

智慧行動應用面面觀

Agenda

- 行動科技發展趨勢
- 行動APP開發模式說明
- 混合式專案管理方法論
- 最佳實例情境應用

行動科技發展趨勢



行動化趨勢

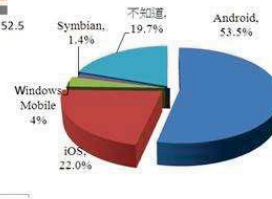


資料來源：資策會FIND/經濟部技術處「科技化服務價值鏈研究與推動計畫」

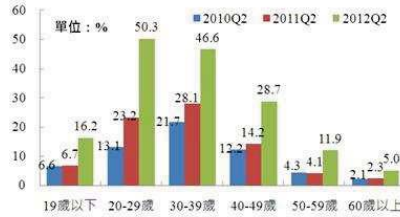
智慧手機趨勢



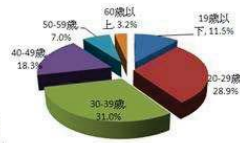
臺灣民眾持有智慧手機普及率



智慧型手機作業系統分佈結構



不同年齡層民眾持有智慧型手機普及率



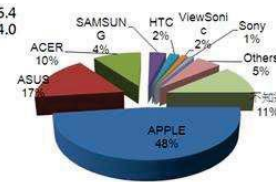
持有智慧型手機民眾年齡結構

資料來源：資策會FIND/經濟部技術處「科技化服務價值鏈研究與推動計畫」

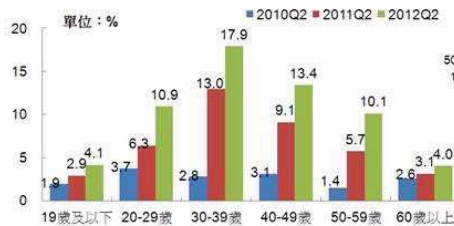
平板電腦趨勢



臺灣民眾持有平板電腦普及率



持有平板電腦的品牌分佈



不同年齡層民眾持有平板電腦普及率



持有平板電腦民眾年齡結構

資料來源：資策會FIND/經濟部技術處「科技化服務價值鏈研究與推動計畫」

行動應用趨勢 - 平板帶領企業IT行動化

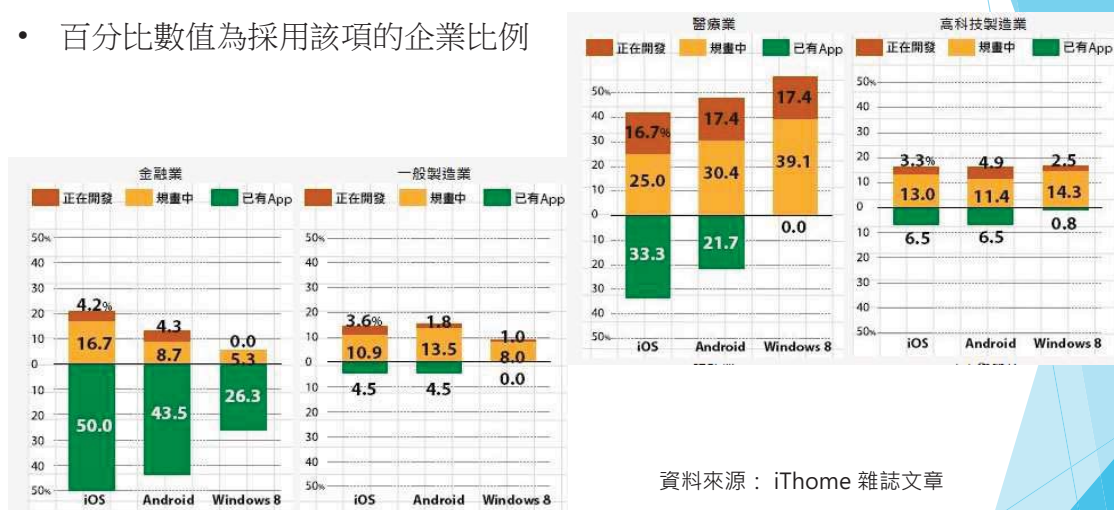
- 高階主管與業務部門是企業平板主力用戶：



資料來源：iThome 雜誌文章

各產業行動App平臺導入趨勢

- 百分比數值為採用該項的企業比例



資料來源：iThome 雜誌文章

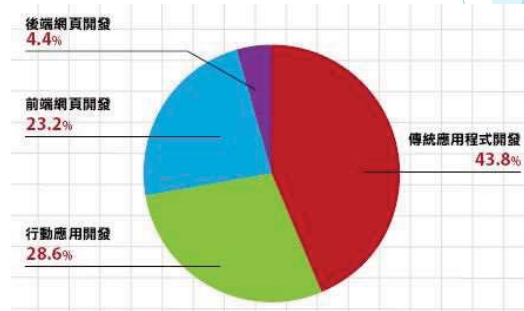
企業內部應用趨勢

企業內部應用的開發趨勢

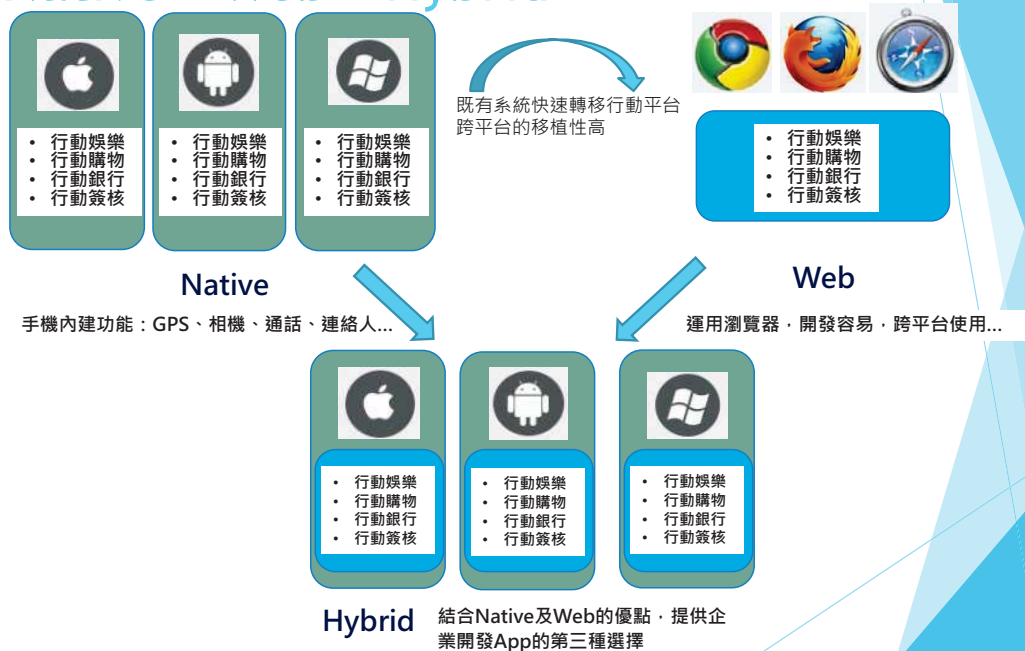


資料來源：iThome 雜誌文章

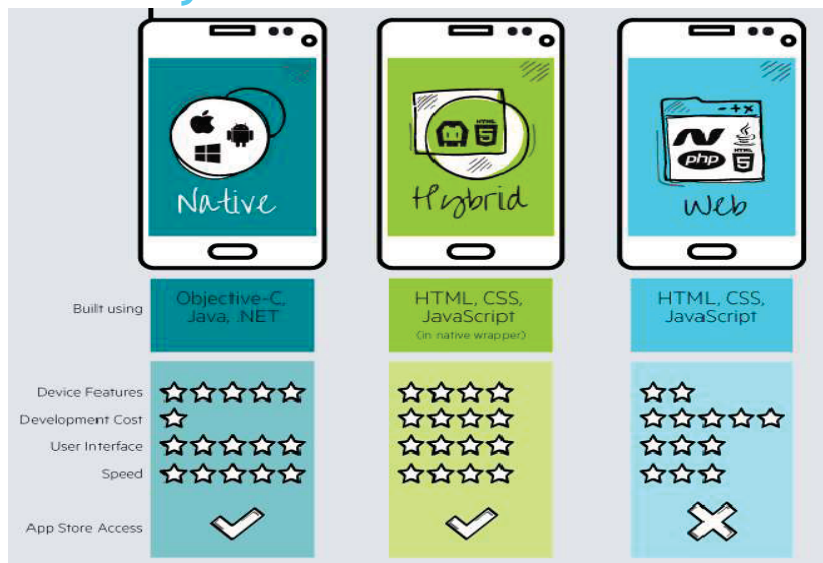
企業擬優先強化的開發技術



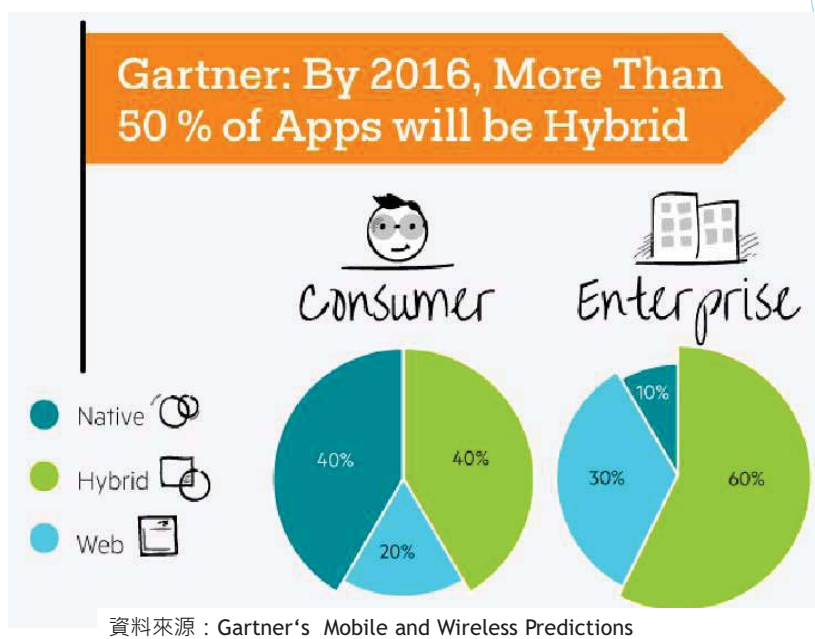
App開發方式分類與發展： Native、Web、Hybrid



APP開發模式： Native、Hybrid、Web的不同特色



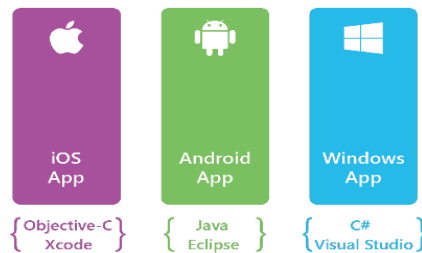
資料來源：Gartner's Mobile and Wireless Predictions



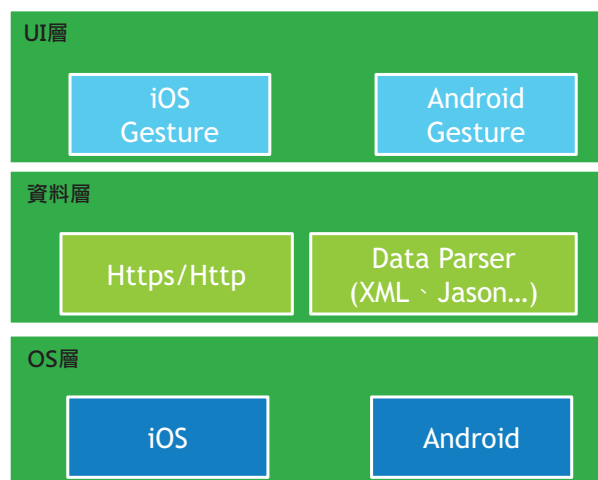
資料來源：Gartner's Mobile and Wireless Predictions

原生應用程式 (Native App) 開發

- 在此所指的原生應用程式 (Native App)，指的是作業系統廠商本身提供 SDK 或建議的開發方式。
- 使用該平台專屬的語言和SDK開發
iOS : Xcode & Objective-C
Android : Eclipse & Java
Window Mobile : Visual Studio & C / C++ / C# / VB.NET



Native App 軟體架構示意：

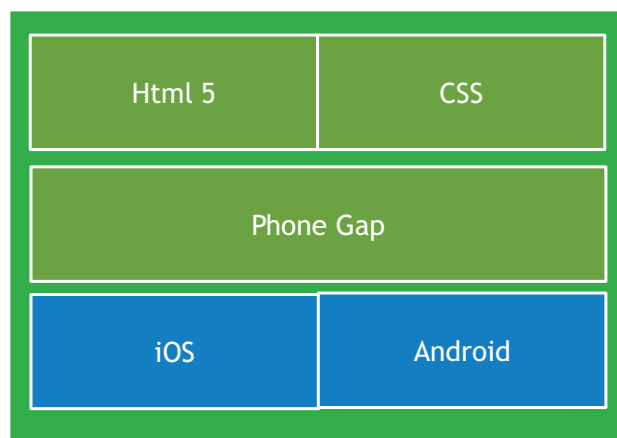


混合式應用程式 (Hybrid App) 開發

- 混合式應用程式 (Hybrid App) 開發在此指的是，以一般以 **Web App** 方式開發用戶端程式，以 **HTML / XHTML / HTML5 + CSS + Java Script ...** 等網頁標準技術製作與硬體無關的部份。
- 透過 **PhoneGap** 框架工具跟行動裝置硬體設備互動，或加上部份原生程式(主要是和硬體互動的部分，和瀏覽器元件)，讓**Web App** 混合一些平台專有的**Code**，包裝成原生 App。



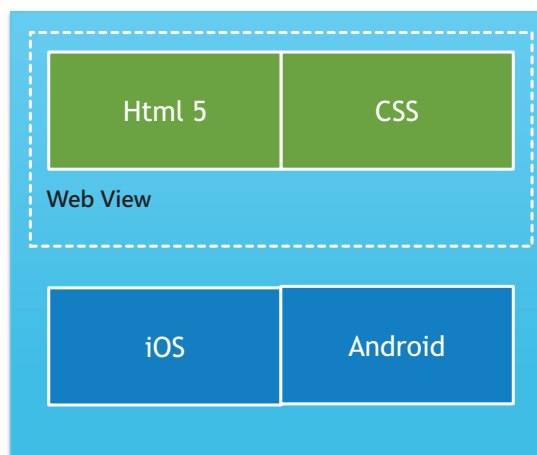
Hybrid App 軟體架構示意：



網頁應用程式 (Web App) 開發

- **Web App** 本身透過網頁瀏覽器操作執行，主要以行動裝置上的瀏覽器應用為主，所以又經常被稱為行動網頁 **App (Mobile App)** 。
- 嚴格來說，不是一個 **App**，必須透過瀏覽器輸入網址後執行。
- 前端網頁使用 **HTML / XHTML / HTML5 + CSS + Java Script ...** 等網頁標準技術製作。
- 後端使用 **PHP、ASP.NET、JSP ...** 等程式語言開發，並連結資料庫或其它資料來源。

Mobile Web 軟體架構示意：



選擇APP開發技術前的思考重點

客戶需求方向	原生應用程式 (Native App)	混合式應用程式 (Hybrid App)	網頁應用程式 (Web App)
使用者操作介面需求較為複雜	V	△	X
獨特互動操作的使用者體驗	<div style="border: 1px solid black; padding: 5px; background-color: #0056b3; color: white; text-align: center;"> 互動性需求高 高規格使用者體驗 與設備高度整合 </div>		X
使用者平台操作經驗的一致性			X
將運用到手機硬體裝置底層的功能			X
必須即時傳送與接收大量數據(IM)			X
資料有離線作業需求	V	△	X
跨平台通吃	X	<div style="border: 1px solid black; padding: 5px; background-color: #008000; color: white; text-align: center;"> 企業內部應用 既有系統快速轉換 企業形象入口 </div>	
開發與維運成本較低	X		
重視網頁SEO	X		

資料來源：美國網頁開發與設計公司 Table XI 提供。

APP的開發考量：設想使用者情境

- 開發企業行動應用程式時，可以多設想使用者的操作情境。如操作地點、網路連線狀況、頻寬、是否經常臨時以單手操作、最重要的資訊內容等，才能針對各種應用程式的特性，選擇較適合的開發方式。
- 例如，如果常常需要離線操作，連線時才批次將異動內容與資料庫同步，就不適合使用WEB APP的方式了。



APP的開發考量：傻瓜思考

- 應用程式開啟後，使用者應該可以很直覺的了解介面具備的功能與如何操作，減少學習的時間。現在的使用者時間寶貴，不希望閱讀使用手冊或教學文件。
- 讓使用者的操作步驟降到最低，減少輸入或是智能輔助輸入。例如透過GPS將地理位置資訊自動帶入特定欄位等。
- 手指操作取代鍵盤滑鼠：傳統應用程式或是網頁，多以mouse over、mouse out、double click等滑鼠事件設計，但是在行動應用程式，則要改變設計邏輯，因為使用者多以手指操作，因此多會使用下拉、上推、左往右滑動、右往左滑動等手勢進行操作。



軟體專案管理

專案管理知識領域-整合管理

專案管理知識領域-範疇管理

專案管理知識領域-時間管理

專案管理知識領域-成本管理

專案管理知識領域-品質管理

專案管理知識領域-人資管理

專案管理知識領域-溝通管理

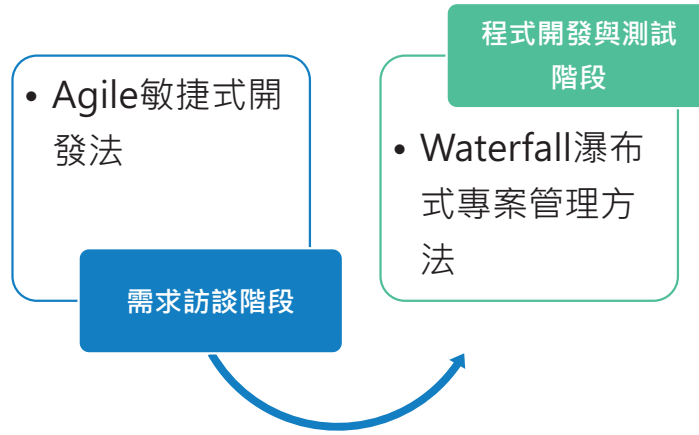
專案管理知識領域-風險管理

專案管理知識領域-採購管理

專案管理知識領域-利害關係人管理

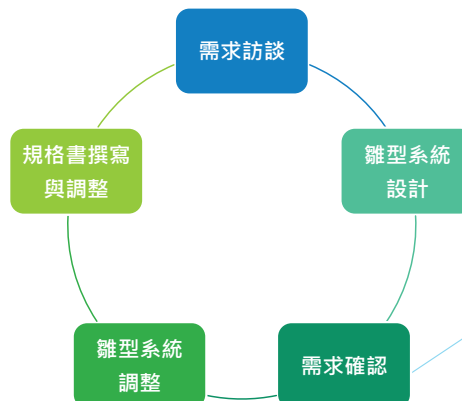
專案管理方法(1)

- ▶ 因應系統需求快速回應，以及程式穩定開發之考量，建議專案採兩階段管理方式：



專案管理方法(2)

- ▶ 需求訪談/系統分析/需求確認階段
 - ▶ 結合Lean UX的開發管理，針對使用者需求快速做回應，並透過迭代的雛型系統設計和需求確認會議，調整修正雛型系統和程式開發架構，以期能滿足使用者需求，並藉由多次的訪談和討論，規劃符合Mobile APP UX設計原則的系統介面和功能。
 - ▶ 於需求訪談階段依序做各功能的需求釐清，並設計雛型系統做畫面確認。



專案管理方法(3)

▶ 系統開發、測試階段

- ▶ 採Waterfall專案管理方式，按部就班將需求訪談階段所確認的項目逐一執行。



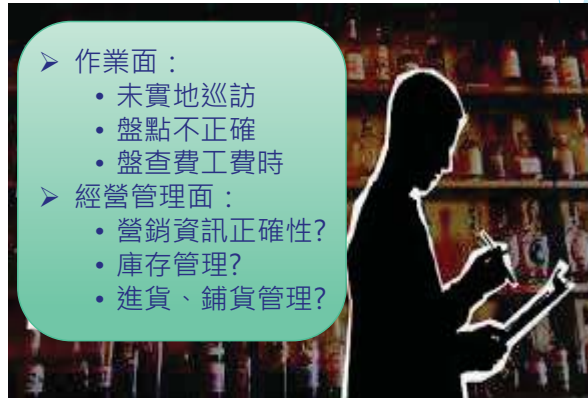
行動應用需求面向



科技零售情境分享



- 個案企業基本資料：
 - 全省 2 萬個店點 ~ 含便利商店及傳統雜貨店
 - 店頭查訪人員數目：1千多人
 - 訪查員工作→
 - ✓ 查訪銷售狀況
 - ✓ 店促活動確認
 - ✓ 店頭庫存盤點
 - ✓ 店家進貨確認
- 問題描述：



- 作業面：
 - 未實地巡訪
 - 盤點不正確
 - 盤查費工費時
- 經營管理面：
 - 營銷資訊正確性？
 - 庫存管理？
 - 進貨、鋪貨管理？

最佳實例：行動查訪



階段一：作業行動化→提昇訪查人員工作效率



階段二：企業流程改造→角色重定位



階段三：強化經營生態圈→一員一企業



