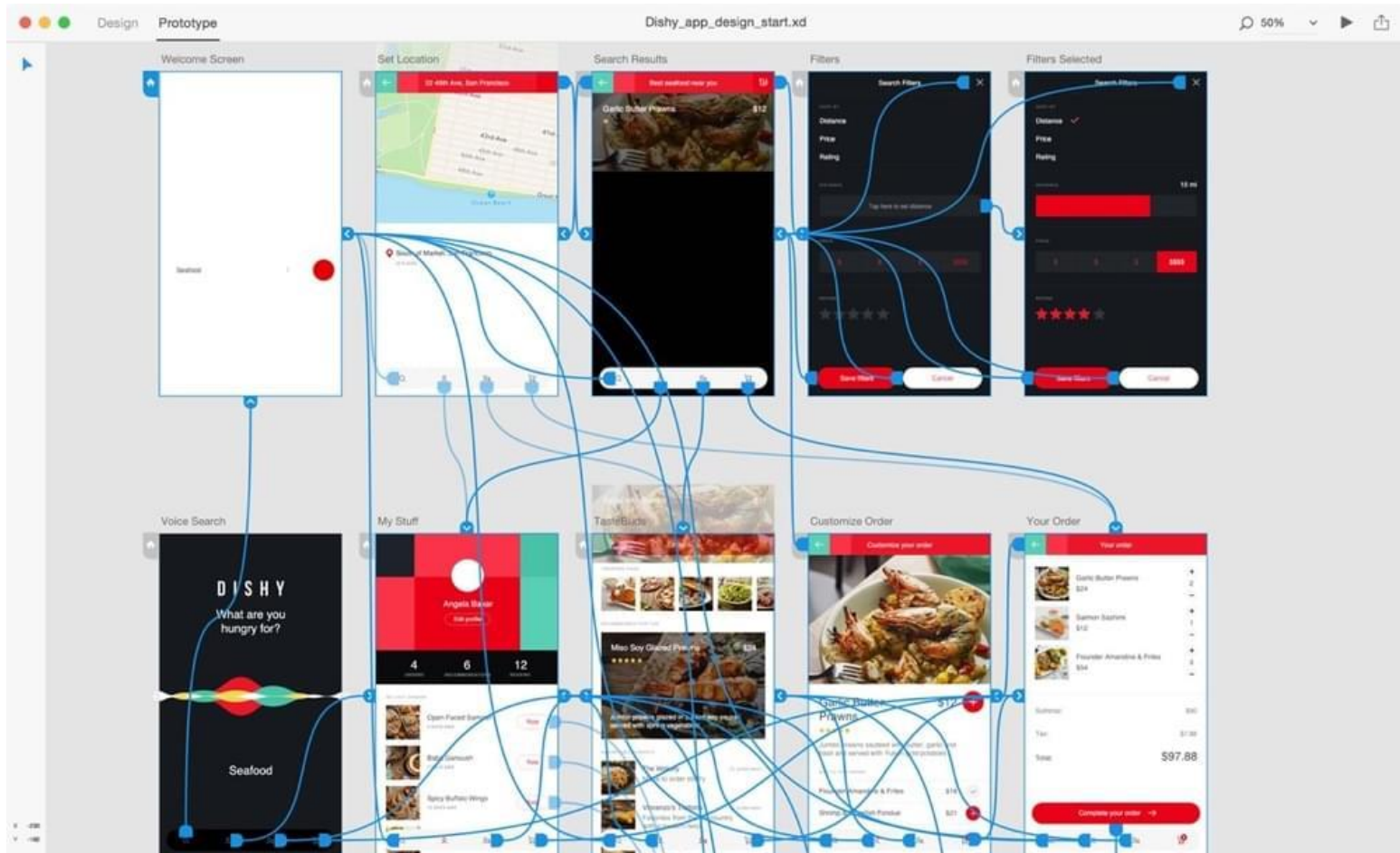
Wireframe、Mockup與Prototype的差異？來看看完整的產品UI設計流程

<https://koku.chat/2017/06/22/wireframe%E3%80%81mockup%E8%88%87prototype%E7%9A%84%E5%B7%AE%E7%95%B0%E7%BC%9F%E4%BE%86%E7%9C%8B%E7%9C%8B%E5%AE%8C%E6%95%B4%E7%9A%84%E7%94%A2%E5%93%81ui%E8%A8%AD%E8%A8%88%E6%B5%81%E7%A8%8B/>





Mockup





Prototype



https://www.researchgate.net/figure/IndustBox-software-prototype_fig5_311929106



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。



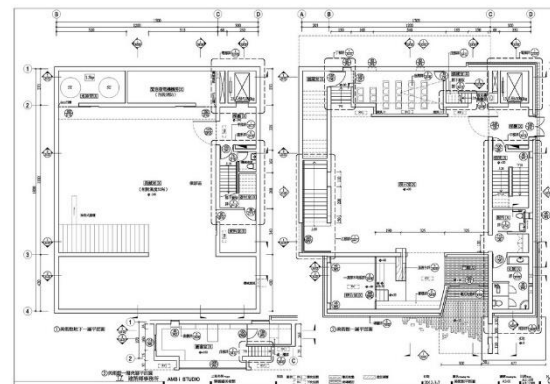
腦力激盪

- ☐ 「客戶」與「使用者」是否有差別？
- ☐ 需求有可能完整嗎？
- ☐ 把需求釐清清楚是誰的責任？客戶？開發團隊？
- ☐ 需求是否有可能被誘導？
- ☐ 若有不合理的需求該怎麼辦？



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)**。





腦力激盪

- ☐ 不良的設計會帶來什麼樣的後果？
- ☐ 有速效的一套設計課程嗎？
- ☐ 設計間是否會有衝突？該怎麼辦？



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。

```
(key)) tempString = tempString.replace("czDate",  
rt(value*pow(10,14-tmpFormat))) tempString =  
elif(typeOfField == "BUFFER"): s = value dataCal  
tempString.replace("czFieldID",str(key)) temp  
elif(typeOfField == "ASCII_STRING"): s = value d  
buffer") tempString = tempString.replace("czDate",  
res: if "date value=" in line and flagCheckRi  
mp(1) if "</Message>" in line: myEvent = "RT  
onlyfilename+"\n" if typeOfFile  
if not os.path.exists(path)
```

<https://www.gqrgm.com/guest-post-11-programming-languages-to-learn-for-new-opportunities/>



腦力激盪

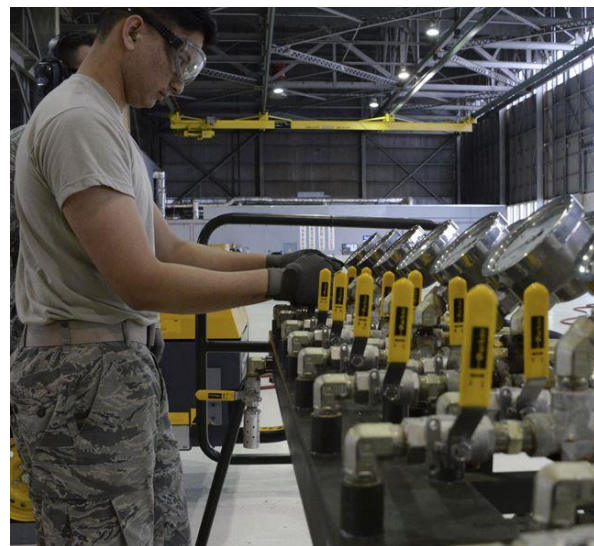
- ☐ 世界上有那些主流平台?
- ☐ 我該盡可能學會所有主流程式語言嗎?
- ☐ 從入門到專精，學習程式語言需要多久?
- ☐ 程式寫的快與寫的好，哪個比較重要?



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。





腦力激盪

- ☐ 有測試過的系統一定就沒問題嗎?
- ☐ 測試需要經驗嗎?
- ☐ 測試到什麼程度即可滿足?



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，不知道該怎么走最快





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 Roder 接收語音，小寶說「風很大」，Roder 收到後說「謝謝您的回報！」

路況事件連署

回報之後 Roder 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 Roder 詢問：「附近駕駛回報此處風大，是否同意？」小華回答「是」，Roder 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，Roder 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 :	System Responses :
	駕駛模式： 1. 點擊螢幕 3. 回答事故語句	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
	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	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

Roder-FR-DM008	語音報告	規劃Roder：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Roder：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。
Roder-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。
Roder-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)。
Roder-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。
Roder-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Roder。



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。

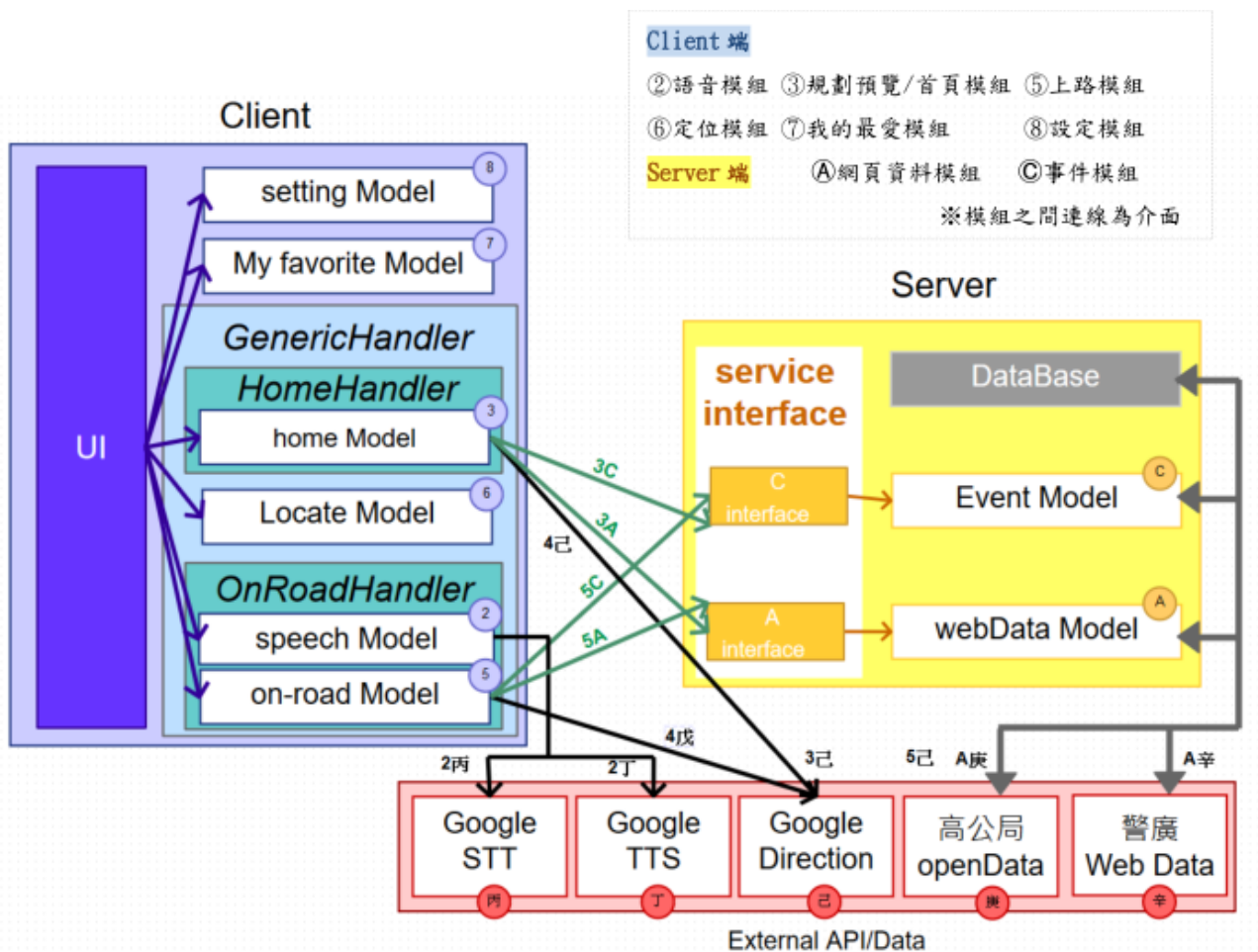


系統架構





軟體架構





測試案例與測試結果

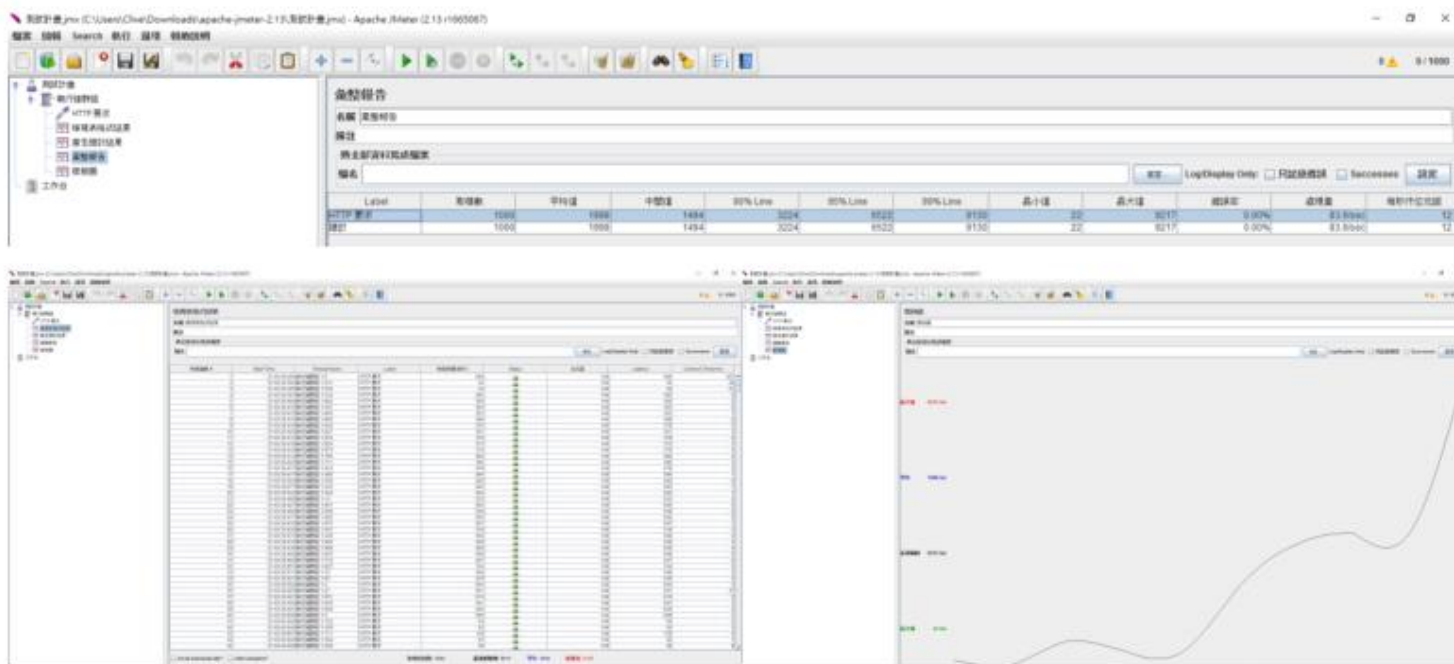
ID	Roader-TC-003
Name	設定路線提供一或多條參考路徑
Tested target	測試計算一至多條可能路線，並依照路程時間排序
Reference	Roader-FR-PM003
Severity	High
Instructions	1.點擊設定路線 2.選擇基隆端(國道一號-0.6K)為起點 3.選擇旗山端(國道十號 33.8K)為終點
Expected result	顯示一至多條路線並依時間排序給予選擇

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

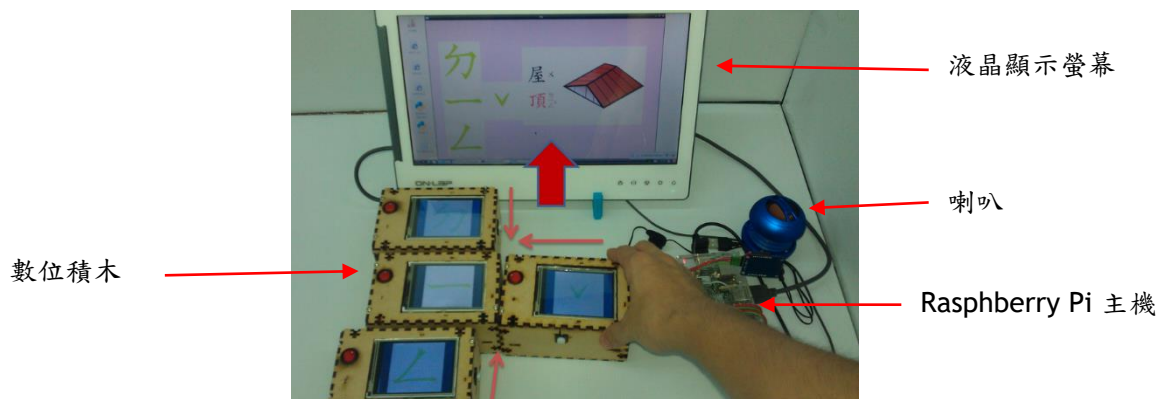


需求描述₁

□ 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

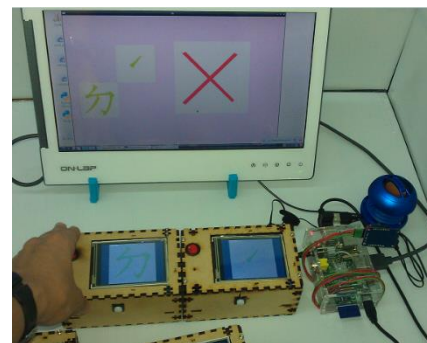
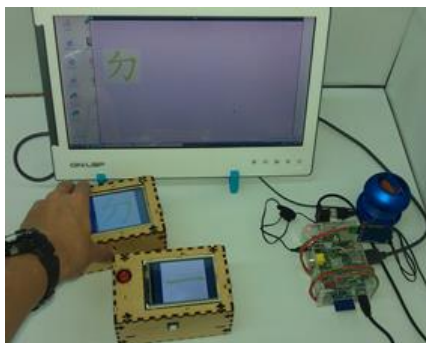
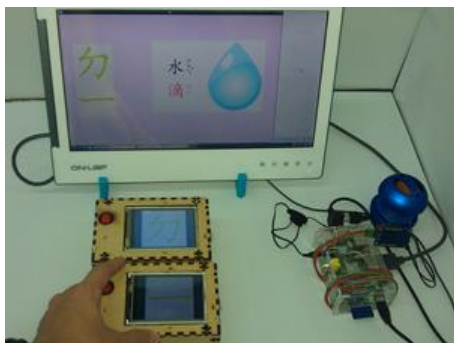
- 兒童將數位積木向鄰近數位積木靠攏，觸發感應開關，將積木螢幕顯示注音符號和四方感應開關接觸狀態，傳回主機。
- 主機接收注音符號與積木開關資訊，運算式否符合注音符號聲母、韻母擺放方式。
- 主機將積木注音符號顯示於主機螢幕上左方，相關拼音語詞顯示螢幕右方。



需求描述³

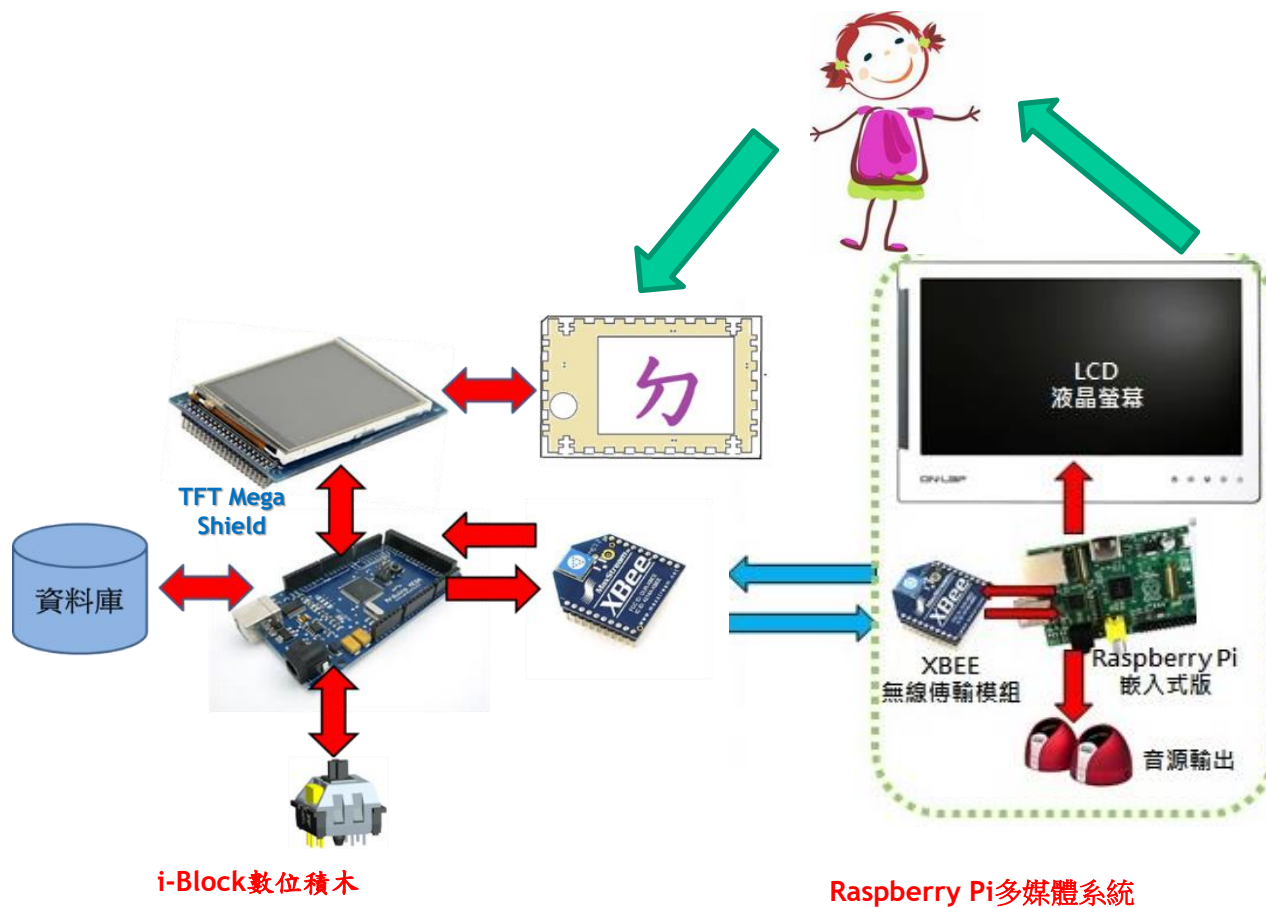
□ 操作方式

- (左) 聲母”ㄉ”、韻母”一”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”丿”開關觸發





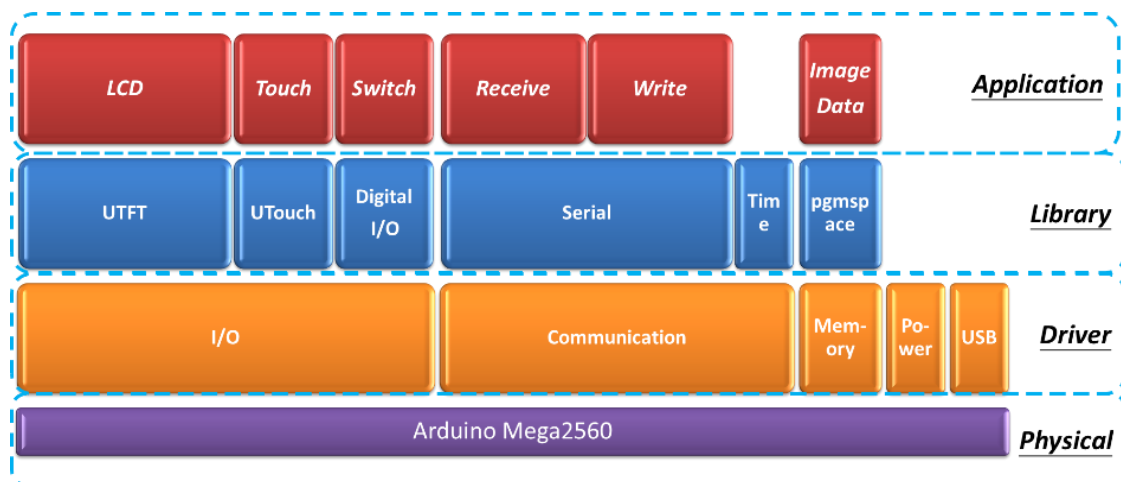
系統架構





軟體架構設計 - 數位積木

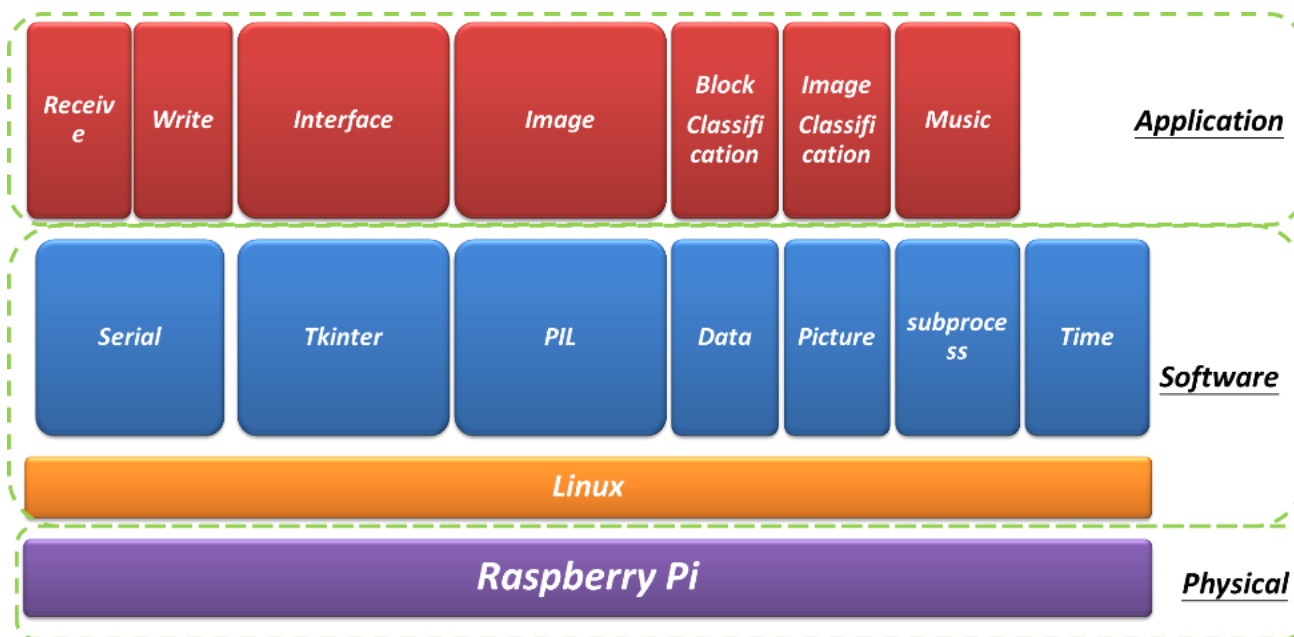
- 分硬體(Physical)、驅動(Driver)、C語言函式庫(Library)、應用(Application)四層。
- 程式撰寫在函式庫與應用層，有積木通訊規則與互動規則。
- 影像傳輸(LCD)、觸控感測(Touch)、感應開關(Switch)、無線資料接收(Receive)、無線資料傳送(Write)、影像資料庫(Image Data)。





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



Lab 1-1 需求描述

- ☐ 請提出一個假想系統並寫下其需求描述 (30分鐘)
- ☐ 討論分享(20分鐘)



Lab 1-2 Wireframe

- ☐ 請依據需求描述繪製出一個以上的Wireframes (30分鐘)
- ☐ 討論分享(20分鐘)



Lab 1-3 系統架構

- ☐ 請繪製出一張系統架構圖(30分鐘)
- ☐ 討論分享(20分鐘)