行動應用App安全開發說明會
iOS基礎概念

指導單位：經濟部工業局

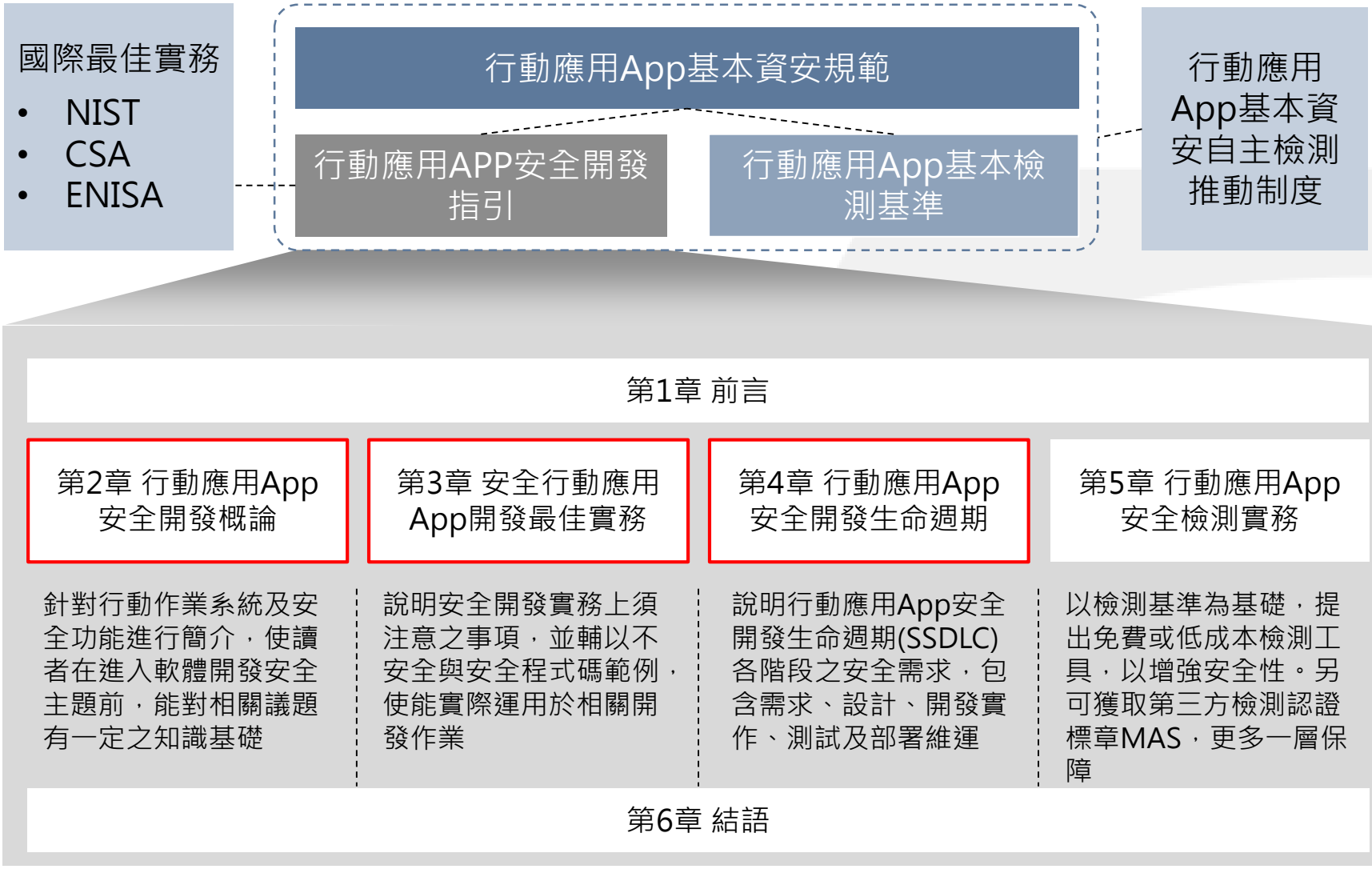
執行單位：財團法人資訊工業策進會、中華民國資訊軟體協會

協辦單位：台北市電腦公會、中華民國資訊安全學會

課程內容簡介

- iOS行動應用App安全基礎概論
  - 簡介iOS行動作業系統及安全功能，及在開發時可能伴隨的風險。
  - 介紹國外不同政府機關及團體組織對於行動裝置之安全規範及標準。
  - 介紹我國「行動應用App基本資安規範」及「行動應用App基本資安檢測基準」。
- 安全軟體開發生命週期
  - 對針對安全軟體發展生命週期(Secure Software Development Life Cycle)的5階段進行說明。
- 安全行動應用設計最佳實務
  - 挑選幾例最佳實務進行說明。

行動應用APP安全開發指引架構簡介



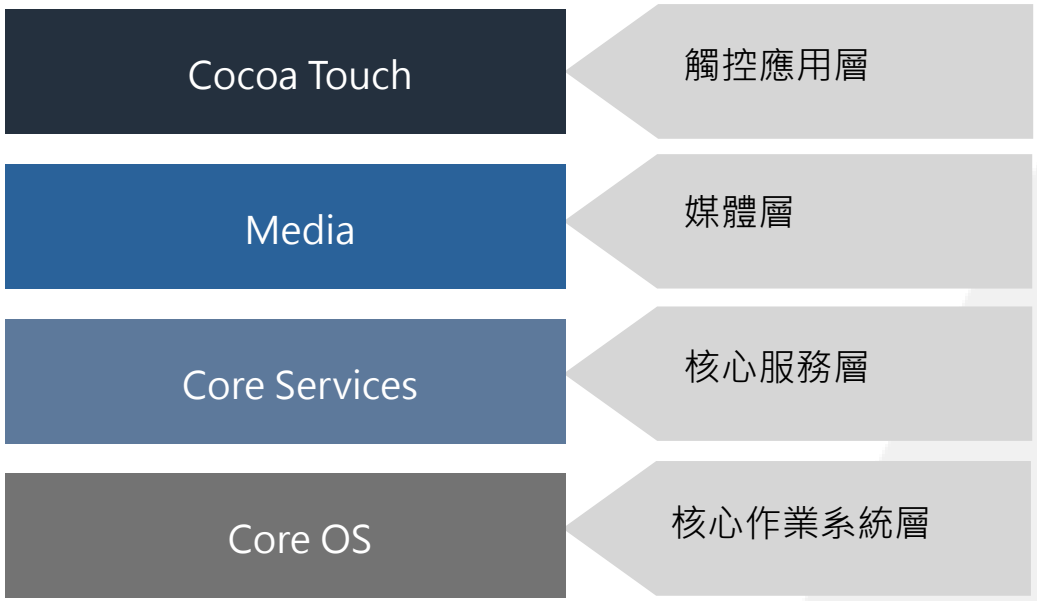
課程大綱

第1單元	行動應用App安全基礎概論	1.0小時
	行動作業系統及安全功能簡介	
	行動應用App開發環境	
	行動應用App資安風險議題	
	各國行動應用App安全開發要點簡介	
	我國行動應用App基本資安規範簡介	
	我國行動應用App基本資安檢測基準及自主檢驗制度簡介	
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

iOS作業系統簡介(1/3)

Apple公司為行動裝置所開發的封閉式作業系統，不支援任何非Apple的硬體裝置。

iOS系統層級



參考來源：developer.Apple.com

iOS 使用基於與Apple麥金塔電腦不同的 ARM 架構的 64 位元或 32 位元之中央處理器，使用以 Darwin 作為基礎的系統架構層次。

iOS作業系統簡介(2/3)

- **使用者介面**
  - 設計特點：iOS 使用者介面能使用按鍵及多點觸控(Multi-Touch)對裝置進行控制。
  - 螢幕介面：以App方格的形式呈現，亦可將多個程式放到一個資料夾中。最底部的一欄為Dock，最多可以有4個(iPad系列產品則為6個)。

- **App商店(App Store)**
  - Apple公司為旗下作業系統所建立和維護的數位應用程式發布平台，允許用戶從 iTunes Store 瀏覽和下載一些由 iOS SDK 或 Mac SDK開發的App。

iOS作業系統簡介(3/3)

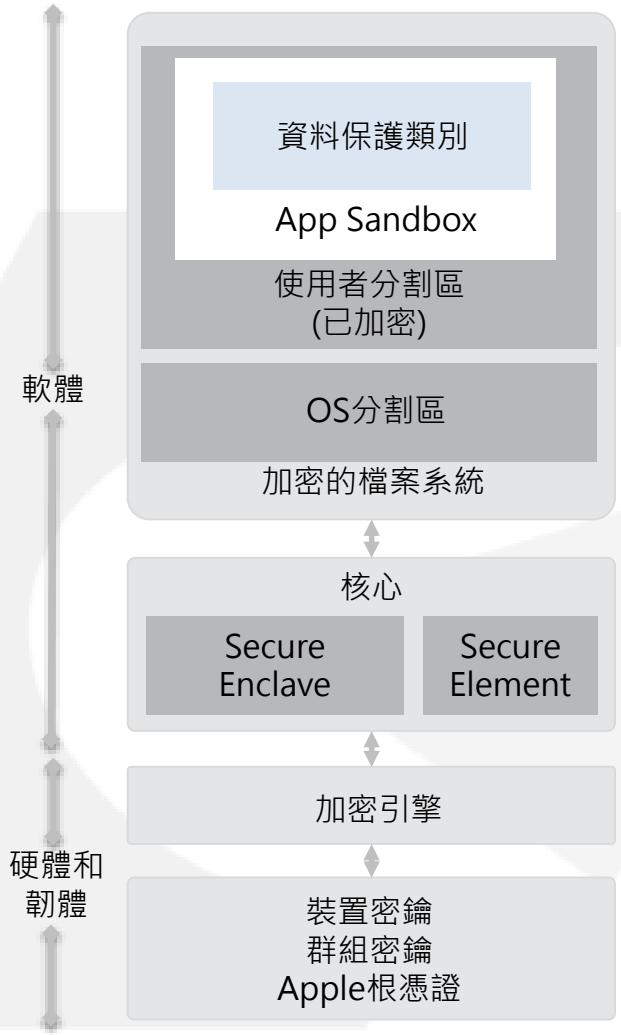
- **iOS 開發人員計劃(iOS Developer Program)**
  - 內容包括為開發人員提供開發工具、技術支援、發布資格及發布稽核等。
  - 需註冊開發人員帳號才能將開發好的App上傳商店。
- **iTunes Connect**
  - 針對每個App採實質審核，包含非法API查核及功能審核。
- **開發工具(iOS軟體開發套件)**
  - 由Apple公司為iOS設計的App開發工具包。
- **裝置越獄(iOS Jailbreaking)**
  - 用於取得iOS最高權限的一種技術手法。

iOS安全功能簡介(1/2)

- 軟體、硬體和服務在每台iOS裝置上緊密合作，一同為使用者提供最高的安全性和直接的使用者體驗。
- iOS不僅保護裝置和其中的靜態資料，同時也保護了整個生態系統，包含使用者在本機、網路上及使用重要Internet服務所執行的所有操作。

更完整內容請見：iOS安全性白皮書。
https://www.apple.com/hk/privacy/docs/iOS_Security_Guide.pdf)限於iOS8.3或以上版本

iOS的安全性架構圖



iOS安全功能簡介(2/2)

系統安全性

經整合且安全的軟硬體平台

- 安全啟動鏈
- 系統軟體授權
- Secure Enclave
- Touch ID

加密資料保護

若裝置遺失或遭竊，或有未經授權的嘗試使用或修改裝時，對使用者資料進行保護的架構與設計

- 硬體安全性功能
- 檔案資料保護
- 密碼
- 資料保護類別
- 鑰匙圈資料保護
- 取用Safari儲存的密碼
- Keybag
- FIPS 140-2

App安全性

經可讓App安全執行且不犧牲平台完整性的系統

- App程式碼簽署
- 執行階段程序安全性
- 延伸功能
- App群組
- App中的資料保護
- 配件
- HomeKit
- HealthKit
- Apple Watch

網路安全性

對傳輸中的資料提供安全認證和加密的產業標準網路通訊協定

- SSL、TLS
- VPN
- Wi-Fi
- 藍牙
- 單一登入
- AirDrop安全性

Internet服務

經Apple以網路為基礎的架構，提供傳訊、同步和備份等服務

- Apple ID
- iMessage
- FaceTime
- iCloud
- iCloud鑰匙圈
- Siri
- 連續性
- Spotlight建議

隱私控制

iOS中可用來控制「定位服務]與使用者資料取用權限的功能

- 定位服務
- 取用個人資料
- 隱私權政策

裝置控制

防止在未經授權的情況下使用裝置，以及在裝置遺失或遭竊時可進行遠端清除的方式

- 密碼保護
- iOS配對機型
- 設定強制執行
- 行動裝置管理(MDM)
- 裝置登記方案
- Apple Configurator
- 裝置限制
- 僅受監管的限制
- 遠端清除
- 尋找我的iPhone與啟用鎖定

更完整內容請見：iOS安全性白皮書。https://www.apple.com/hk/privacy/docs/iOS_Security_Guide.pdf)限於iOS8.3或以上版本

課程大綱

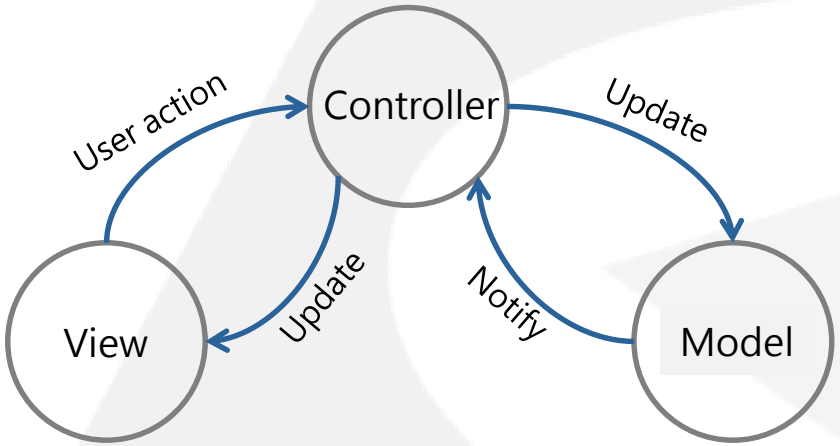
第1單元	行動應用App安全基礎概論	1.0小時
	行動作業系統及安全功能簡介	
	行動應用App開發環境	
	行動應用App資安風險議題	
	各國行動應用App安全開發要點簡介	
	我國行動應用App基本資安規範簡介	
	我國行動應用App基本資安檢測基準及自主檢驗制度簡介	
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

整合開發環境(1/2)

Xcode是Apple公司開發設計的整合開發環境(IDE)，目前最新版本為第8版，可在Mac App Store上免費下載。

MVC(Model View Controller)架構

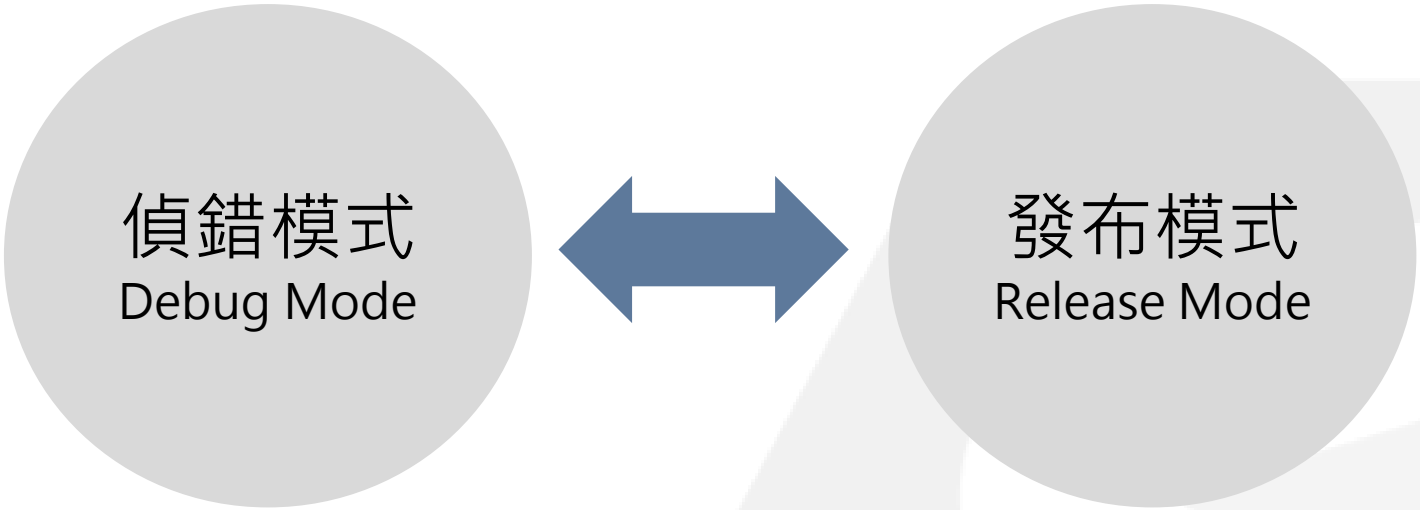
- View則負責畫面的呈現。
- Controller負責流程的控管，用來描述Model，例如某個物體要放在畫面的哪個位置。
- Model則負責處理資料。



參考來源：
https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html

整合開發環境(2/2)

偵錯(Debug)模式及發布(Release)模式



偵錯(Debug)模式可分為使用模擬器與使用實際裝置執行。

發布(Release)模式可分為Ad Hoc模式與上架模式。

以上模式都需要互相配合的裝置描述檔(Provisioning Profile)、App ID與憑證等。

綜合來說，要開發iOS App，你要準備幾件事

1. 註冊Apple ID
2. 加入iOS 開發者計劃
3. 購買Mac電腦
4. 安裝Xcode軟體
5. 使用iOS SDK
6. 學習Objective-C或Swift語言

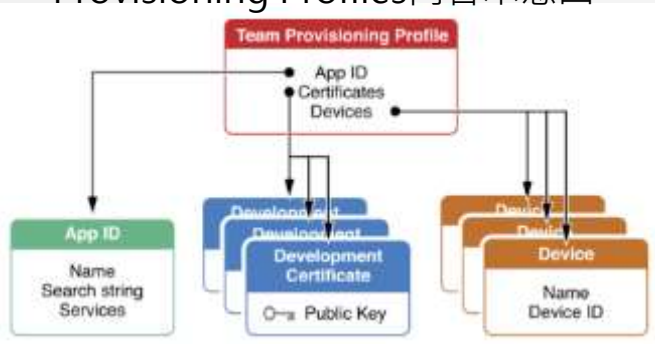
App要上架，你應該做這幾件事

STEP 1	STEP 2	STEP 3	STEP 4	STEP 5
取得帳號	**製作檔案**	**填申請單**	**上傳App**	**上架管理**
註冊一個開發者帳號並付會員費	產出P12檔案及Provisioning Profile*	送審核申請單給Apple。使用iTunes Connect，進入「我的App」，填寫App基本資料	上傳App，等待審核。可使用Xcode，透過Application Loader上傳ipa檔	審核完畢成功上架後，可採用iTunes Connect裡的功能，了解App的下載量及銷售熱度

什麼是裝置描述檔(Provisioning Profile)？

- 在Xcode或其他開發環境進行測試或正式發布時，所需使用的簽章憑證，用來確認App的合法性。
- 內容包含App ID、憑證(Certificates)及裝置(Devices)。該描述檔有2類，分別為開發(Development)及發布(Distribution)。後者又分為App Store及Ad Hoc。

Provisioning Profiles內容示意圖



資料來源：
https://developer.apple.com/library/content/documentation/IDEs/Conceptual/AppStoreDistributionTutorial/CreatingYourTeamProvisioningProfile/CreatingYourTeamProvisioningProfile.html

Apple Store的上架流程之名詞補充

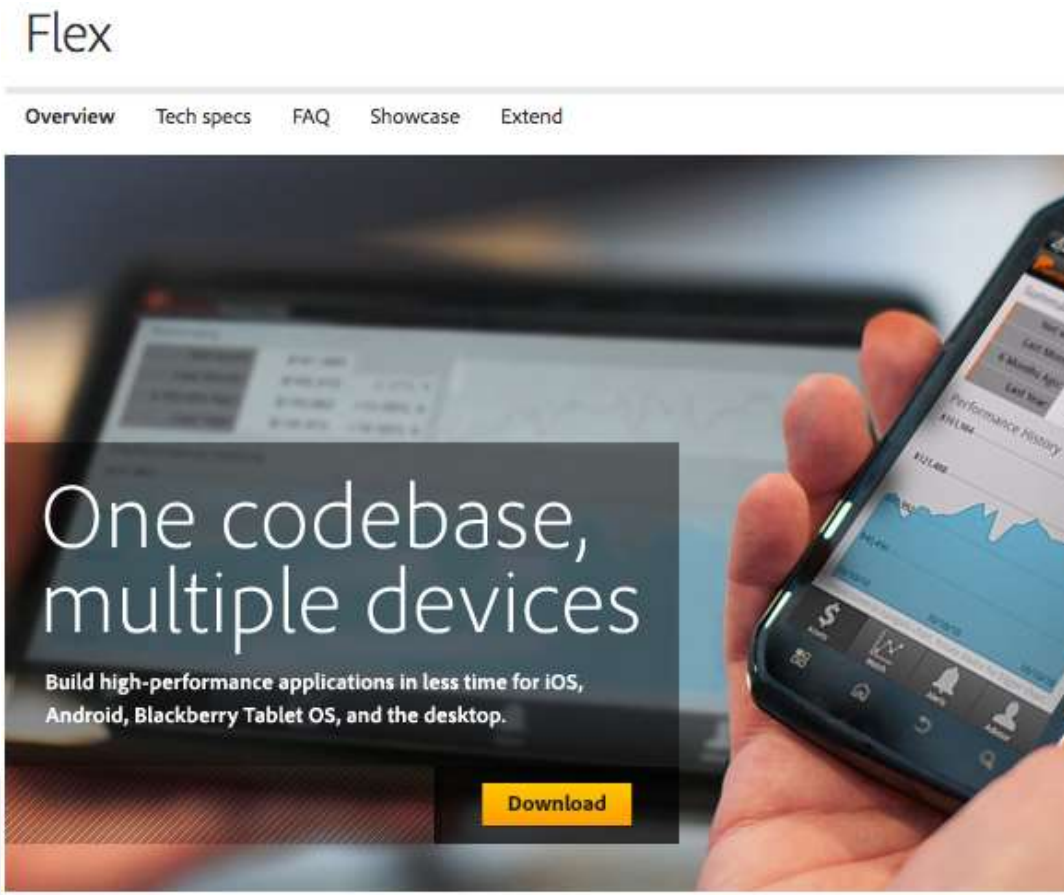
iTunes Connect是什麼？有什麼功能？

 我的App	列出所有建立過的App清單，以管理App內容(售價/版本/開放地區/描述/說明)、更新、上下架及IAP付費(InAppPurchase)等維護。
 App分析	提供App/App套裝之曝光次數、購買次數、銷售額、執行次數及當機次數等。
 銷售與趨勢	提供帳號銷售之報表分析系統。可依區域、裝置、類別、內容類型、交易類型、完成狀態及版本等。
 付款與財務報告	提供帳號之每月銷售金額、稅額及餘額報表。
 協議、稅務與銀行業務	用於管理與Apple之間的合約、提供與App開發人員付款和扣繳稅款相關的必要財務資訊及追蹤iTunes協議的狀態。
 使用者與職能	可賦予不同職能給不同的使用者，讓這些使用者分別檢視/管理各項工作。
 資源與輔助說明	提供各項資源，並輔助說明，以利相關人員使用與開發。

第三方開發商(1/3)

Flex Builder(Adobe)

Flex Builder是Adobe公司推出的軟體開發IDE產品，原本只能開發Flash程式，自從2011年推出的4.5版，就有支援App開發，可直接於該軟體開發iOS與Android平台的手機App。



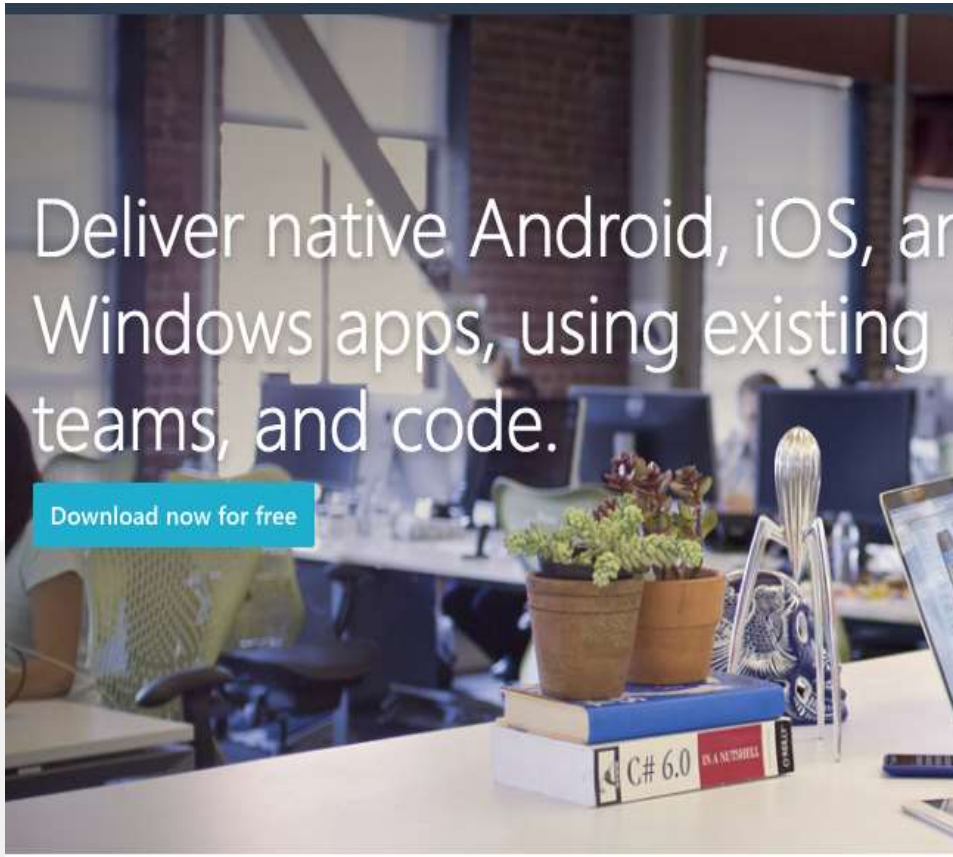
優點	可直接使用Flex元件開發，將App進行上傳
缺點	•使用者介面非原生，使用者適應不佳 •App程式底層過大 •與原生的介面、應用差異度過大，非直接使用原生Framework •非原生開發：可能有潛在安全風險無法對安全性修補。

第三方開發商(2/3)

Xamarin Stduio (Microsoft)

- 原本在Novell的Mono專案團隊獨立成立一家公司，2011年正式將名稱改為Xamarin。
- 提供跨平台開發能力，開發人員透過Xamarin開發工具與程式語言，即可開發出iOS、Android與Windows 等平台的原生(Native) App 應用程式，不須個別使用各平台的開發工具與程式語言。

優點	• 可直接使用C#語言進行開發跨平台App(含iOS/Android/Microsoft Phone)。 • Xamarin Forms：可寫一套原始碼，應用於跨平台App。 • 支援原生開發人員功能：可寫作原生App。
缺點	▪ 使用者介面非原生、使用者適應不佳。 ▪ App程式底層過大。 ▪ 雖號稱支援原生者開發功能，但程度與版本不一。 ▪ 仍非完全原生，可能有潛在安全風險、無法對安全性修補。



第三方開發商(3/3)

PhoneGap (Adobe Cordova)

PhoneGap是一款開放原始碼的行動裝置開發框架，旨在讓開發人員使用HTML、Javascript、CSS等Web APIs開發跨平臺的行動裝置App。原本由Nitobi公司開發，現在由Adobe Systems擁有。

優點	• 可直接使用Html 5為基礎的語言進行開發跨平台App。 • Html 5 Base：可寫一套原始碼，應用於跨平台App。
缺點	• 使用者介面非原生、使用者適應不佳(類似使用網頁瀏覽器)。 • 使用網頁技術，反應取決於網路等效能，使用者經驗較差。 • 安全問題高，開發人員底層技術掌握度低。 • 仍原生，可能有潛在安全風險、無法對安全性修補。

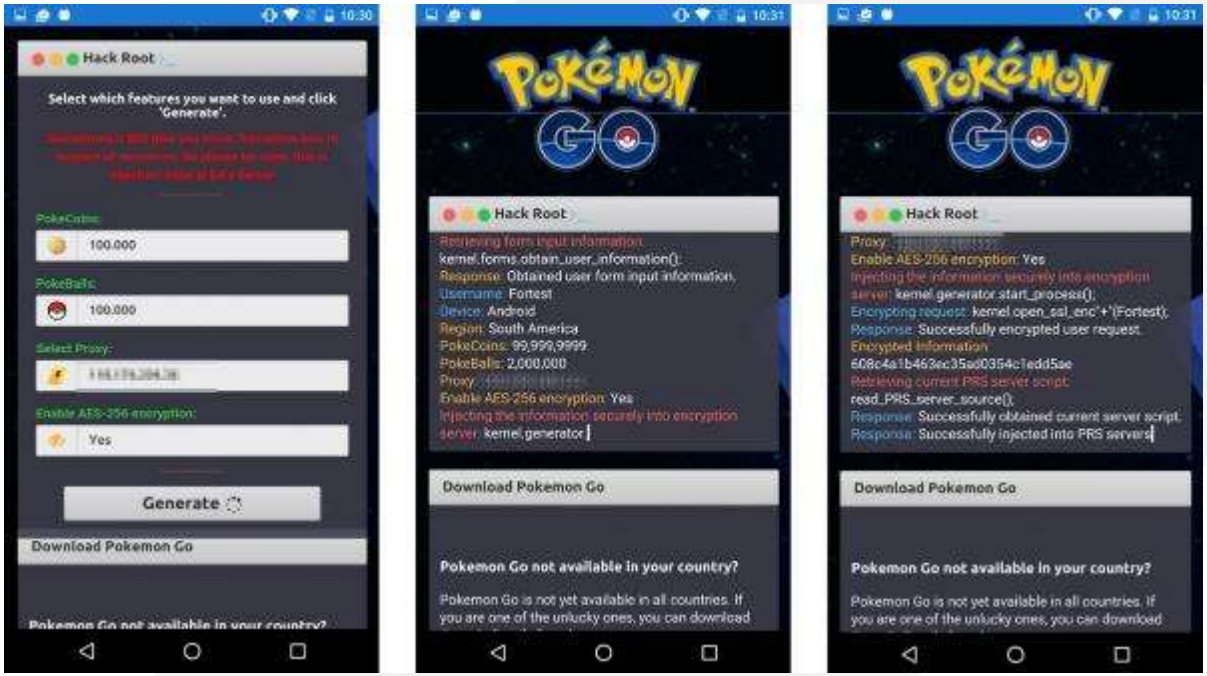
課程大綱

第1單元	行動應用App安全基礎概論	1.0小時
	行動作業系統及安全功能簡介	
	行動應用App開發環境	
	行動應用App資安風險議題	
	各國行動應用App安全開發要點簡介	
	我國行動應用App基本資安規範簡介	
	我國行動應用App基本資安檢測基準及自主檢驗制度簡介	
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

冒牌Pokémon Go

App開發者，常限於開發時間緊迫、人力成本緊縮及未考慮到使用情境的安全問題，或是開發者從未接受安全程式設計實務訓練，寫出有漏洞程式。

- 夾帶木馬程式
- 暗藏著惡意的彈出視窗或廣告



資料來源：趨勢科技，http://blog.trendmicro.com.tw/?cat=15 (105/8/11)

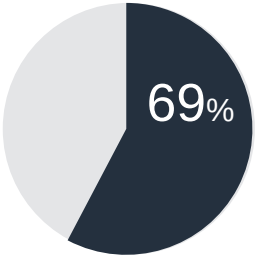
OWASP十大行動安全風險

國際知名開放軟體安全計畫(Open Web Application Security Project, OWASP)
發布2016年行動應用App前10大安全風險侯選清單

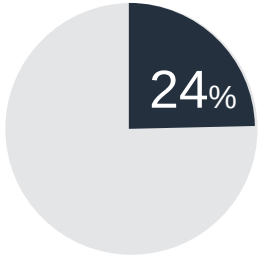
1. M1不當使用行動作業平台(M1 - Improper Platform Usage)
2. M2不安全資料儲存(M2-Insecure Data Storage)
3. M3-不安全通訊(M3-Insecure Communication)
4. M4-不安全身分認證(M4-Insecure Authentication)
5. M5-不足夠的加密(M5-Insufficient Cryptography)
6. M6-不安全授權(M6-Insecure Authorization)
7. M7-用戶端程式碼品質(M7-Client Code Quality)
8. M8-程式碼篡改(M8-Code Tampering)
9. M9-逆向工程(M9-Reverse Engineering)
10. M10-多餘的功能(M10-Extraneous Functionality)

惡意行動應用App

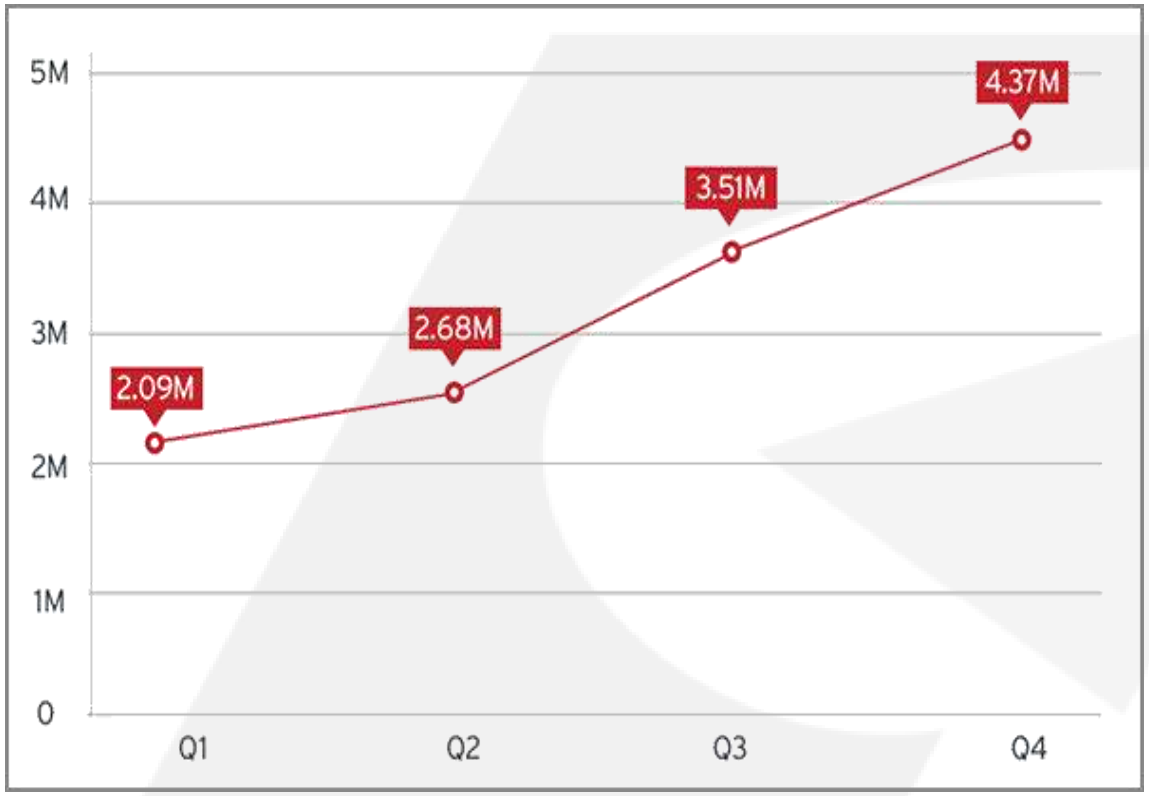
有極大數量的行動應用App帶有惡意或隱密蒐集使用者敏感性資料的意圖



69%的行動裝置威脅都屬於逾權廣告程式



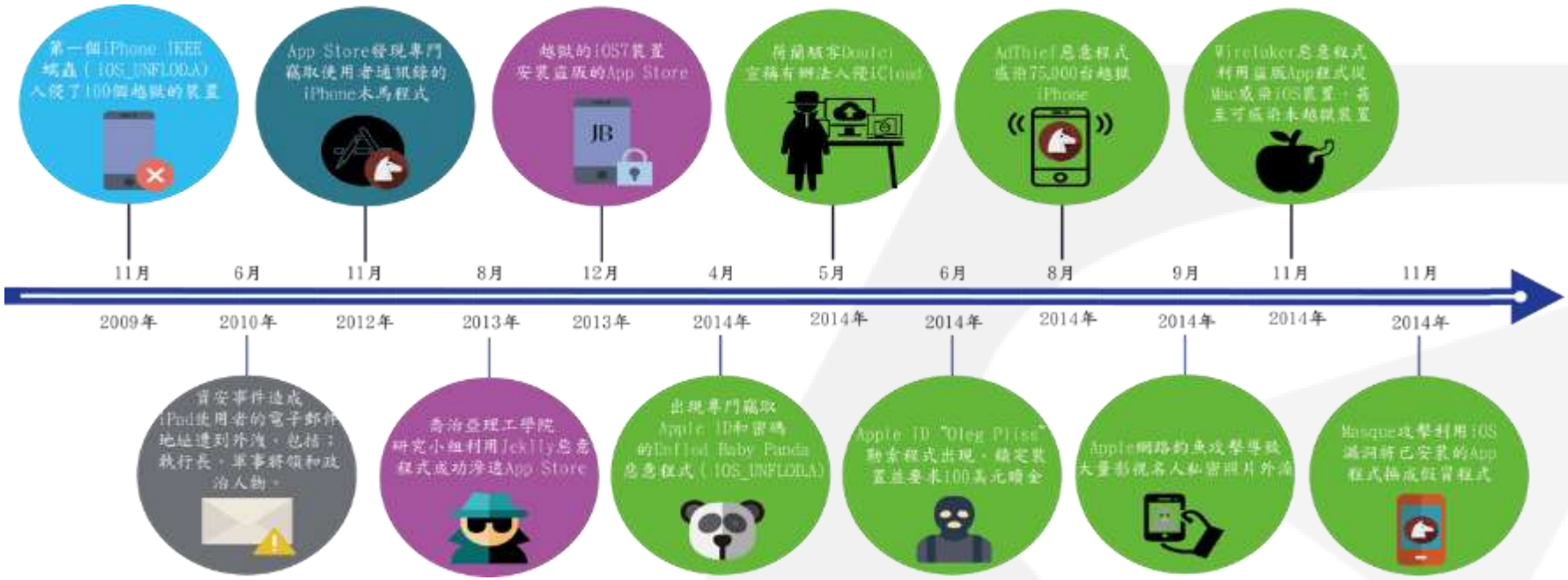
24%屬高費率服務盜用程式(PSA)



2014年惡意及高風險的App程式成長趨勢

針對行動應用平台攻擊事件

iOS資安事件時序表(2009至2014年)



資料來源：趨勢科技

行動應用App常見威脅

趨勢科技將行動應用App常見威脅

1 行動裝置勒索程式

加密使用者資料，勒索使用者換取解密金鑰。

2 越權廣告程式

未經使用者同意取得或獲取過多權限，取用使用者敏感性資料。

3 費率服務盜用程式

在背景偷用一些高費率服務，讓使用者的手機帳單費用無故飆高。

4 免越獄植入木馬

竊取憑證，將木馬程式安裝在沒有越獄的裝置上。

5 有漏洞程式庫/程式開發套件

處理敏感性資料程式庫及開發工具漏洞被利用。

6 偽造的程式ID

可讓惡意程式假冒正常程式的Android FakeID漏洞。

7 內建瀏覽器跨站來源腳本存取漏洞

惡意程式可能存取正常網站所使用的資料和Cookie，取得敏感性資料。

課程大綱

第1單元	行動應用App安全基礎概論	1.0小時
	行動作業系統及安全功能簡介	
	行動應用App開發環境	
	行動應用App資安風險議題	
	各國行動應用App安全開發要點簡介	
	我國行動應用App基本資安規範簡介	
	我國行動應用App基本資安檢測基準及自主檢驗制度簡介	
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

中國大陸(China)

大陸工業與信息化部發布了一系列行動智慧裝置安全標準

行動智慧裝置安全能力設計導則 | 行動智慧裝置安全能力技術要求 | 行動智慧裝置安全能力測試方法 | 移動終端晶片安全技術要求和測試方法

YD/T 2407-2013 行動智慧裝置安全能力技術要求

- 僅制訂其基本原則：「行動智慧裝置上發生的行為和應用要符合用戶的意願 」，無規定具體的實現方法和措施。
- 作為安全開發準則參考之一：
  - 硬體安全能力要求
  - 作業系統安全能力要求
  - 週邊接口安全能力要求
  - 應用軟體安全要求
  - 使用者資料安全保護能力要求

與行動應用App安全開發相關

YD/T 2408-2013 行動智慧裝置安全能力測試方法

針對技術要求提出相對應的技術指標設計及測試方法，用於驗證行動智慧裝置是否滿足技術要求的規定。

歐盟(Europe Union, EU)

歐洲網路暨資訊安全局(ENISA)於2011年發布的「智慧型手機開發者安全開發指引(Smartphone Secure Development Guidelines for App Developers, SSDGAD)」

針對下列3個計畫進行行動應用App的安全風險評估

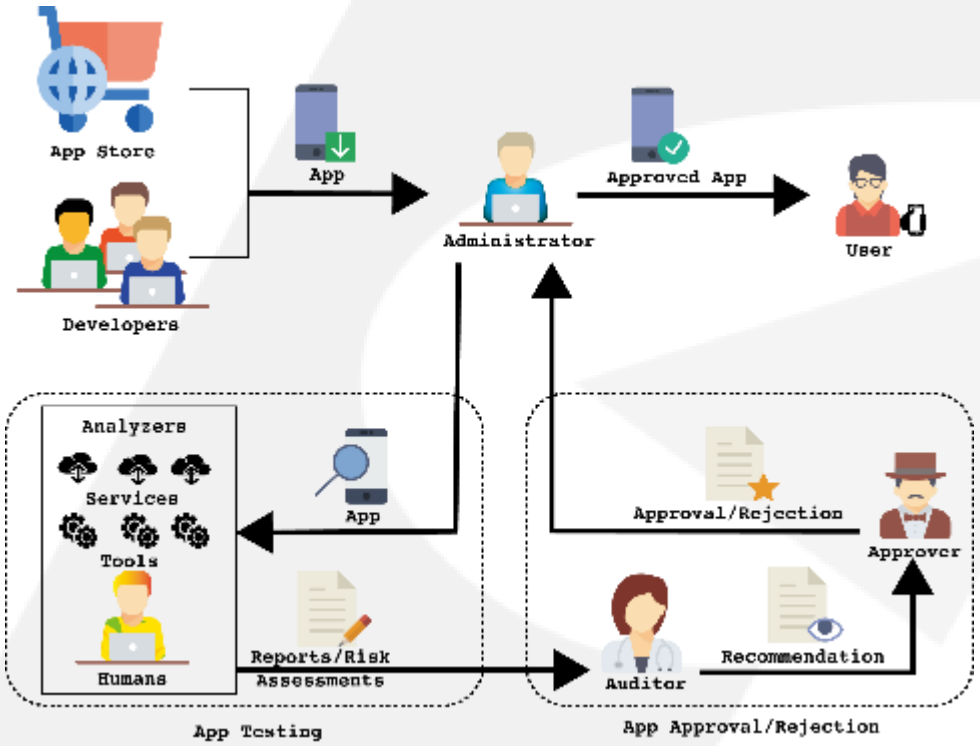
OWASP Mobile Top 10 Risks	OWASP Web Top 10 Risks	OWASP Cloud Top 10 Risks

行動應用App前10大控制措施分類	控制措施
1.在行動設備上識別和保護敏感資料	14
2.在設備上安全地處理密碼憑證	10
3.確保敏感資料在傳輸過程中受保護	7
4.正確地建置用戶認證、授權和連線管理機制	6
5.確保後端的API(服務)和平台(伺服器)的安全	5
6.確保使用第三方服務和App時的資料安全	3
7.應有機制蒐集及保留使用者所簽署的個資使用同意證明	6
8.建立機制以防止付費資源(電子錢包、簡訊、電話等)被未經授權的存取	7
9.確保得以安全的發布 /更新行動應用App	3
10.小心檢查程式碼在執行直譯時可能會發生的錯誤	4
合計	65

美國(United States of America, USA)

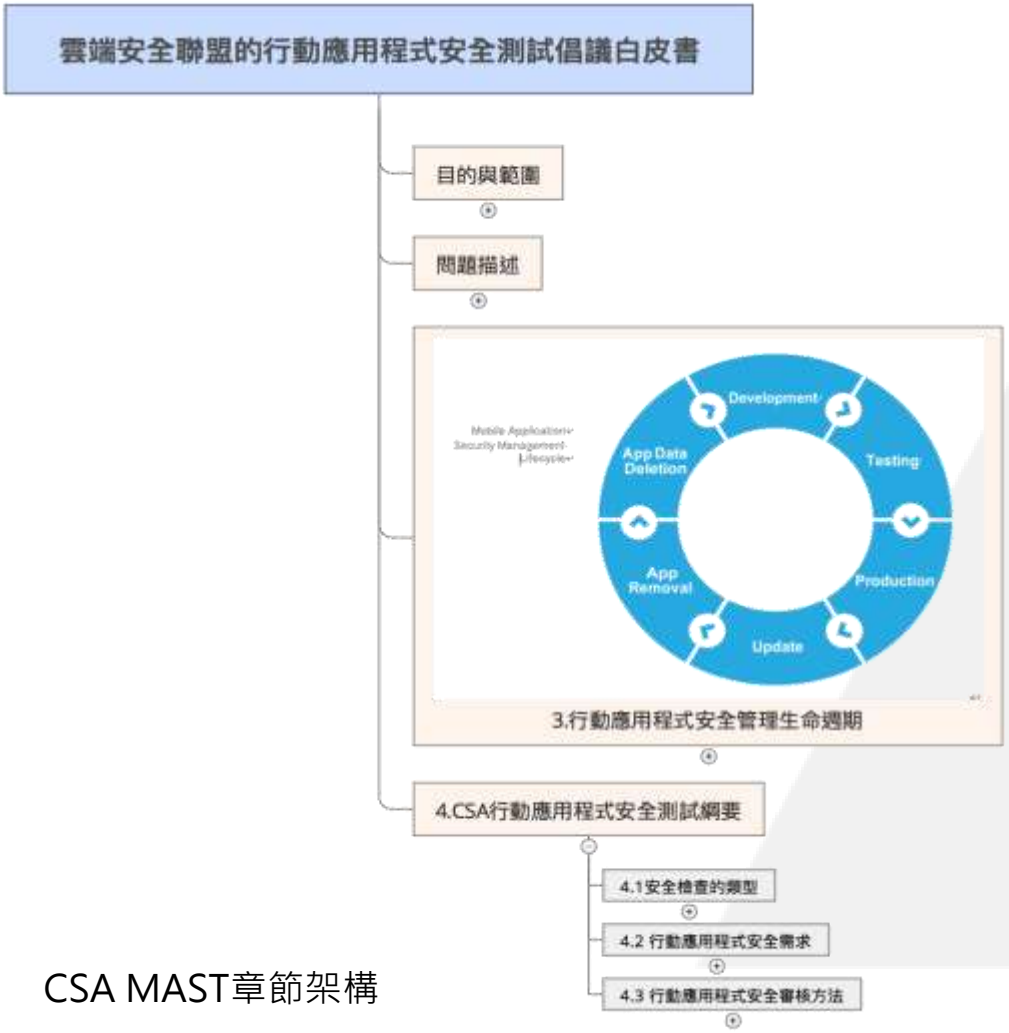
國家標準與科技機構(NIST)所發布NIST SP 800-163行動應用App安全審核(Vetting the Security of Mobile Applications)。

- 幫助企業了解App審核流程
- 協助企業規劃進行審核的整個過程
- 協助企業了解開發App的安全要求
- 協助企業了解各種App的漏洞，和測試這些漏洞的方法
- 協助企業了解該App是否適合安裝在組織的行動裝置上



行動應用App審核流程及角色　資料來源：NIST SP800-163

雲端安全聯盟的行動應用App安全測試倡議白皮書



CSA MAST章節架構

- 由雲端安全聯盟(Cloud Security Alliance, CSA)提出的行動應用APP安全測試(Mobile Application Security Testing, MAST)白皮書，旨在協助企業或個人在使用行動應用App時，可降低其潛在的風險及資安威脅。
- MAST定義了一個行動應用App的安全開發框架，以處理使用者隱私與資安的議題。

課程大綱

第1單元	行動應用App安全基礎概論	1.0小時
	行動作業系統及安全功能簡介	
	行動應用App開發環境	
	行動應用App資安風險議題	
	各國行動應用App安全開發要點簡介	
	我國行動應用App基本資安規範簡介	
	我國行動應用App基本資安檢測基準及自主檢驗制度簡介	
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

行動應用App基本資安規範

- 於104年由經濟部工業局委託資策會研議，供業界於開發行動應用App時的自主遵循參考。
- 為非強制性規定，主要目的在於提升我國行動應用App基本安全防護能力，該規範建議從設計初始階段即導入基本資安概念，並提醒App開發者應強化資訊安全意識，逐步完善自身App全防護能力。
- 該規範列舉17項關於行動應用App開發的資安技術要求，並提出3種不同的行動應用App安全分類。

下載網址:http://www.communications.org.tw/news/policy/item/8743-0814.html

編號	資訊安全技術要求事項	安全分類		
		一	二	三
1	4.1.1.1.行動App發布	V	V	V
2	4.1.1.2.行動App更新	V	V	V
3	4.1.1.3.行動App安全性問題回報	V	V	V
4	4.1.2.1.敏感性資料蒐集	V	V	V
5	4.1.2.2.敏感性資料利用	V	V	V
6	4.1.2.3.敏感性資料儲存	V	V	V
7	4.1.2.4.敏感性資料傳輸		V	V
8	4.1.2.5.敏感性資料分享	V	V	V
9	4.1.2.6.敏感性資料刪除	V	V	V
10	4.1.3.1.付費資源使用			V
11	4.1.3.2.付費資源控管			V
12	4.1.4.1.使用者身分認證與授權		V	V
13	4.1.4.2.連線管理機制		V	V
14	4.1.5.1.防範惡意程式碼與避免資訊安全漏洞	V	V	V
15	4.1.5.2.行動App完整性			V
16	4.1.5.3.函式庫引用安全	V	V	V
17	4.1.5.4.使用者輸入驗證	V	V	V

安全分類
第一類：純功能性。
第二類：具認證功能與連網行為。
第三類：具交易功能(包括認證功能及連網行為)。

課程大綱

第1單元	行動應用App安全基礎概論	1.0小時
	行動作業系統及安全功能簡介	
	行動應用App開發環境	
	行動應用App資安風險議題	
	各國行動應用App安全開發要點簡介	
	我國行動應用App基本資安規範簡介	
	我國行動應用App基本資安檢測基準及自主檢驗制度簡介	
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

行動應用App基本資安檢測基準

以「行動應用App基本資安規範」中對行動應用App的安全分類為主，再參考OWASP之Mobile Security Project-Top Ten Mobile Risks與NIST SP 800-163 Vetting the Security of Mobile Applications，評估及審驗相關行動應用App之風險項目，藉此制定安全檢測項目並規劃提出各項目之細項檢查事項、執行條件與預期結果等。

亦可作為
- 第三方檢測機構進行檢測之參考基準
- 開發者針對App安全進行檢測之依據

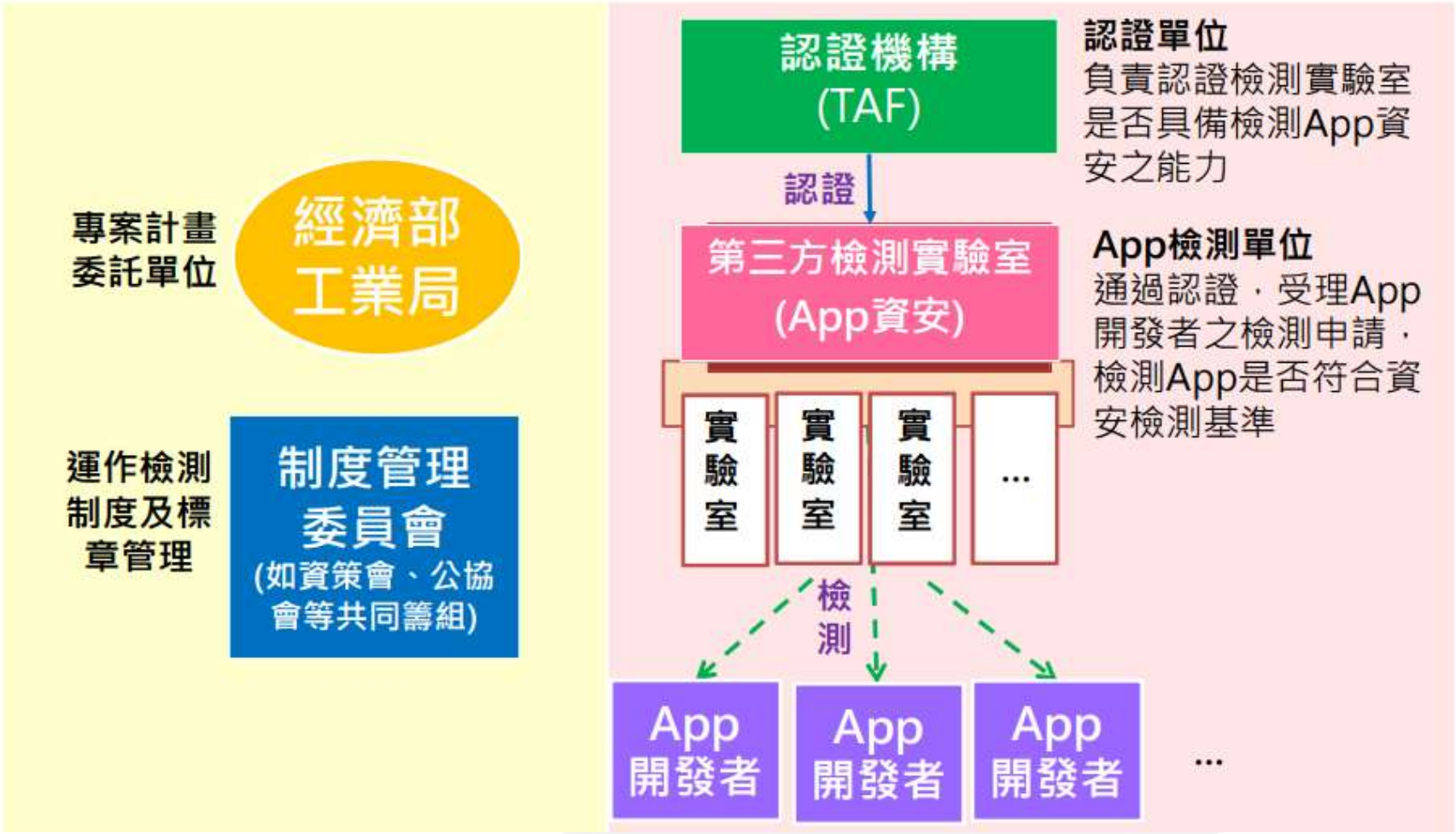
經濟部工業局行動應用App安全基本規範

下載網址:https://www.communications.org.tw/phocadownloadpap/app/v2.0_1.pdf

行動應用App基本資安檢測基準-文件架構

資料來源：經濟部工業局行動應用App基本資安檢測及制度說明簡報

行動應用App基本資安自主檢測推動制度-運作架構



資料來源：經濟部工業局行動應用App基本資安檢測及制度說明簡報

行動應用App基本資安自主檢測推動制度

課程大綱

第1單元	行動應用App安全基礎概論	1.0小時
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

行動應用App安全開發生命週期方法論

常見安全開發生命週期(SSDLC)方法論

- Cigital的Touchpoint
- Microsoft的Security Development Lifecycle(SDL)
- OWASP的Security Assurance Maturity Model(SAMM)

方法論比較摘述請詳見附件1「安全軟體開發生命週期方法論比較」

Cigital Touchpoint方法論

資料來源：
www.cigital.com

安全軟體生命週期比較

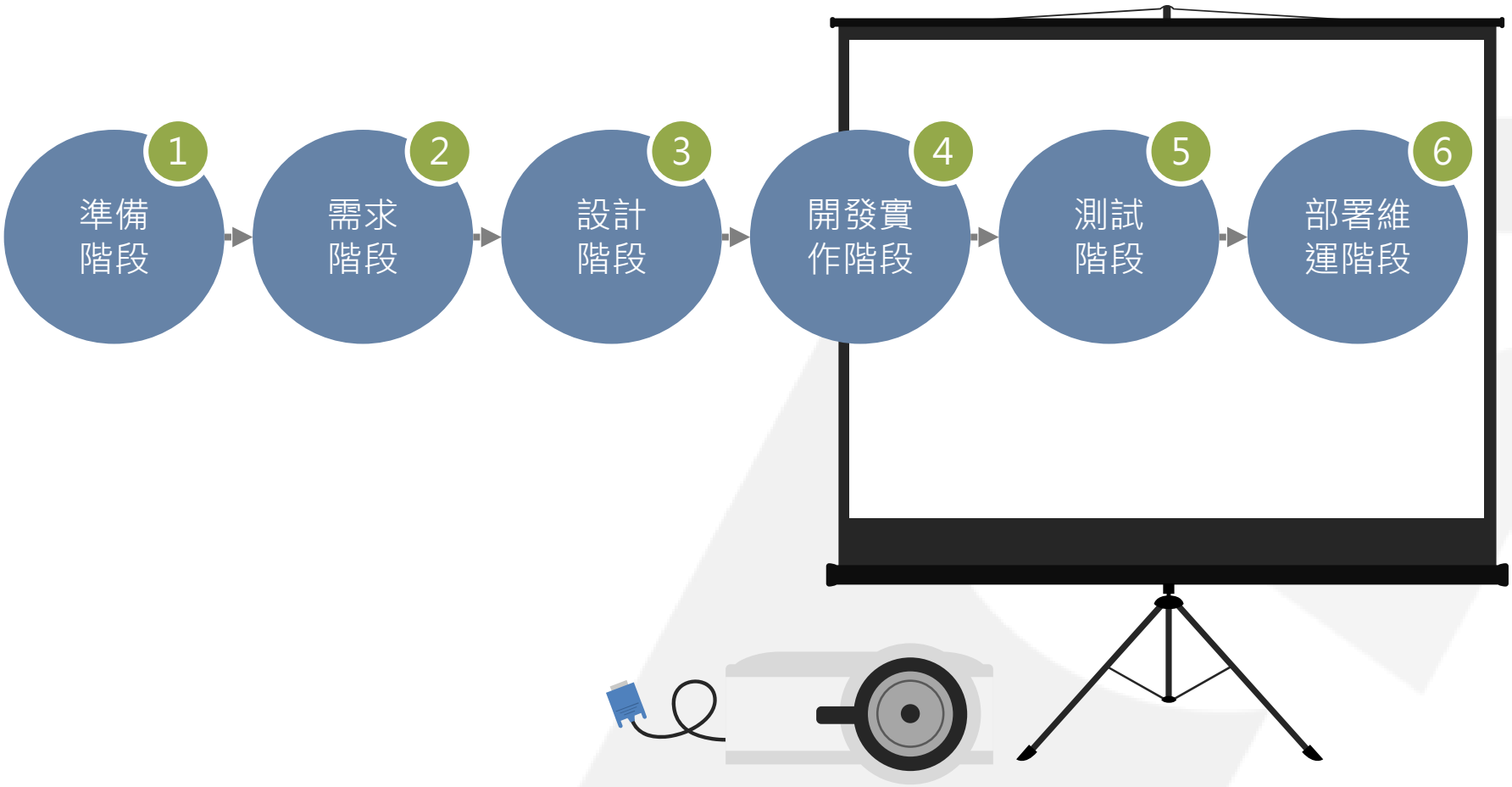
以TouchPoint方法論為例

參考Software Security Building Security In, Gary McGraw, Cigital CTO

加入敏捷式開發概念

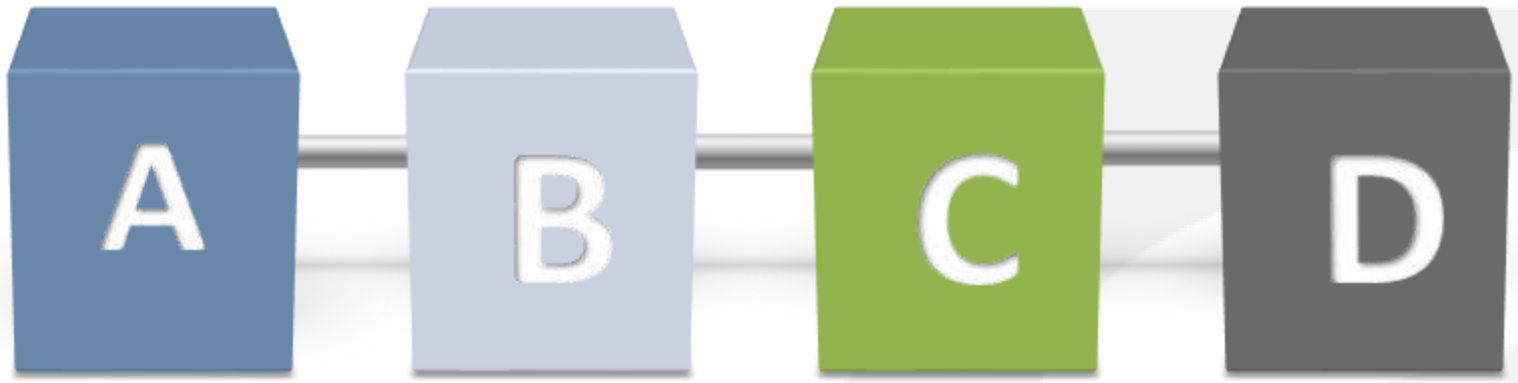
參考Software Security Building Security In，Gary McGraw,Cigital CTO & http://www.screenmedia.co.uk/blog/2014/08/what-is-agile-development-a-brief-introduction/

行動應用App安全開發生命週期各階段



1.準備階段

在行動應用App專案正式開始之前，除了準備開發環境外，還需要準備什麼？



建立安全開發流程

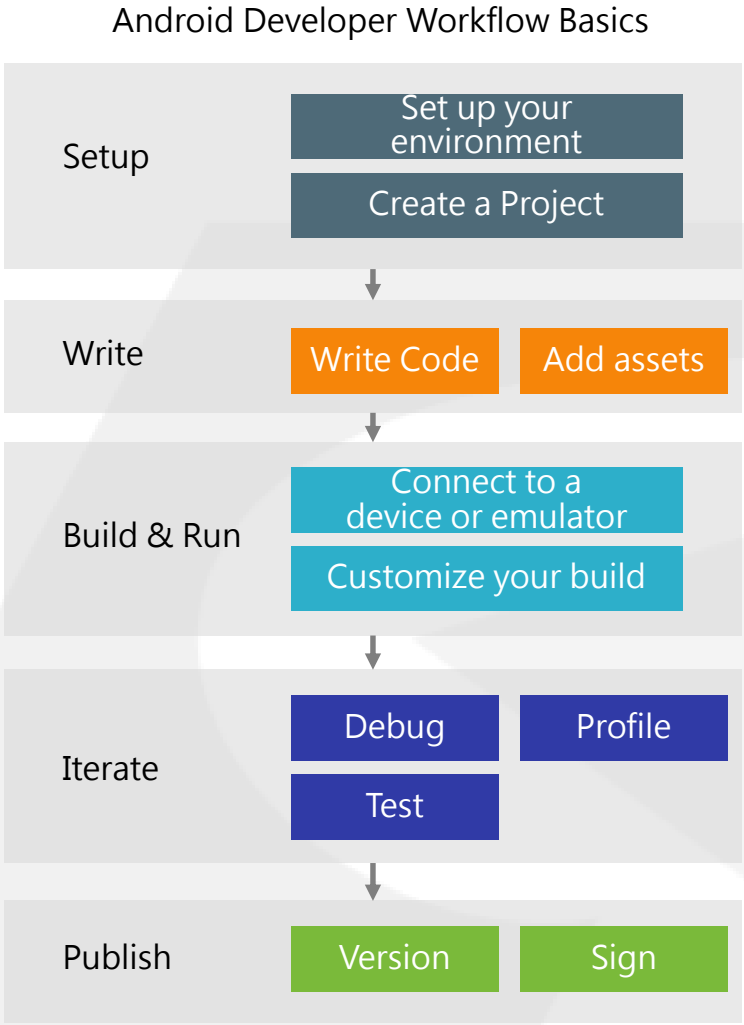
進行安全開發實務教育訓練

測試第三方API/函式庫

建立原始碼安全控管機制

A.建立安全開發流程

- 中大型的開發團隊
  - 採用傳統SDLC
    - 企業中應有完整的「應用系統開發管理規範」，包含需求單、系統分析/設計文件、程式規格書、測試報告、上線/過版申請單等文件
  - 採用敏捷式開發
    - 可運用Trello、Jira等專案及問題追蹤工具，建立專案團隊的溝通管道及確保軟體開發議題能被有效解決。
- 個人開發者
  - 可參考如Android建議的開發流程。



資料來源：Android Developer

B.執行安全開發教育訓練

行動應用App開發相關人員每年應至少參加1次以上的內部或外部訓練，範圍包含：

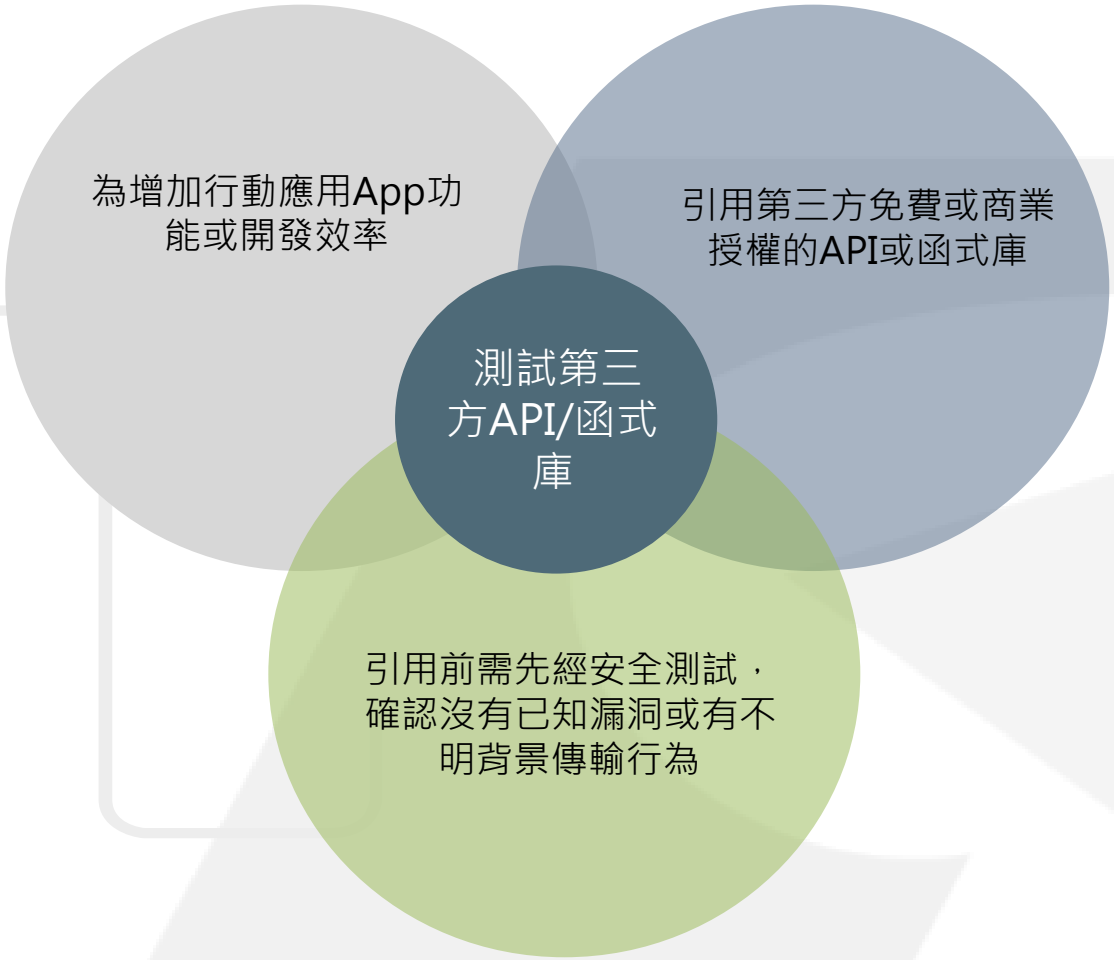
- 安全的軟體設計
- 安全的程式開發
- 安全的軟體測試
- 行動應用平台最新安全機制

技術要求 / SSDLC	4.1. 行動應用程式資訊安全技術要求事項					4.2. 伺服端資訊安全技術要求事項
	4.1.1. 行動應用程式發布安全	4.1.2. 敏感性資料保護	4.1.3. 付費資源控管安全	4.1.4. 身分認證、授權與連線管理安全	4.1.5. 行動應用程式碼安全	本規範旨在針對行動應用程式安全提出基本資訊安全要求，如行動應用程式涉及伺服端之資訊安全需求，建議應由業者自我宣告或切結其伺服端資訊安全防護與管理措施，或對於其伺服端服務之資訊安全防護與管理，出具第三方檢測通過證明。
A.需求階段	N/A	4.1.2.1. 敏感性資料蒐集 4.1.2.2. 敏感性資料利用 4.1.2.3. 敏感性資料儲存 4.1.2.4. 敏感性資料傳輸 4.1.2.5. 敏感性資料分享 4.1.2.6. 敏感性資料刪除	4.1.3.1. 付費資源使用 4.1.3.2. 付費資源控管	N/A	N/A	
B.設計階段	N/A			4.1.4.1. 使用者身分認證與授權 4.1.4.2. 連線管理機制	4.1.5.1. 防範惡意程式碼與避免資訊安全漏洞 4.1.5.2. 行動應用程式完整性 4.1.5.4. 使用者輸入驗證	
C.開發實作階段	N/A					
D.測試階段	N/A					
E.部署維運階段	4.1.1.1. 行動應用程式發布 4.1.1.2. 行動應用程式更新 4.1.1.3. 行動應用程式安全性問題回報	N/A	N/A	N/A	4.1.5.3. 函式庫引用安全	

我國行動應用App基本資安規範第4章技術要求與SSDLC對應關係

C.測試第三方API/函式庫

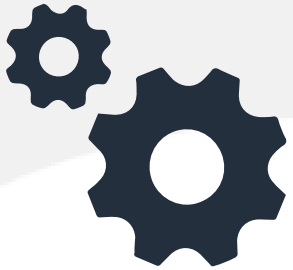
引用不安全第三方API/函式庫，造成資安漏洞，像2014年被發現OpenSSL heart bleed漏洞，影響層面就非常廣泛。



D.建立安全程式碼控管機制

經安全檢測的原始碼或API可以重複引用，以降低行動應用App漏洞發生機率。

下一步則要建立安全原始程式碼控管機制，如權限控管、簽入簽出流程、版本控管。



透過版本控管工具及變更管理機制，確保不會誤用不安全的程式碼。

2.需求階段

資安需求蒐集與分析

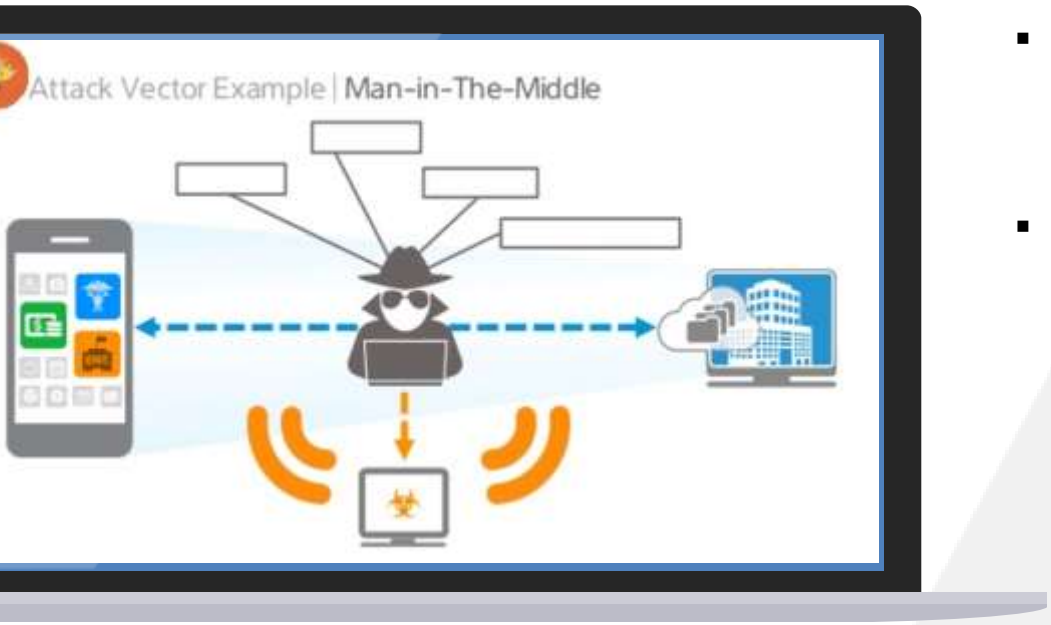
- 由PM及SA針對行動應用App整體安全需求進行評估與分析，包括識別作業系統及軟體可能面臨之安全風險，並完成安全與隱私風險評鑑，了解各種安全威脅與隱私風險狀況，定義正常與錯誤使用(濫用)案例。
- 行動應用App的資安需求的制訂，其重要性應等同功能及效能需求的制訂。
- 資安需求可包含：

- 為使開發人員(擔任PM/SA/IS角色)執行需求階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「需求階段安全檢核表」，詳見該指引附件2。

A.定義安全需求

- 是否有蒐集、處理及利用個人資料或其他敏感資料？
- 是否有來自產業的資料安全規範，如PCI DSS？
- 是否有上架當地國家資料隱私相關法規，如歐盟隱私資料保護指令？
- 軟體中敏感資料在其生命週期中的保護需求為何？
- 軟體中的敏感性資料是否有接收自或傳送至第三方的需求？
- 是否涉及付費或金流機制？
- 屬「行動應用App基本資安規範」的那一個安全分類？
- 使用對象為一般不特定消費者，還是企業內部員工？

B.風險分析



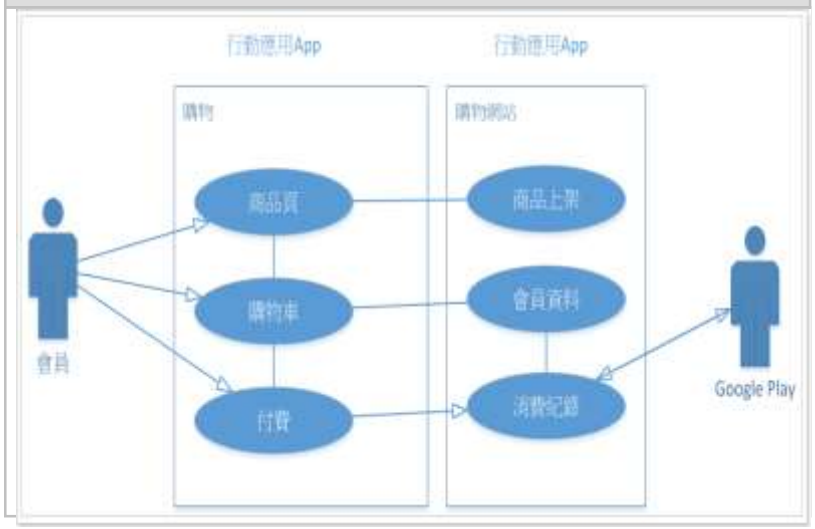
- 由PM/SA/IS與需求單位共同討論行動應用App的使用情境，以攻擊者的角度識別可能影響行動應用App與後端系統的威脅，並進行評估。
- 識別與評估威脅後，執行定性的風險分析。再對風險進行分級，對高風險事項選用適當的控制措施，以利快速形成行動應用App開發專案的安全需求重點。

C.定義正常使用與錯誤使用案例

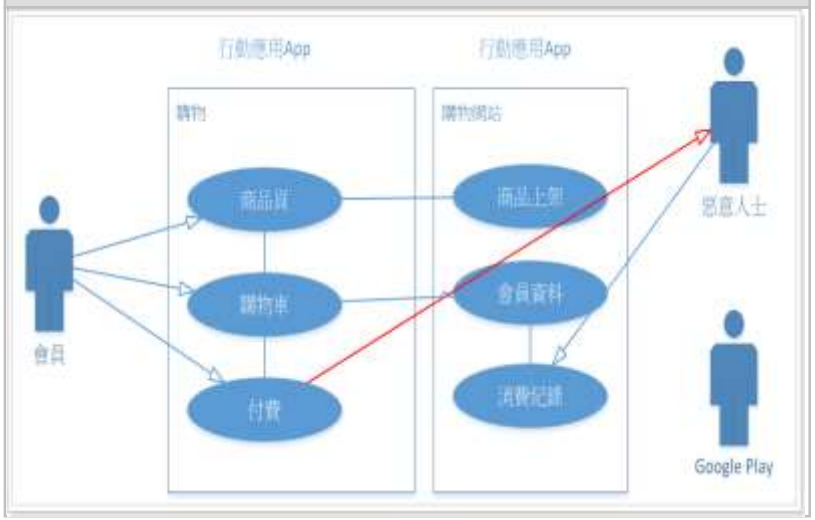
建立使用案例的目的在於將需求單位未於訪談時表達的潛在需求具體化，可透過：

- 需求(Requirements)
- 分析與設計(Analysis and Design)
- 實作(Implementation)
- 測試(Testing)

正常使用案例圖



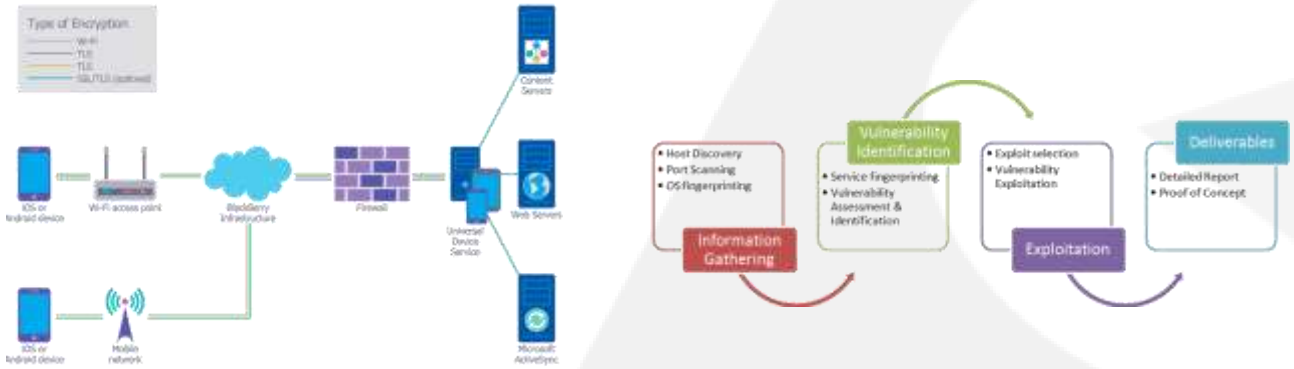
誤用案例圖



3.設計階段

- 安全與隱私防護功能應及早於系統設計初期納入，以避免於後期納入導致成本大幅的增加。
- 系統設計人員應詳細描述資訊安全的實作方法，設計過程中可與資料庫管理員及伺服器維運人員討論架構設計的可行性與安全性，包括：威脅建模、限制非必要服務、最小權限及縱深防禦等。
- 可依據「行動應用App基本資安檢測基準」建立查核點。
- 本階段應實行下列2項安全設計活動：

A.安全架構與設計 → B.安全測試計畫

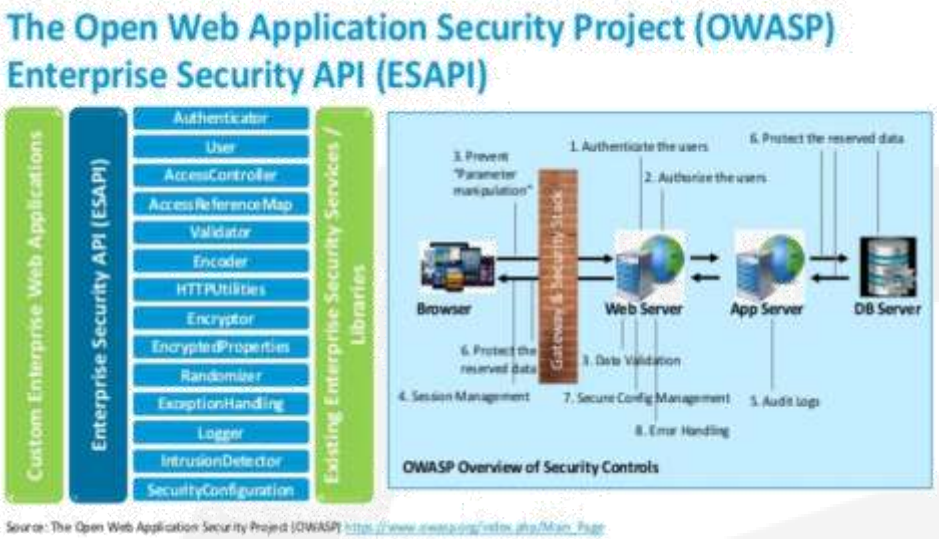


- 為使開發人員(擔任系統設計角色)執行設計階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「設計階段安全檢核表」，詳見該指引附件3。

A.安全架構與設計

1.安全架構師或系統設計人員依需求階段的產出， 設計系統的安全架構

- 應優先考量行動應用App作業系統的內建安全功能，例如：安全啟動鏈、Keychain、ATS及檔案加密系統。
- 其次參考iOS的安全開發指南建議及「行動應用App安全開發指引」已整理的安全實務，進行App本地端及伺服器間安全設計。



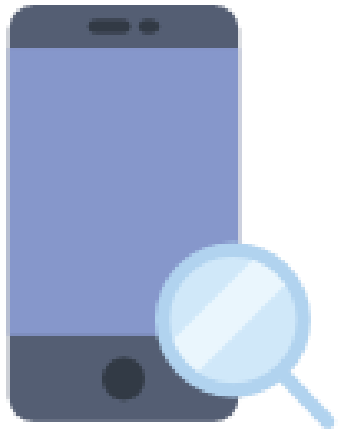
2.系統架構與安全設計，至少需包括：

- 系統架構圖
- 儲存區分配規劃
- 資料儲存安全設計
- 資料傳輸安全設計
- 安全界面設計
- 資料消除設計
- 權限設計
- App間安全通訊設計

3.對敏感性資料的保護，應以整個資料生命週期進行考量，選用符合法規及產業規範的控制措施。

B.安全測試計畫

在完成系統架構與安全設計後，應依其設計擬訂以風險為基礎的安全測試計畫，以利在開發實作完成後，進行安全需求驗證。

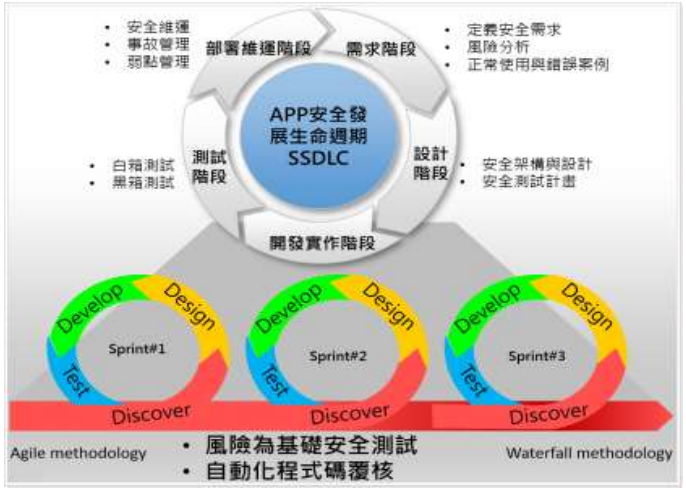


測試計畫內容	
測試目標	資源與時程規劃
測試策略 – 風險分析 – 測試方法 – 測試範圍 – 測試環境	測試報告格式
測試程序及接受標準	應變措施

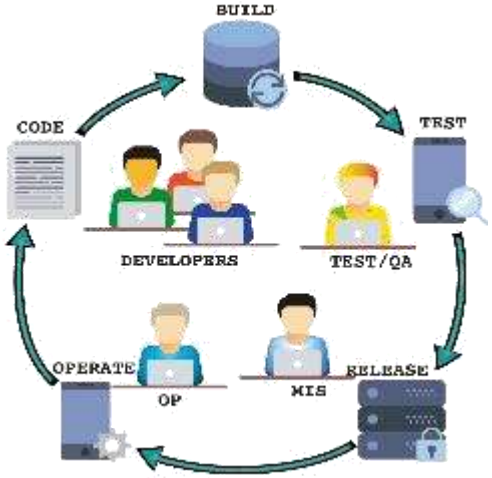
4.開發實作階段

- 建立經認可的開發工具清單，對於所有使用之函式庫皆審查其安全歷史，並研擬安全之替代方案。
- 程式開發人員需依行動應用App安全開發實務撰寫安全程式碼並實施單元測試，可依據「行動應用App基本資安檢測基準」建立查核點，同時應執行程式碼靜態分析，可使用靜態自動分析工具或由主管或不同開發人員實施必要之人工審核。
- 本階段簡介下列2項開發人員應該要知道的基礎知識：

瀑布式SSDLC與敏捷式開發



行動應用App之DevOps開發流程



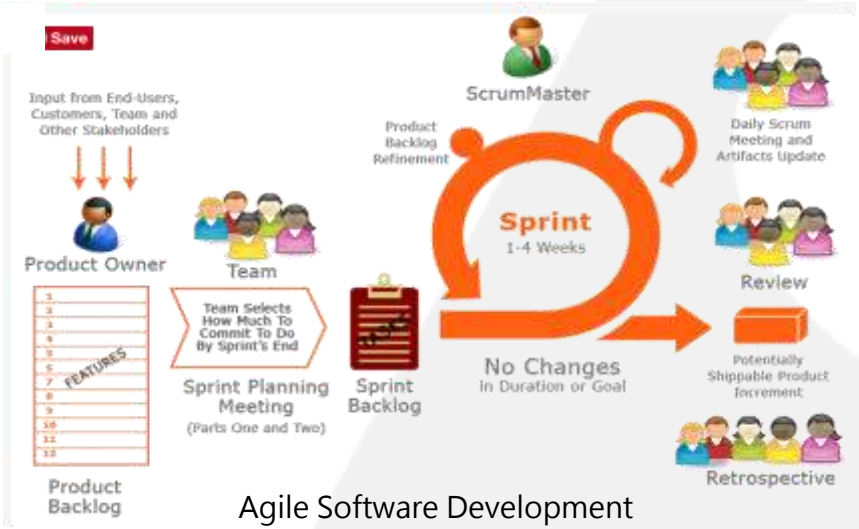
- 為使開發人員執行開發實作階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「開發實作階段安全檢核表」，詳見該指引附件4。

A.瀑布式SSDLC 與敏捷式開發

1.在敏捷式開發中，軟體專案的構建被切分成多個子項目，各個子專案的成果都經過測試，具備整合和可運行的特性。

2.由麻省理工學院史隆管理學院評論(MIT Sloan Management Review)所刊載的一篇為時2年對成功軟體開發專案的研究報告：

- 大多數的成功軟體開發專案是使用反覆循環的開發方式，而不是瀑布式過程。因在反覆循環的開發方式下，開發團隊原則上每天至少會有1次(以上)的機會，把新程式碼整併至既有系統中，並通過測試，如此頻繁的變更，可讓團隊更快的做出必要的反應。
- 開發團隊具備運作多個產品的工作經驗。
- 很早就致力於構建和提供內聚的架構。



Agile Software Development

資料來源：http://www.essentialsln.com/agile-software-development/

3.建議在進行敏捷式開發時，開發者應熟悉相關的安全實務，再輔以靜態自動分析工具，才能維持敏捷的特性。

B.行動應用App之DevOps開發流程

- DevOps一詞係來自Development和Operations的組合詞。
- DevOps是一種用來促進「軟體開發人員」和「IT運維技術人員」之間的溝通、合作與整合的過程及方法的統稱。
- 近年來不論大型或小型的開發團隊皆廣為採用。

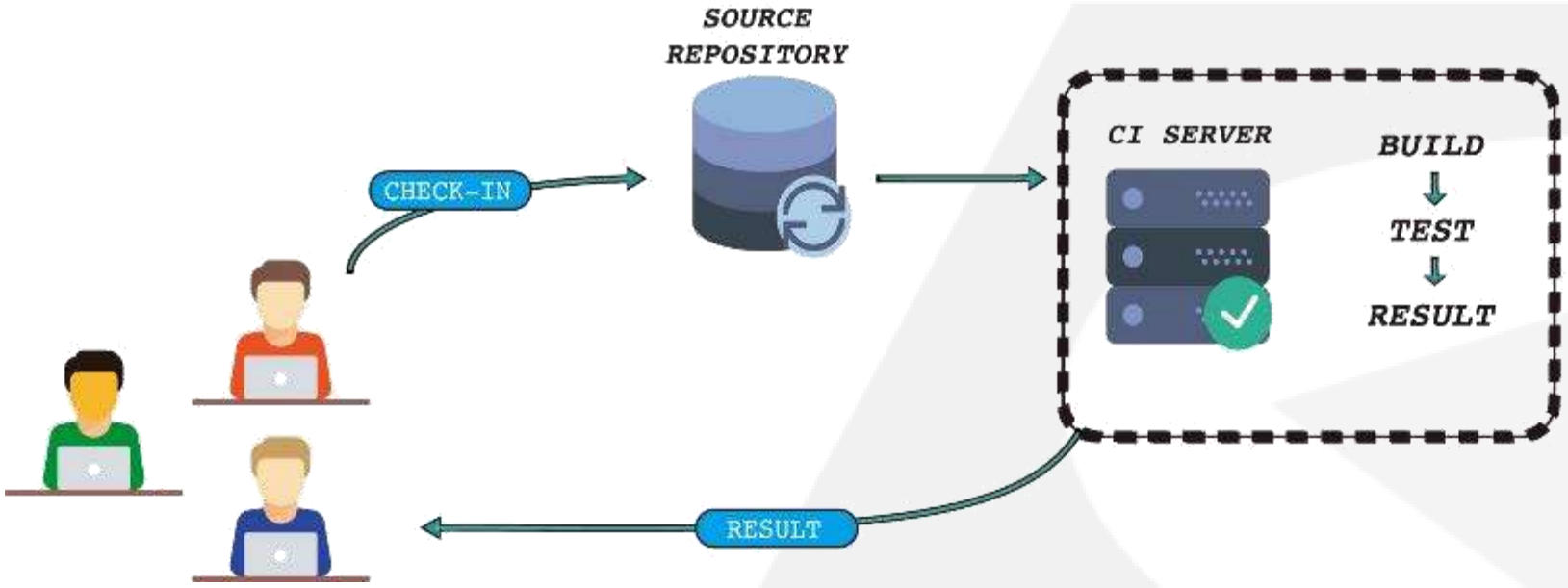
參考Suzie Prince於2016年2月11日發表的「The Product Managers' Guide to Continuous Delivery and DevOps」一文中，說明下列3個重要概念。

持續整合 Continuous Integration	持續交付 Continuous Delivery	持續部署 Continuous Deployment

ithome DevOps專區：http://www.ithome.com.tw/devops

持續整合(Continuous Integration)

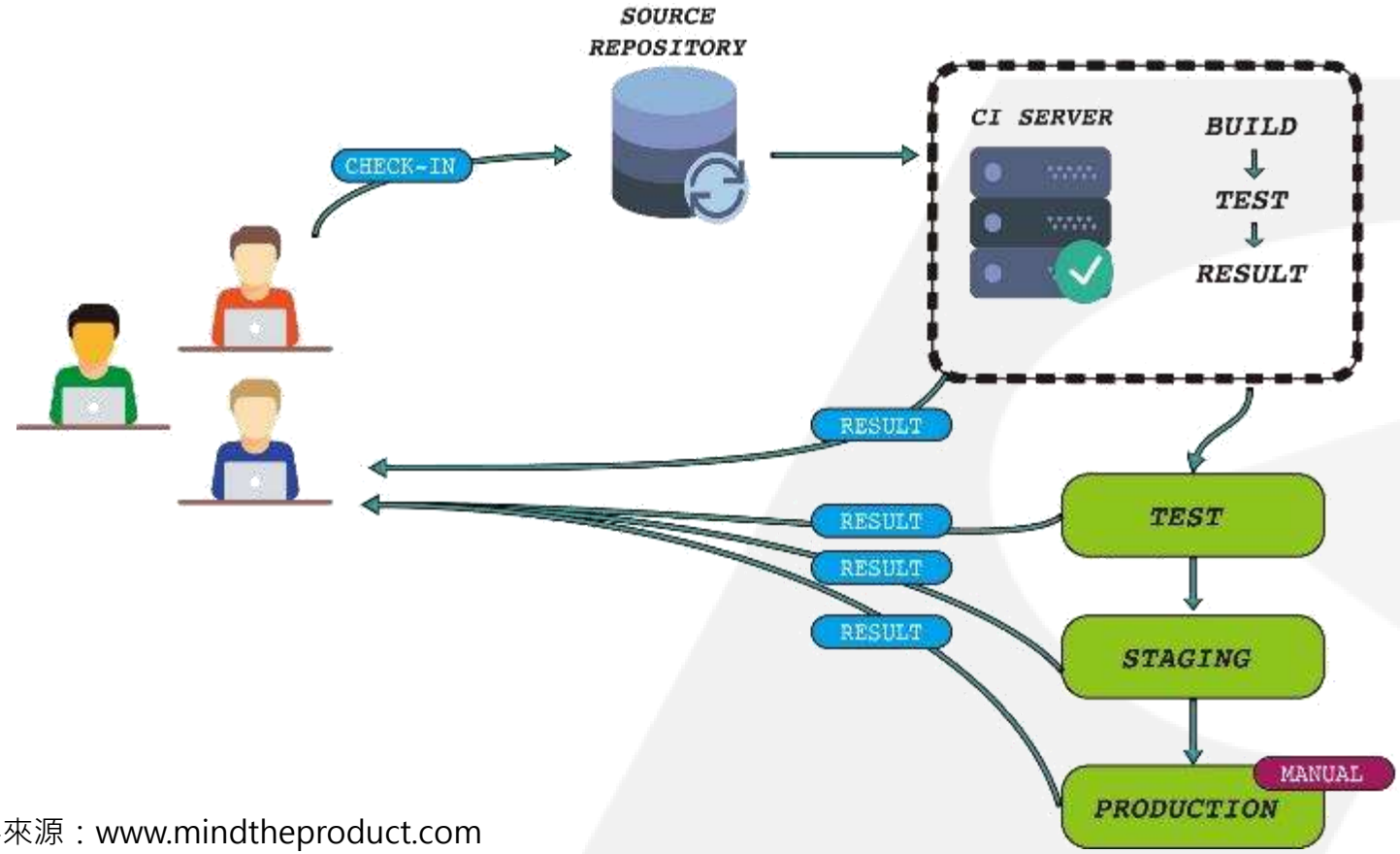
強調開發人員提交程式碼之後，立刻進行構建、(單元)測試。根據測試結果，以確定程式碼和原有程式碼能否正確地整合在一起。



資料來源：www.mindtheproduct.com

持續交付(Continuous Delivery)

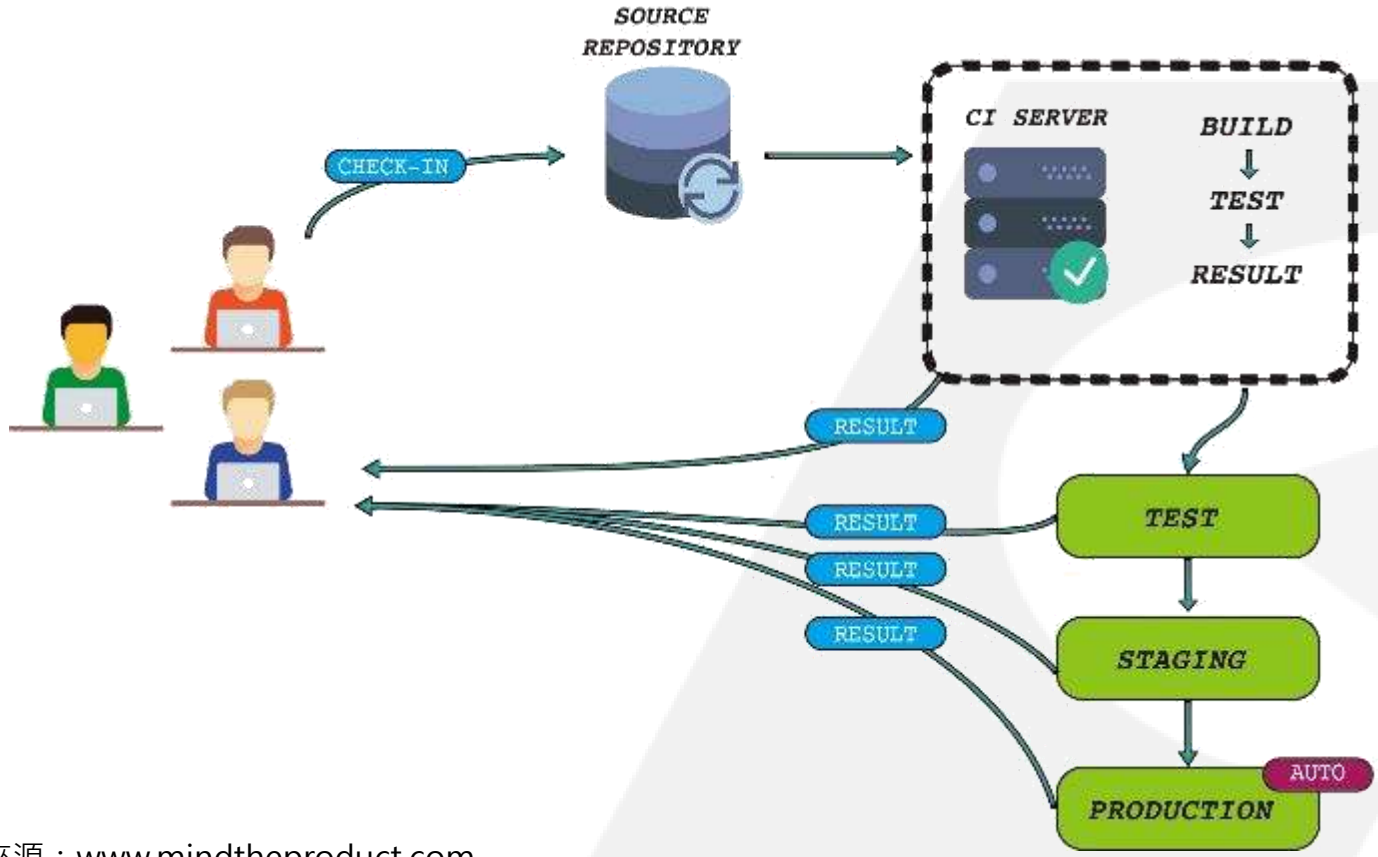
在持續整合的基礎上，將整合後的程式碼部署到更貼近真實運行環境的「類正式環境」(production-like environments)中。



資料來源：www.mindtheproduct.com

持續部署(Continuous Deployment)

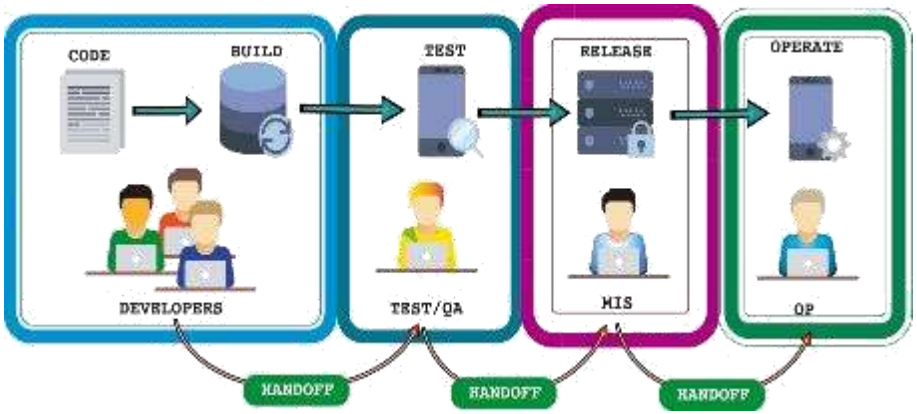
持續部署則是在持續交付的基礎上，把部署到正式環境的過程自動化。

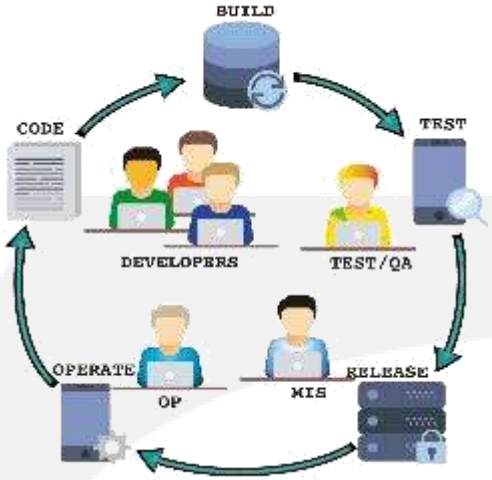


資料來源：www.mindtheproduct.com

傳統開發/測試/上線/維運與DevOps的對比流程

傳統開發/測試/上線/維運流程示意圖





DevOps開發/測試/上線/維運流程

資料來源：www.mindtheproduct.com

DevOps對比傳統流程最大的差異就是各角色以專案為單位整合成為單一團隊，持續整合、持續交付及持續部署不間斷，藉由自動化工具與DevOps運作團隊文化，加速App產品化的效率。

5.測試階段

- 為確保行動應用App之需求、規格及安全性功能如預期運作，需針對開發完成的App進行檢測。
- 本階段將帶領學員了解：

A.檢測目標

B.檢測流程

C.檢測方法

- 為使開發人員執行測試階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「測試階段安全檢核表」，詳見該指引附件5。

A.行動應用App檢測目標

進行「驗證設計規格」與「確認符合需求」

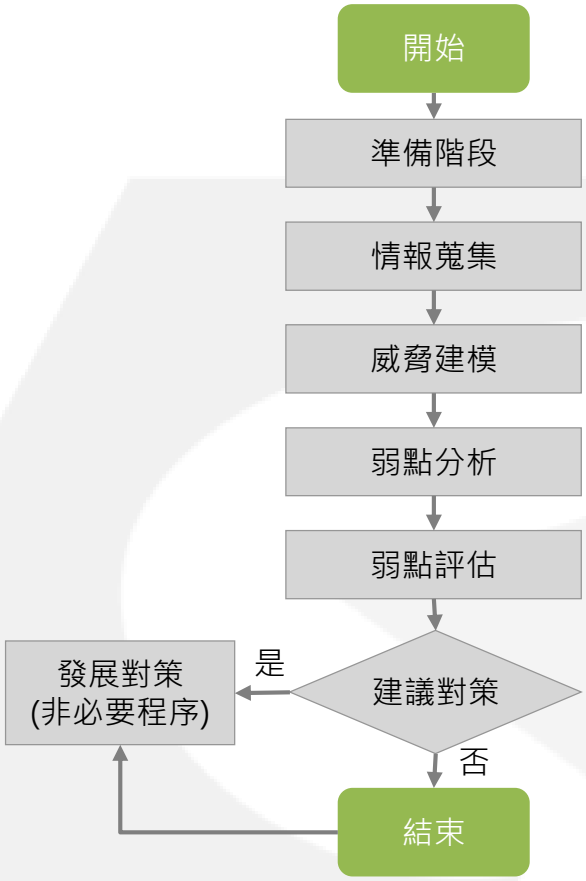
除確保其符合客戶之功能或非功能性要求，安全規格亦是當前政府與使用者重視的要點。

	驗證	確認
ISO/IEC 12207	• 檢查軟體產品並提出客觀證據，以證實達成訂定的規格要求。 • 判斷某發展階段的軟體，達成前項發展階段訂定的需求或限制。	• 檢查軟體產品並提出客觀證據，以證實達成某一特定預期功用的需求 • 確定依據需求規格製作的最終軟體產品，是否滿足特定使用目的。
CMMI相關指引	確保工作產品符合其指定需求的規格。	在展示最終軟體產品或產品元件在需求環境中，實現客戶需要的產品。

B.行動應用App檢測流程

OWASP的行動應用App的安全測試指引

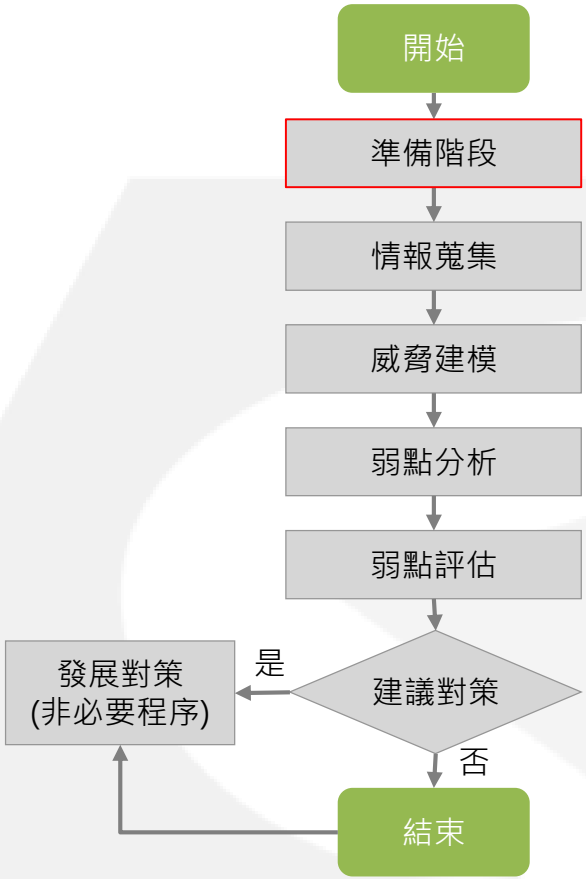
參考採用之基本App核心檢測流程，透過App資料擷取、威脅建模及弱點分析等流程，最後以指引說明或工具檢測等方式，獲得自我檢測結果與建議。



資料來源：The OWASP Foundation

階段1：準備階段(Preparation)

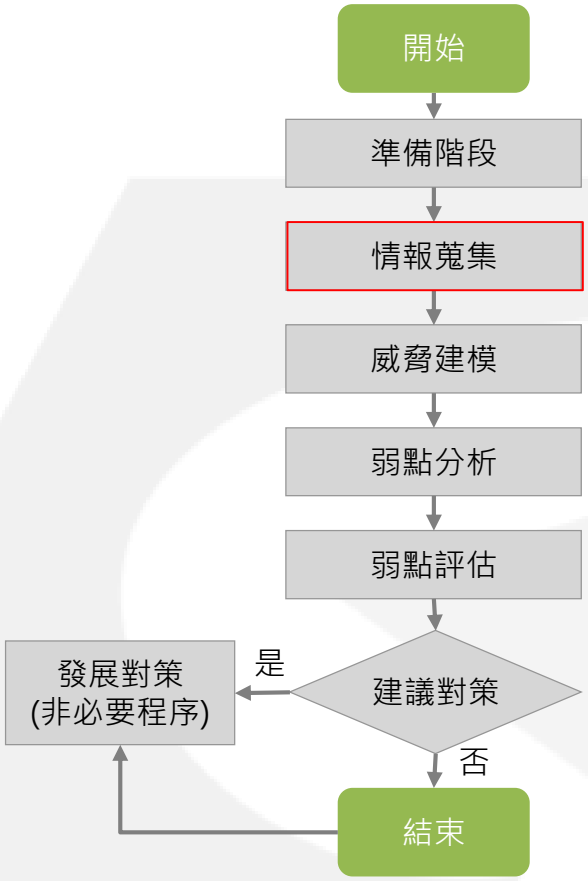
- **主要任務**
  - 建置一個單純且不受干擾之測試環境，並可操控與掌握各項控制因子，諸如越獄(JB/root)與否、系統平台、系統版本、App介面、行動裝置管理等等，可作為檢測時成為考量因素操控使用。
- **檢測前**
  - 應提供檢測者關於該App之開發事項或心得等，有助於檢測者更有效率的完成檢測，並且有系統地進行檢測流程步驟及完善檢測紀錄與報告。
- **檢測過程**
  - 應考量攻擊者可能的攻擊角度與手法，透過App內外部、網路或裝置存取以及可能的檢測方法與工具等。



資料來源：The OWASP Foundation

階段2：情報蒐集(Intelligence Gathering)

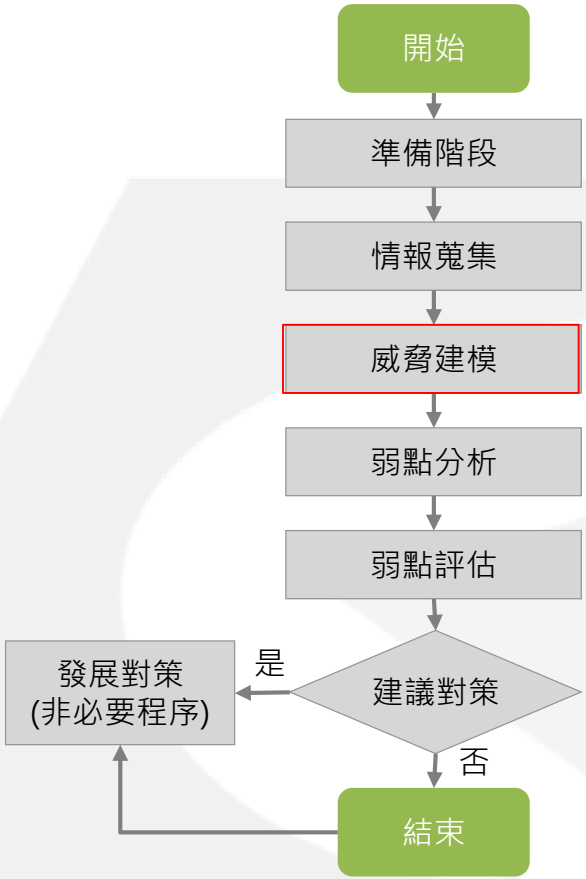
- **主要任務**
  - 盡可能蒐集該App的相關資訊。
- **進行2種分析**
  - 環境分析(Environment Analysis)
    - 進行商業案例(Business Case)與相關利害關係者(Stakeholder)的辨識與分析，亦需分析內部的流程與架構。
  - 結構分析(Architectural Analysis)
    - 針對App本身：網路介面、使用資料、相關資源的溝通、會談管理(Session Management)、越獄(JB/root)偵測等。
    - 針對運行環境(Runtime Environment)：行動裝置管理(MDM)、越獄(JB/root)與作業系統版本。
    - 針對後端服務(Backend Service)：App伺服器(Application Server)、資料庫(Database)與防火牆(Firewall)等。



資料來源：The OWASP Foundation

階段3：威脅建模(Threat Modeling)

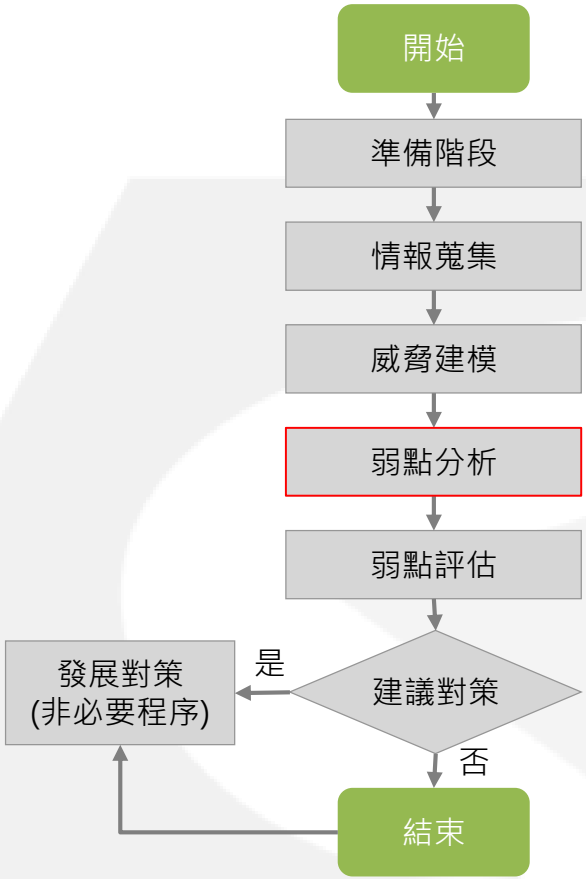
- **主要任務**
  - 辨識App的各項威脅。
  - 評估威脅可能帶來的風險，並進行高低排序。
  - 發展對應的改善控制措施。
- **執行步驟**
  1. 切割與群集App
  2. 威脅辨識
  3. 威脅評估
  4. 風險分級
  5. 發展改善計畫
  6. 制定測試案例(Test Case)



資料來源：The OWASP Foundation

階段4：弱點分析(Vulnerability Analysis)

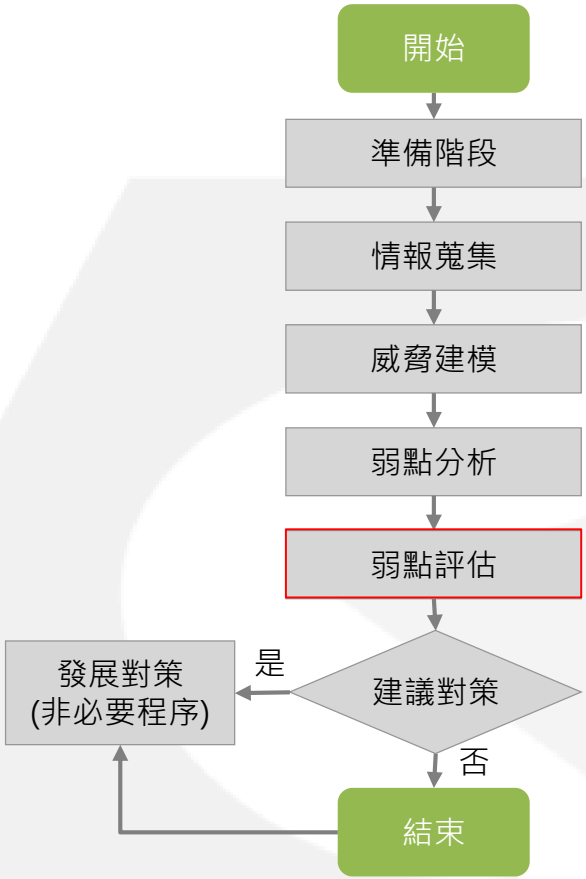
- **主要任務**
  - 依測試案例(Test Case)，辨識行動應用App的各項弱點。
  - 執行測試案例時，可使用下列3種技術：
    - 靜態方法(Static Method)
      - 逆向工程(Reverse Engineering)
      - 自動化與人工原始碼分析(Automatic and manual source code analysis)
    - 動態方法(Dynamic Method)
    - 鑑識方法(Forensic Method)



資料來源：The OWASP Foundation

階段5：弱點評估(Vulnerability Assessment)

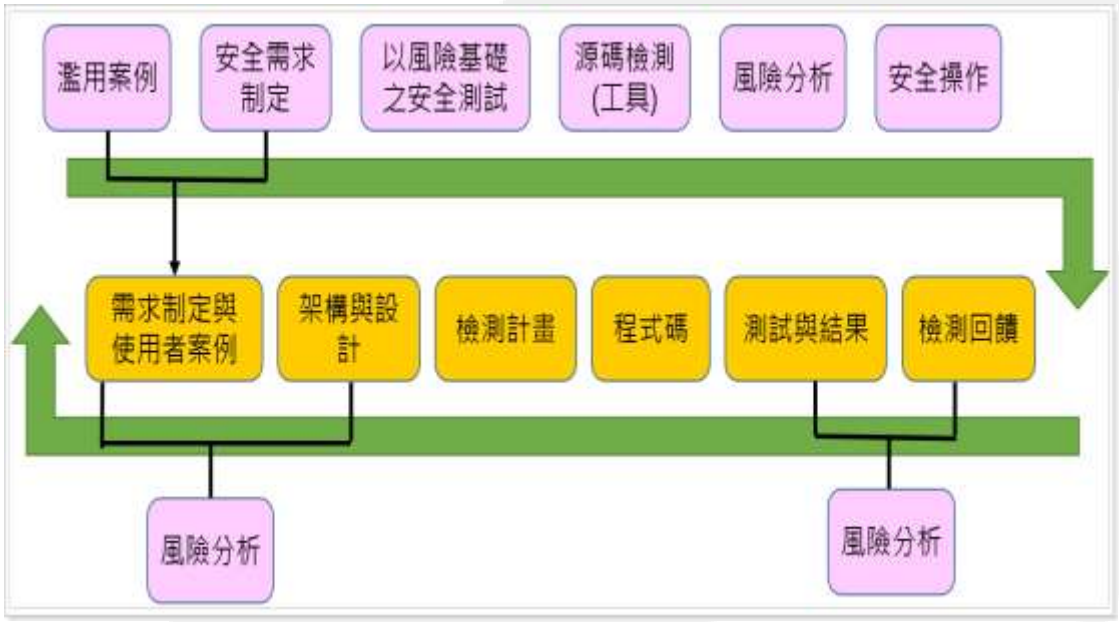
- 透過使用自動化的弱點掃描工具，進行不同目的之檢測方法，以發現App中存在的已知弱點。
- 在偵測弱點後進行弱點相關的評估與統計報告，提供開發者進行App改善之建議參考 。



資料來源：The OWASP Foundation

C.行動應用App檢測方法(1/2)

- 參考McGraw所提出的TouchPoint Model。
- 透過「需求制定與使用者案例」、「架構設計」、「檢測計畫」、「程式碼」、「測試與結果」、「檢測回饋」等面向，提出7種最佳檢測實務。
  - 源碼檢測(以使用工具為主)
  - 風險分析
  - 滲透測試
  - 以風險為基礎之安全測試
  - 濫用案例(Abuse cases)
  - 安全需求制定
  - 安全操作



TouchPoint Model方法論

資料來源：https://www.cigital.com/presentations/ARA10.pdf

C.行動應用App檢測方法(2/2)

3種行動應用App弱點分析技術

靜態方法

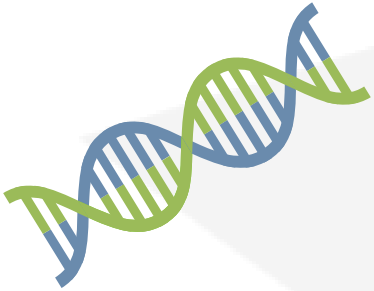
- 在不需要執行App的情況下，確認需求與規格是否符合預期。
- 常見的執行方法為檢視(Inspection)、結構化逐步審查(Structured Walkthrough)與主動審查(Active Review) 。
- 可針對標的為App產品計畫、需求規格文件與程式碼等內容。
- 使用工具如dex2jar、otool、androwarn、Flawfinder等。

動態方法

- 觀察App的活動細節是否有異常。
- 常見的執行方法為黑箱測試(Black Box)與白箱測試(White Box)。
- 針對App的各項介面、功能與非功能需求，以及相關執行邏輯等正確性。

鑑識方法

- 透過時間軸及Log分析，驗證與確認App之安全規格與行為合理性。



6.部署維護階段

- 行動應用App進入部署維運階段時
  - MIS人員應先對作業系統、伺服器、資料庫及軟體本身等部署環境進行安全檢視工作，包括使用正確的安全設定、關閉非必要之服務與網路埠、移除測試帳號與資料，以及使用最低權限執行等。
  - 遇有軟體或作業系統環境變動時，應做好變動管理工作，包括識別可能變動及分析變動對安全性之影響，另應進行持續性之監測，以驗證軟體運作安全性。

- 為使開發人員執行部署維運階段需要從事的安全活動有所依循，請參照「行動應用App安全開發指引」已整理「部署維運階段安全檢核表」，詳見該指引附件6。

A.安全維運

安全組態管理

行動應用App以雲端服務模式呈現時，應強化伺服端系統管理。

特別強化下列安全管理領域：

- 系統管理：系統更新、關閉不必要服務、限制軟體安裝。
- 網路管理：網路配置、防火牆設定。
- 模組元件管理：版本控管、取得來源管理。
- 伺服器管理：帳號權限管理、加密連線設置、監控與稽核。
- 應用程式：帳號權限管理、開發與測試過程所使用的功能應移除或關閉。

安全部署管理

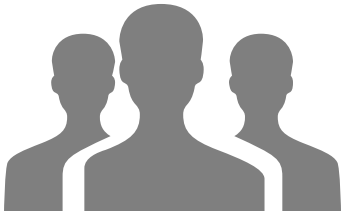
根據「系統管理」、「網路管理」、「伺服器管理」、「模組元件管理」及「伺服應用程式管理」等項目類別，進行各個類別的部署管理並具體採用前述之工具與檢核表進行安全項目自我檢測，以完成伺服端之安全部署管理。

亦可採用滲透測試進行檢測，可參考：

- OWASP
- IECOM
- SANS

B.事故管理

儘管已執行相關的資安檢核與設置，仍有發生資安事故之可能



組織應考量：

- 資訊安全通報及應變程序
- 資訊安全弱點回報分析措施
- 資訊安全改善與矯正措施

詳細資料亦可參考ISO/IEC 27001與ISO/IEC 20000等標準

C.弱點管理

1 管理規範	2 技術強化	3 持續落實
應制定整體弱點管理政策、程序落實及執行與後續追蹤及改善等，並進行相關弱點稽核與紀錄。	在弱點管理上可建置或安裝相關管理系統，如防毒軟體、防火牆與入侵偵測防禦系統等，協助相關弱點之即時偵測防禦與追蹤紀錄管理。	當弱點被偵測且分析後，應進行相關後續修補與更新動作。

課程大綱

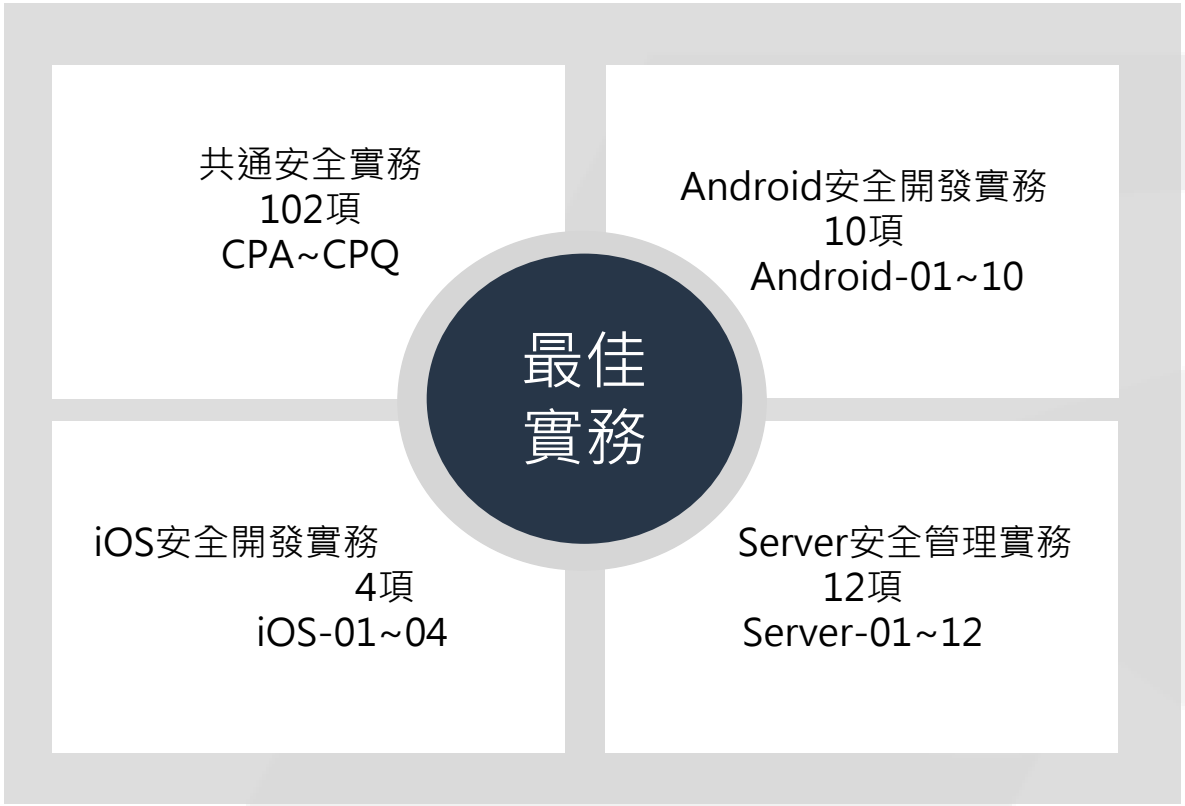
第1單元	行動應用App安全基礎概論	1.0小時
第2單元	安全軟體開發生命週期	1.0小時
第3單元	安全行動應用設計最佳實務	1.0小時

行動應用App安全開發最佳實務

依據我國經濟部工業局「行動應用App基本資安規範」及「行動應用App基本資安檢測基準」及彙整中國大陸、歐盟、日本、美國及CSA行動應用App安全開發實務要點。

4
個實務類別

128
項實務準則



行動應用程式發布(A)

CPA-01 於行動應用程式商店提供及應用程式內實作提供隱私權政策說明連結，說明欲存取之敏感性資料、行動裝置資源及宣告權限用途。

簡介

- 使用者下載安裝行動應用App時，除了功能及價格因素，還會考量隱私的議題。
- 事前規劃並撰寫此行動應用App的隱私權政策。
- 可參考Center for Democracy & Technology (CDT)的行動應用程式開發最佳實務的隱私權宣告範本(https://www.cdt.org/files/pdfs/Apps%20Best%20Practices%20v%20beta.pdf) 。

隱私權政策的參考範本如下：

- 隱私權保護政策的適用範圍
- 個人資料的蒐集、處理及利用方式
- 資料之保護
- 網站對外的相關連結
- 與第三人共用個人資料之政策
- Cookie之使用
- 隱私權保護政策之修正

開發生命週期	需求階段、開發實作階段、部署維運階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.1.1	行動應用App基本資安檢測基準	4.1.1.1.2

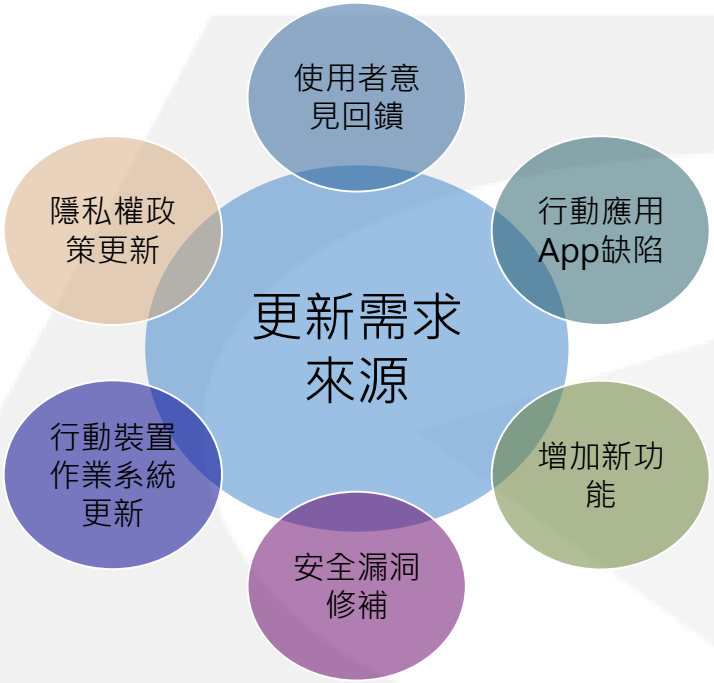
行動應用程式更新(B)

CPB-01 行動應用App應提供功能性與安全性更新，以因應新功能加入、發現漏洞及平台安全性提升之需求。

簡介

使用者自平台付費或免費取得行動應用APP後，在其持續使用的過程中，行動應用App可能會因為一些因素有更新需求。

- 可使用電子郵件或App的提示功能告知使用者更新資訊。
- 開發人員應隨時注意重大更新事項，尤其與安全或隱私有關之議題。



開發生命週期	部署維運階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.1.2	行動應用App基本資安檢測基準	N/A

行動應用程式安全性問題回報(C)

CPC-01 於可信任之應用程式商店或行動應用程式內，提供聯絡網頁、電子郵件、電話或其他類型聯絡方式。

簡介

- Google Play的商店資訊及iTunes Connect App屬性的版本資訊中，必需填寫開發人員的聯絡網頁、電子郵件、電話或其他類型聯絡方式，並注意此資訊應定期維護。
- 如果是中大型開發商，建議使用客服專線或是公務用的固定聯絡方式，避免以登記個別開發人員之聯絡資訊，以避免當該開發人員職位異動時，開發商無法接收到客訴或安全性通報資訊。

開發生命週期	部署維運階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.1.3	行動應用App基本資安檢測基準	4.1.1.3.1

敏感性資料蒐集(D)

CPD-01 需求階段需要識別計畫開發行動應用App蒐集、處理及利用敏應資料類別(例如密碼，個人資料、地理位置、財務資訊及錯誤日誌等)及行動應用App平台與發行國的隱私法令要求，定義安全性需求。

簡介

- 通常免費或無償提供行動應用App，多數負有蒐集、利用及分享行動智慧裝置擁有者個人資料有關識別與行為資料任務 。
- 個人資料的蒐集、處理及利用應遵循發行國之個資保護相關法令規定。

> 明確告知蒐集當事人的個人資料由那一個公務或非公務機關以何種目的蒐集個資當事人的何種類別資料，且會利用之期間、地區、對象及方式，及當事人對已被蒐集的個人資料有何權利，讓使用者在下載之前或是蒐集使用者本人個人資料前有判斷的依據，讓使用者可以選擇同意或是拒絕，如果選擇拒絕是否會影響公務或非公務機關對當事人服務的品質。

- 行動應用App專案經理與系統分析人員應充分識別需求單位處理個人資料在內的敏感資料類別，並定義出安全保護需求，由系統分析 / 設計人員進一步設計安控及隱私保護措施。

開發生命週期	需求階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.1	行動應用App基本資安檢測基準	N/A

敏感性資料利用(E)

CPE-01 不要使用設備或與其他App共享持久性敏感資料作為使用者識別碼，如設備ID，最好採用隨機產生識別碼(參考CPM-02)，並採用相同的資料最小化權限原則應用在行動應用App的Session ID及HTTP Session ID/cookies等。

簡介

- 為保持使用者良好體驗，使用者不需持續的進行身分認證，通常以儲存Session ID/Cookies在使用者端用來讓Server端識別使用者身分用，故攻擊者只要獲取使用者Session ID/Cookies就可以使用者身分進入伺服器竊取。

  常用的攻擊為猜測(Session Prediction)、連線劫持(Session Hijacking)及詐取固定(Session Fixation)。

- 系統設計及開發人員在Server端實作網頁應用程式時，最好在每次登入時更換隨機產生或是可變量的Session ID(參考CPM-02)，並設定逾時就清除session ID。

- 在行動應用App的本地端的Session ID也要限制被其他 App存取，以最小化權限原則設計實作，傳送至伺服器端應使用TLS連線加密，或將Session ID加密傳送，至server端再解密，不要將 Session ID 使用 URL (GET) 以明文方式來傳遞。

開發生命週期	設計階段、開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.2	行動應用App基本資安檢測基準	N/A

敏感性資料儲存(F)

CPF-01 行動應用App如需要儲存敏感性資料需依CPD-02規劃於可信任之應用程式商店或行動應用程式內聲明。

簡介

- 行動應用App如需要儲存”敏感性資料”參考CPA-01將行動應用App需要儲存敏感性資料於Google Play及App Store及行動應用App的隱私權政策中聲明。

敏感性資料(Sensitive Data)
指依使用者行為或行動App之運作，建立或儲存於行動裝置及其附屬 儲存媒介之資訊，而該資訊之洩漏有對使用者造成損害之虞，包括但不限 於個人資料、通行碼、即時通訊訊息、筆記、備忘錄、通訊錄、地理位置、行事曆、通話紀錄及簡訊。

開發生命週期	設計階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.3	行動應用App基本資安檢測基準	4.1.2.3.1

敏感性資料傳輸(G)

CPG-01 行動應用App傳輸敏感性資料應規劃並實作傳輸全程全時使用TLS 1.1以上加密，維護敏感性資料機密性及完整性。

簡介

- 新的Web應用協定HTTP/2和SPDY並不支援SSL，只支援TLS。
- Google所有公共服務的加密都是使用TLS，故如果要整合Google的一些功能，只能使用TLS。
- PCI DSS規範SSL在2016年6月30日之後不得繼續使用。

Apple將在2017年1月強制所有iOS App開始使用ATS(App Transport Security, 支援TLS v1.2)，參考iOS-04。

Android支援標準TLS實作，在官方開發人員網站有完整的概念及實作介紹，並包含實例及常見問題。

開發生命週期	設計階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.3	行動應用App基本資安檢測基準	4.1.2.3.1

敏感性資料分享(H)

CPH-01 敏感性資料分享給不同行動應用App應實作使用者同意或拒絕選項，提示使用者選擇時機可於(1)安裝時(2)當敏感資料被儲存或傳送前(3)預設設定為關閉同意，需使用者自行開啟。

簡介

- 參考CPA-01將行動應用App需要分享敏感性資料類別於Google Play及App Store及行動應用程式的隱私權政策中聲明。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.5	行動應用App基本資安檢測基準	4.1.2.5.1(1) 4.1.2.5.1(2)

敏感性資料刪除(I)

CPI-01 行動應用App儲存敏感性資料，應預先針對各國個人資料保護法或隱私要求及實務需求，定義資料於行動智慧裝置合理最長保存期限，預設App自使用者行動智慧裝置刪除或反安裝同時，針已儲存持久性敏感資料屆期可詢問是否需要刪除或是進行永久性加密。

簡介

- 資料刪除或銷毀是使用者敏感性資料在行動應用App生命週期的終點站。

- 非持久性資料(如地理位置、日誌檔等)應視需要設定留存時間。
- 持久性敏感性資料(如個人基本資料、聯絡人資料、付費資訊等)需考量各國個人資料保護法或隱私要求及實務需求，定義資料於行動智慧裝置合理最長保存期限，預設為App自使用者行動智慧裝置刪除或反安裝同時，並將敏感性資料保留期限，納入行動應用程式商店及行動應用App內隱私權政策聲明中，讓使用者了解。

開發生命週期	需求階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.2.6	行動應用App基本資安檢測基準	N/A

付費資源使用(J)

CPJ-01 於行動應用App內執行付費指示前，是否主動通知使用者，其資訊至少包含付費資源名稱、數量、金額及付費方式，並實作提示使用者同意及拒絕選項。

簡介

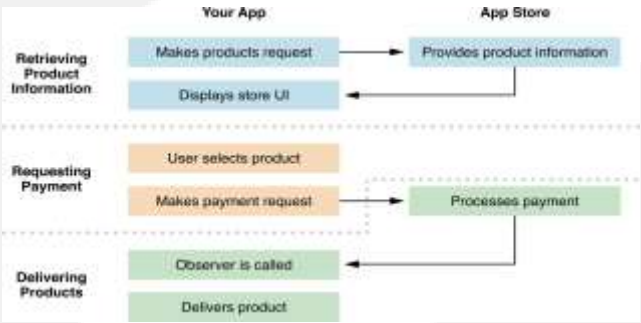
Android開發人員可以在行動應用App內呼叫In-app Billing API，向Google Play發送一個商品資訊，並且快速回覆儲存在商店付費資源名稱、數量、金額。
商品資訊可在Google Play Developer Console上設定，可定義產品資訊如下：

- 唯一產品識別碼
- 產品分類
- 價錢
- 產品描述
- Google Play 產品處理及追蹤

在正式執行付費指示前，需實作提示使用者同意及拒絕選項

iOS可以運用App Store提供的Store Kit framework嵌入程式中，並至iTune connect設定商品資訊。

App內購買流程



詳細參考In-App Purchase Programming Guide

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.3.1	行動應用App基本資安檢測基準	4.1.3.1.1 4.1.3.1.2(1), (2)

付費資源控管(K)

CPK-01 有付費資源之行動應用App需要執行CPJ-01同意選項，需經身分認證後始能呼叫付費API。

簡介

在Android需要呼叫Google Play Service的IInAppBillingService的API，呼叫API需要認證使用者端的憑證簽章及API的密鑰，實作請參考Authenticating Your Client。

在iOS中App的付費將會以呼叫Store Kit的API提出付款請求，Store Kit將會導至App Store，必須經身分認證，才能完成交易。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.3.2	行動應用App基本資安檢測基準	4.1.3.2.1

使用者身分認證與授權(L)

CPL-01 行動應用App應設計並實作適當身分認證機制，並依使用者身分授權，以防止敏感資料被非授權人員存取。

簡介

- 開發實作人員需以「使用者意願」及「最小權限」為主要安全原則。
- 在App運行先後順序上以「1.識別使用者身分」、「2.徵詢使用者意願(企業存取政策)」及「3.取得權限」，所以使用者身分認證是行動應用App取用敏感性資料權限的必要機制。
- 行動應用App設計及開發人員應視法規及實務需求，設計且實作適當身分認證機制。

常見行動應用App身分認證方式

- 行動智慧裝置解鎖PIN碼
- 行動智慧裝置內建生物辨識(指紋、虹膜、面部、聲紋)
- 雲端服務認證：Apple ID、Google ID、Microsoft Live ID
- 開放式認證：OAuth 2.0
- App自建私有身分認證
- 企業整合認證：Microsoft AD、Novell LDAP

常見實作的技術

- 帳號 / 密碼
- 觸控螢幕滑動手勢
- PKI
- 多因素認證
- 生物辨識

現行最常運用的認證機制為OAuth 2.0，使用者可以使用Google、Facebook及Windows Live。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.4.1	行動應用App基本資安檢測基準	4.1.4.1.1 4.1.4.1.2

連線管理機制(M)

CPM-01 行動應用App實作CPG-02 TLS連線，需使用編碼長度為128位元(含)以上之交談識別碼(Session ID) 。

簡介

- 攻擊者可以在經過攫取分析一系列的ID值，破解交談識別碼。
- 交談識別碼必須足夠長，以防止暴力攻擊，並驗證有效交談的存在。
- 交談識別碼長度必須至少為128位元(16位元組)，但該數目不應該被認為是絕對的最小值，作為其他實施因素可能影響其強度。例如，有公認良好的實作方式，如Microsoft ASP.NET中，利用120位元的隨機數為它的交談識別碼，可以提供很好的有效熵，和(20字符的字符串表示)其結果，可以認為足夠長的時間，以避免猜測或暴力攻擊。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.4.2	行動應用App基本資安檢測基準	4.1.4.2.1(1)

防範惡意程式碼與避免資訊安全漏洞(N)

CPN-01 如有程式碼，使用可檢測行動應用App原始碼安全檢測工具或人工進行靜態分析，檢視權限並比對是否與CPD-03安全設計及使用者設定權限相符(permission mapping analysis)。

簡介

- 行動應用程式的安全性審查可透過程式碼檢測工具或人工程式碼檢視進行審查，此過程需按照ISO/IEC 17025標準進行，且需審查之項目應考量相關對應之標準或規範施行檢測。
- 建議提供安全需求項目與理想檢核結果並加以說明，在檢核過程中不強制規範使用靜態或動態方法或兩者並用，作為履行審核項目要求的方法組合。

- 測試人員執行行動應用App原始碼安全檢測，並以工具或人工進行靜態分析時，應檢視權限並比對是否與CPD-03安全設計及使用者設定權限相符。

開發生命週期	測試階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.5.1	行動應用App基本資安檢測基準	4.1.5.1.1(5)

行動應用程式完整性(O)

CPO-01 行動應用App程式碼應使用平台供應商核發企業或個人開發者憑證簽章。

簡介

Android

開發者需使用有效的憑證以正確的簽章自己的APKs。
注意：

- 一個正式上線的應用程式，需要被正式且有足夠有效期間的憑證來簽章。
- Google建議至少要使用2048位元的版本。
- Keystore內的正式憑證一定要被保護且只讓最少量的人可以存取。

iOS

iOS 會要求所有可執行的程式碼均需使用Apple核發的憑證進行簽署。若要在 iOS 裝置上開發並發布 App

- 開發者必須向 Apple 註冊並加入 Apple開發者計畫。
- Apple 會先驗證每位開發者的真實身分，再核發憑證。
- 開發者使用該憑證對App進行簽署，並上傳至App Store進行發布。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.5.2	行動應用App基本資安檢測基準	N/A

函式庫引用安全(P)

CPP-01 行動應用App使用第三方函式庫前，需先確認其是來自可靠來源、有持續更新並經測試沒有漏洞、後端木馬及不明傳送目的地。

簡介

- 行動應用App當引用第三方函式庫時：
  - 需了解其來源是否可靠性。
  - 確認更新到最新版本的並進行使用前測試。
  - 需注意漏洞及檢測是否有不明伺服器端連線。
- 若自行編譯函式庫時，也需注意元件組成是否安全，是否有惡意程式碼被夾帶進來。
- 著名的XCodeGhost事件，或者更多的駭客、病毒的注入，都是類似的作法。包含Android及iOS的編譯紀錄均應詳細檢查。在Unix-like的作業系統，需注意LD_PRELOAD的使用，並可透過set -x的方式，檢查編譯紀錄是否有惡意的注入。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	N/A
行動應用App基本資安規範	4.1.5.3	行動應用App基本資安檢測基準	4.1.5.3.1

使用者輸入驗證(Q)

CPQ-01 行動應用App應實作驗證使用者輸入字串資料型別及長度之正確性，避免惡意輸入導致應用程式毀損、緩衝區溢位、各種注入攻擊發生。

簡介

- 即使是從應用程式產生的資料也有可能被截取和處理，這可能導致包括應用程式意外中止(產生含密鑰日誌)攻擊、緩衝區溢位(Buffer Overflow)及或隱碼攻擊(SQL Injection)等。在iOS中，這可以很容易地透過實作UITextFieldDelegate的方式進行防範。
- 實作適當的Web應用程式輸入安全控制機制，預設從客戶端的所有輸入應必須被視為不可信，而必須被有條件的驗證與處理。服務必須透過過濾和從應用程式和使用者驗證輸入，適當有條件的阻擋惡意輸入，輸入包括傳送前和所有的使用者輸入接收。

開發生命週期	開發實作階段		
不安全程式碼範例	N/A	安全程式碼範例	參考補充講義
行動應用App基本資安規範	4.1.5.4	行動應用App基本資安檢測基準	4.1.5.4.1(1) 4.1.5.4.1(2)

問題與討論