



軟體工程實務

Class 2: Requirements

逢甲大學 人工智慧研究中心
資訊工程學系

許懷中

需求

何為軟體工程

- Software engineering is an **engineering discipline** that is concerned with **all aspects** of software production throughout its **whole life cycle**. - Sommerville



How the customer explained it

何為需求

- 描述資訊系統所**應提供之服務**
- 描述資訊系統**運行的限制**
- 使用者**意欲**資訊系統**達成的目標**

需求的多樣性

- 抽象而概略
- 具體而詳盡

需求的多樣性 (cont.)

- 用來招標的標案文件
 - 概略而抽象的描述
 - 令廠商們有競逐標案的空間
- 與開發商所締結的合約
 - 具體而詳盡的描述
 - 驗收的標準
- 所提供之服務與運行上之限制

需求有兩種！



需求文件的種類

- 使用者需求 (User Requirements)
 - 以自然語言與圖表寫成
 - 服務與限制
 - 系統的性質與功能的描述
- 系統規格 (System Specification)
 - 對於系統功能、服務與運行限制的詳盡描述
 - 精確 (Exactly) 地描述將要實作的系統
 - 合約的一部分，作為未來驗收的標準

User requirements definition

- 1.** The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

User requirements definition

- 1.** The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

- 1.1** On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- 1.2** The system shall generate the report for printing after 17.30 on the last working day of the month.
- 1.3** A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
- 1.4** If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
- 1.5** Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

利害關係人 (Stakeholder)

- 與所開發之資訊系統有利害關係的人
 - 終端使用者 (End User)
 - 系統管理者 (System Manager/Administrator)
 - 系統擁有者 (System Owner)
 - 外在利害關係人 (External Stakeholder)

利害關係人 (cont.)

- 以心理保健病患資訊系統 (MentCare) 為例
 - 病患 (其資料登錄在該系統上)
 - 醫生 (負責治療病患，並且使用該系統)
 - 護士 (幫助醫生，管理處方)
 - 接待人員 (安排看診預約)
 - IT 人員 (系統安裝與維護)
 - 醫療倫理官員 (確保醫療行為合乎醫療倫理)
 - 健保官員 (審閱醫療記錄與管理資訊)
 - 醫療記錄管理員 (確保醫療記錄的正確與完整)

Exercise

- 討論選課系統的使用者需求
- 討論選課系統的利害關係人

功能需求與非功能需求

功能需求

- 描述系統所提供的服務
 - 系統如何應對各式特定的輸入
 - 系統在各種特定情境下的行為
- 描述系統**應有/不應有**的行為

範例: MentCare 的功能需求

- 使用者必須能夠搜尋所有診所的全部的預約清單
- 系統必須每天針對每個診所產生一份當天預約的病患名單
- 使用系統的每位工作人員，都應該使用自己獨一無二的8 位數員工編號以供識別

不精確的功能需求

- 模稜兩可、各自 ~~表述~~ 解讀
- 『使用者必須能夠搜尋所有診所全部的預約清單』 *這個需求描述有問題嗎？*
- 使用者的理解
 - 在全部診所的預約名單中搜尋
- 開發者的理解
 - 使用者先選擇某家診所後再開始搜尋

對於功能需求的要求

- 完整性
 - 功能需求文件應涵蓋所有使用者需要資訊系統所提供的服務與功能
- 一致性
 - 功能需求彼此間的描述不應該存在矛盾
 - 使用者能夠搜尋所有診所的所有預約清單
 - 使用者只能看到自己所屬診所的病患資訊
- 現實上，有時無法百分之百達成

非功能需求

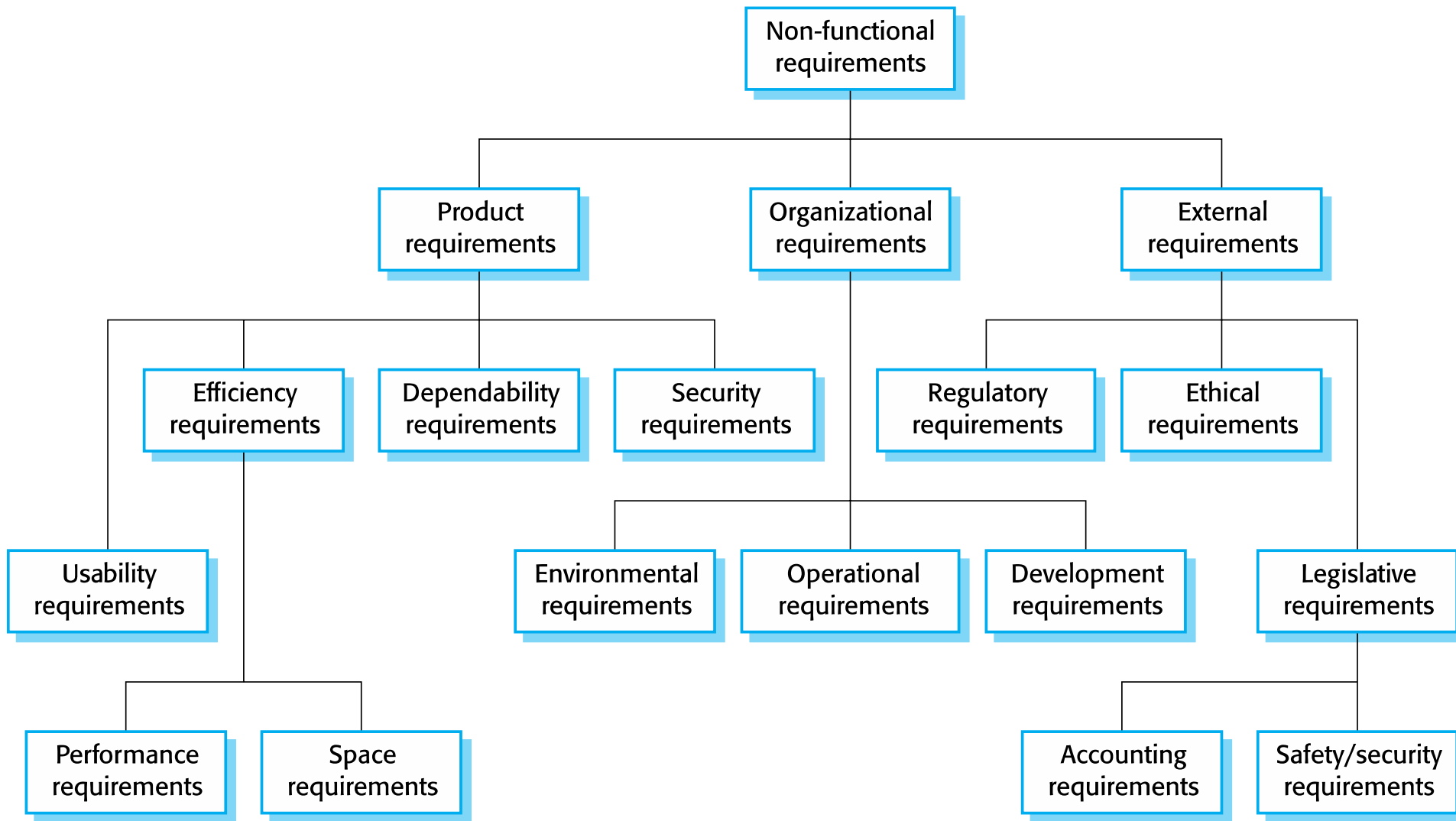
- 系統所提供之服務與功能的限制
 - 效能要求、相容性要求、硬體限制
 - 開發上的要求
 - 使用的程式語言、開發環境、遵循的標準
- 經常是對於系統整體的限制而非對於單一功能或模組的限制
- 非功能需求有時較功能需求更重要

非功能需求 (cont.)

- 非功能需求有時候會影響功能需求的實作
 - 對於反應時間的要求 → 影響網路通訊的實作
 - 對於服務人數的要求 → 影響系統架構
- 非功能需求有時會帶來新的功能需求
 - 對於安全性的要求 → 實作加密演算法

非功能需求的類型

- 系統面 (Product)
 - 系統應遵循的限制
 - 在一般上班時間 (週一到週五 8am-5pm) 必須要能讓所有診所使用
 - 期間每日服務中斷累計不能超過 5 秒
- 組織面 (Organizational)
 - 使用者應以其所屬組織的 ID 卡驗證身份
- 外在影響 (External)
 - 系統必須符合醫療法中與醫療隱私相關之規範



可驗證的非功能需求

- 非功能需求有時難以清楚地描述
- MentCare 系統在上班時間必須隨時可用
- MentCare 系統必須方便醫護人員使用

可驗證的非功能需求 (cont.)

- MentCare 系統在上班時間必須隨時可用
 - MentCare 系統每日在上班時間累計下線時間不得超過 5 秒
- MentCare 系統必須容易使用 (ease of use)
 - 接受過 4 小時訓練課程的醫護人員，操作 MentCare 平均每小時犯錯的比率應低於 2 次

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Exercise

- 請回顧先前所定義的選課系統需求，嘗試分析其何為功能需求、何為非功能需求
- 為你的選課系統加入必要的功能需求與非功能需求
- 討論目前的需求是否有模稜兩可的問題

Summary

- 功能需求描述系統所提供的服務，如醫生可以利用 MentCare 系統登錄病患資料
- 非功能需求描述系統運作上的限制，如 MentCare 系統在上班時間，每日下線累計不得超過 5 秒
- 模稜兩可的功能需求描述會令開發者無法做出使用者想要的產品
- 將非功能需求量化
 - MentCare 系統必須要能即時查詢病患的預約記錄
 - MentCare 系統必需要在 0.5 秒內，回應病患對於查詢預約記錄的要求

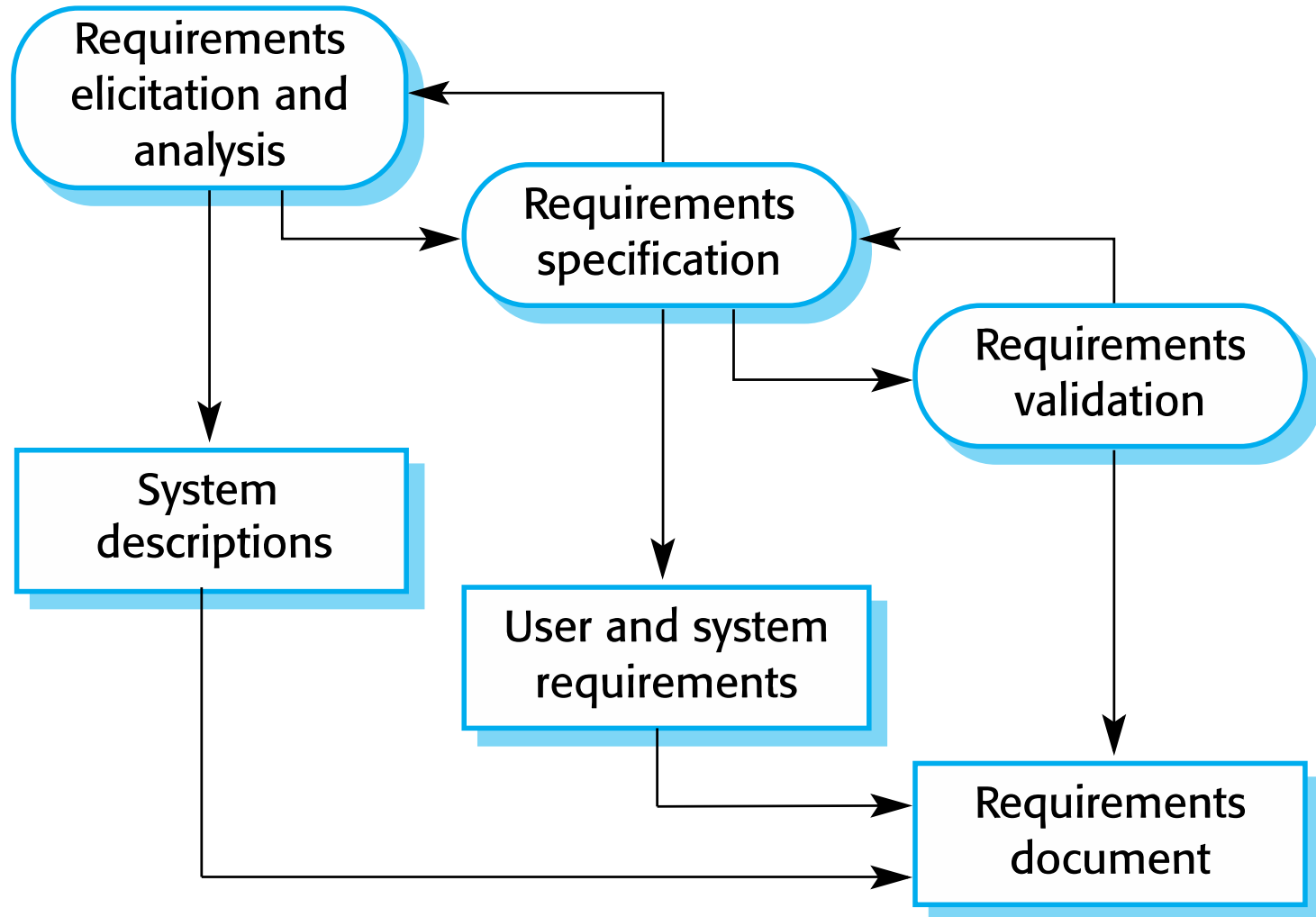
軟體工程導論 – 需求工程

需求工程程序

步驟

- 需求獲取與分析 (Elicitation and Analysis)
 - 與 Stakeholders 互動
- 需求規格 (Specification)
 - 將需求訴諸文件
 - 使用者需求與系統需求
- 需求驗證 (Validation)
 - 確認完整性與一致性

需求工程程序



軟體工程導論 – 需求工程

需求獲取與分析

需求獲取中會遭遇的問題

- Stakeholders 不知道自己想要什麼
- Stakeholders 與 開發者之間的知識落差與溝通障礙
- Stakeholders 之間對於系統的期待有所衝突
- 組織與政治因素
- 其他，如需求變動、新的 Stakeholders 或是客觀環境的改變

需求獲取

- 與 Stakeholders 一同進行
- 釐清與探索系統應提供的功能與運行上的限制
- 步驟
 - 需求探索 (Discovery & Understanding)
 - 需求組織與分類 (Organization & Classification)
 - 排列優先順序與協商 (Prioritization & Negotiation)
 - 撰寫需求文件

獲取需求的技術

- 訪談 (Interview)
- 觀察 (Observation)
 - 又稱民族誌法 (Ethnography)
- 雛形法 (Prototyping)

訪談的類型

- 封閉式的訪談 (Closed)
 - 固定的問題
- 開放式的訪談 (Open)
 - 沒有固定的議題
 - 開發團隊與 Stakeholder 一同探索系統需求

訪談實務

- 混合式訪談
- 開放性的思維，避免成見
- 提出對於系統建構的建議，而非詢問 Stakeholders 想要什麼

訪談法可能遭遇的問題

- 開發者無法瞭解領域專家的語言
- 某些知識對於使用者而言太過直覺，無法以言語形容
- 開發者對於領域知識的誤解或成見

訓練如何問問題

- 杜拉克式問句
 - 想做什麼？
 - 為什麼想做？
 - 現在在做什麼？
 - 為什麼這樣做？
- 問題要淺
 - 簡單、基本
- 問題要重要
 - 問題的答案會產生什麼影響



觀察法

- 又稱民族誌法 (Ethnography)
- 開發者觀察使用者日常工作的情形
- 使用者無須解釋他們的工作
- 可以發掘組織與企業政策上的需求

觀察法 (cont.)

- 揭露使用者**實際上**如何工作，而非他們**應該**如何工作
- 藉由協同工作與對於使用者行為的瞭解發掘需求
- 觀察法只能瞭解既有的狀況，無法發掘新的系統需要
- 來自已經不復存在之歷史因素的工作習慣

訓練如何進行觀察

- 巴黎司機訓練法
 - 將一個隨處可見的計程車司機（或某人），描述唯一一個全世界獨一無二的計程車司機
- 想像訓練
 - 閉上眼睛，想像自己從學校回家的過程
 - 將這段過程描述清楚
 - 越多細節越好



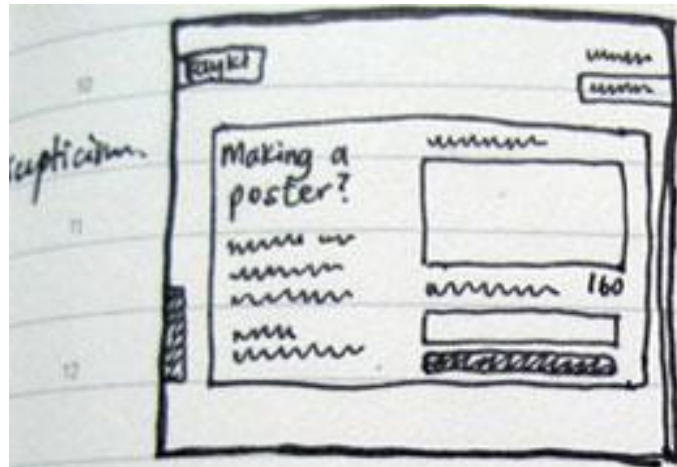
雛形法 (Prototyping)

- **快速**開發出**原型**供使用者使用
- 類型
 - 用後即丟
 - 演進式
- 作法
 - UI prototyping: Wireframes, Mockups
 - Scenario
 - Demonstration

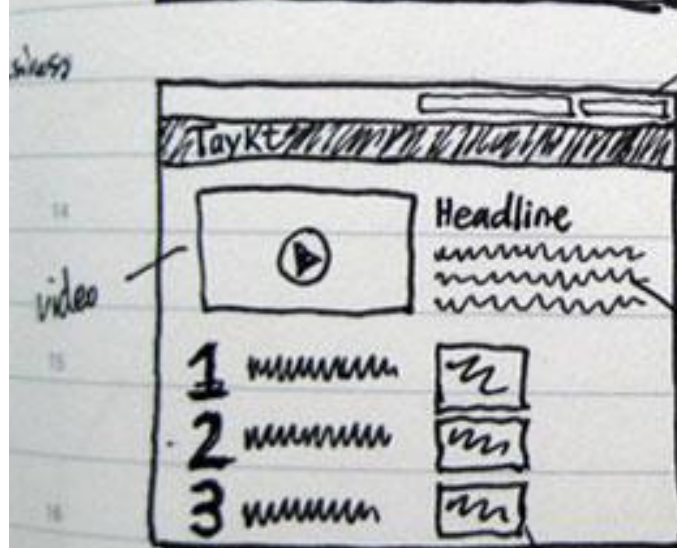
Prototyping – Wireframe (線框圖)

- 以簡單的草稿圖，把本次產品的目標功能、佈局規劃在每一頁 UI 頁面上
 - 不能出現過度的細節 – 專注在功能探討上

Wireframe (cont.)



Current: I get the feeling it has been designed for a mobile. It's a bit flat; short on explanation.



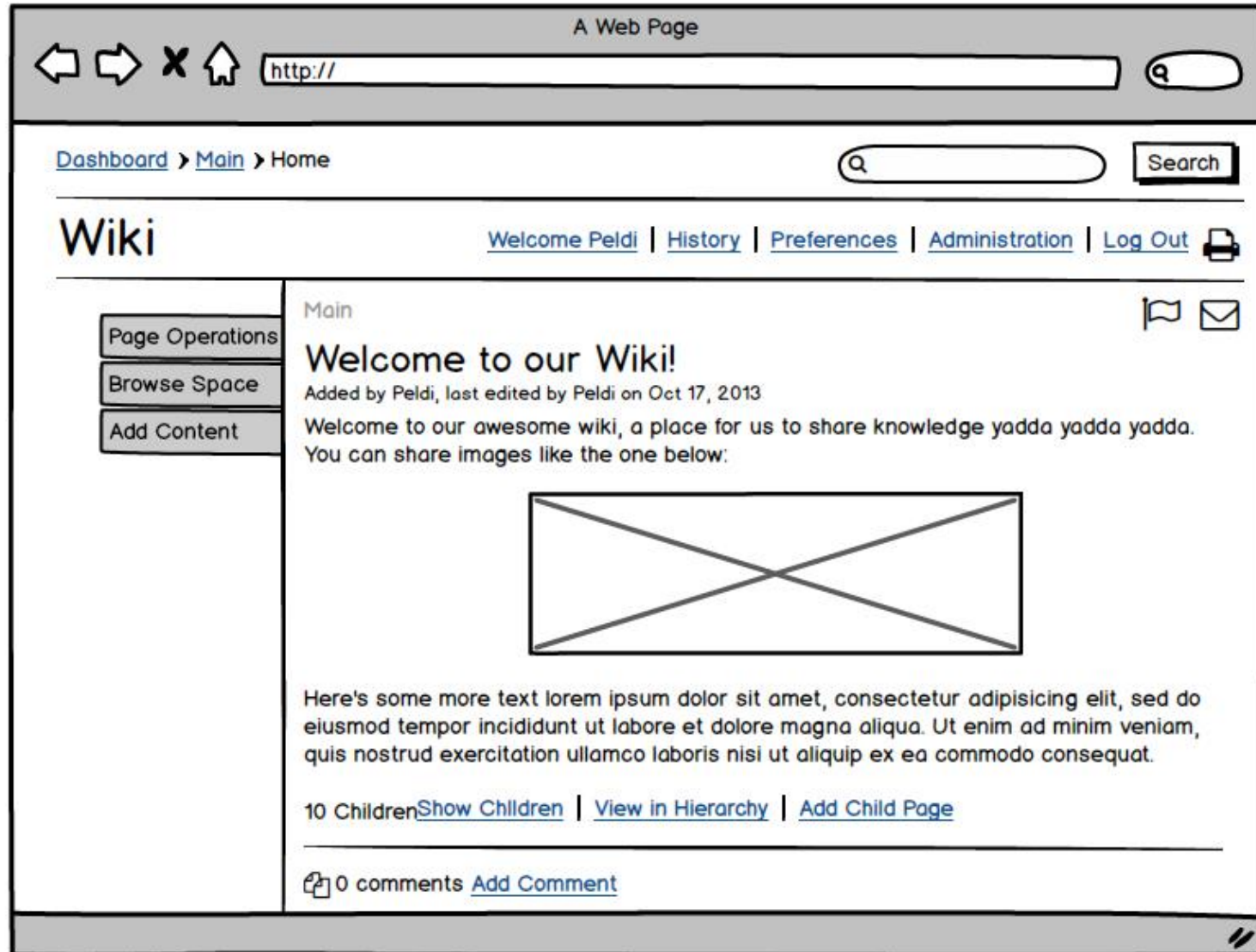
Idea: As this app is driven by the user - UGC - it needs a web 2.0-like look; feel: colour fade, shadow, images, video...

intro (what it is)

easy steps

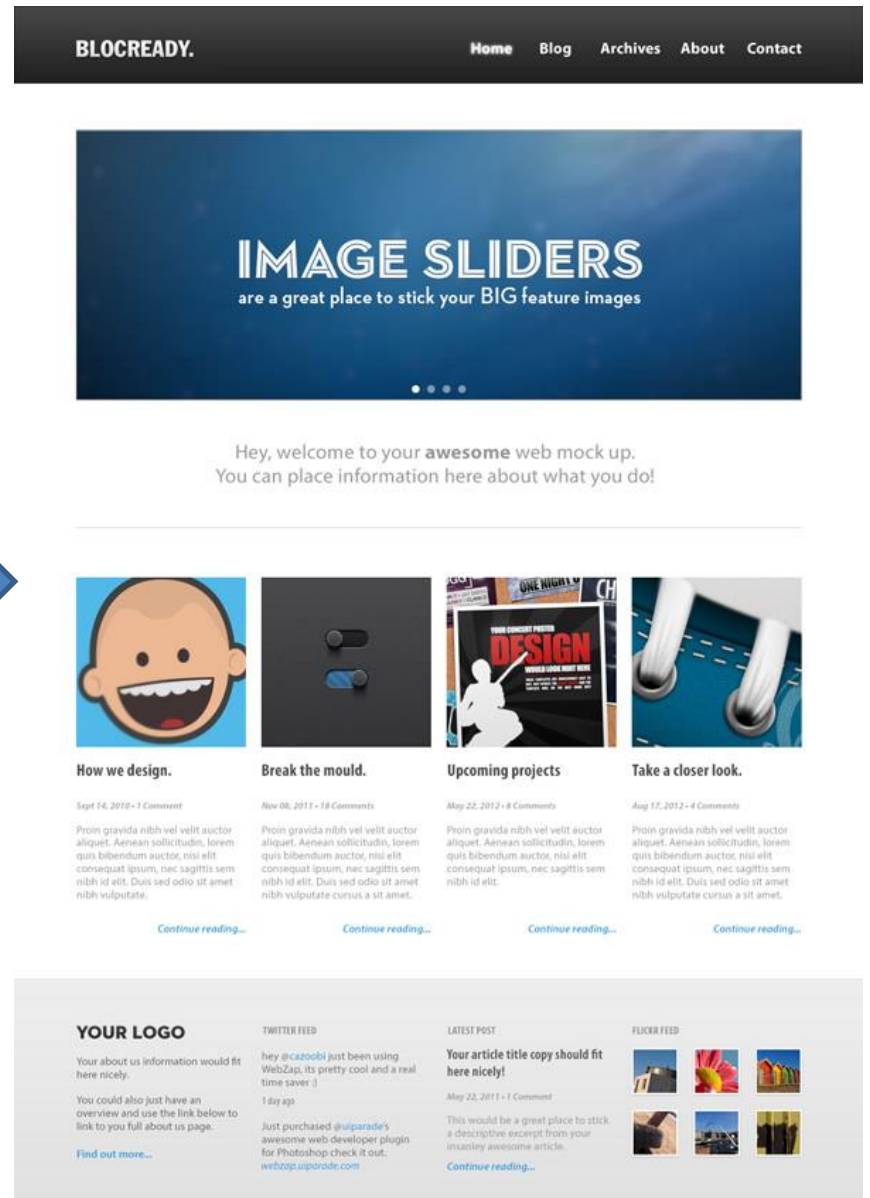
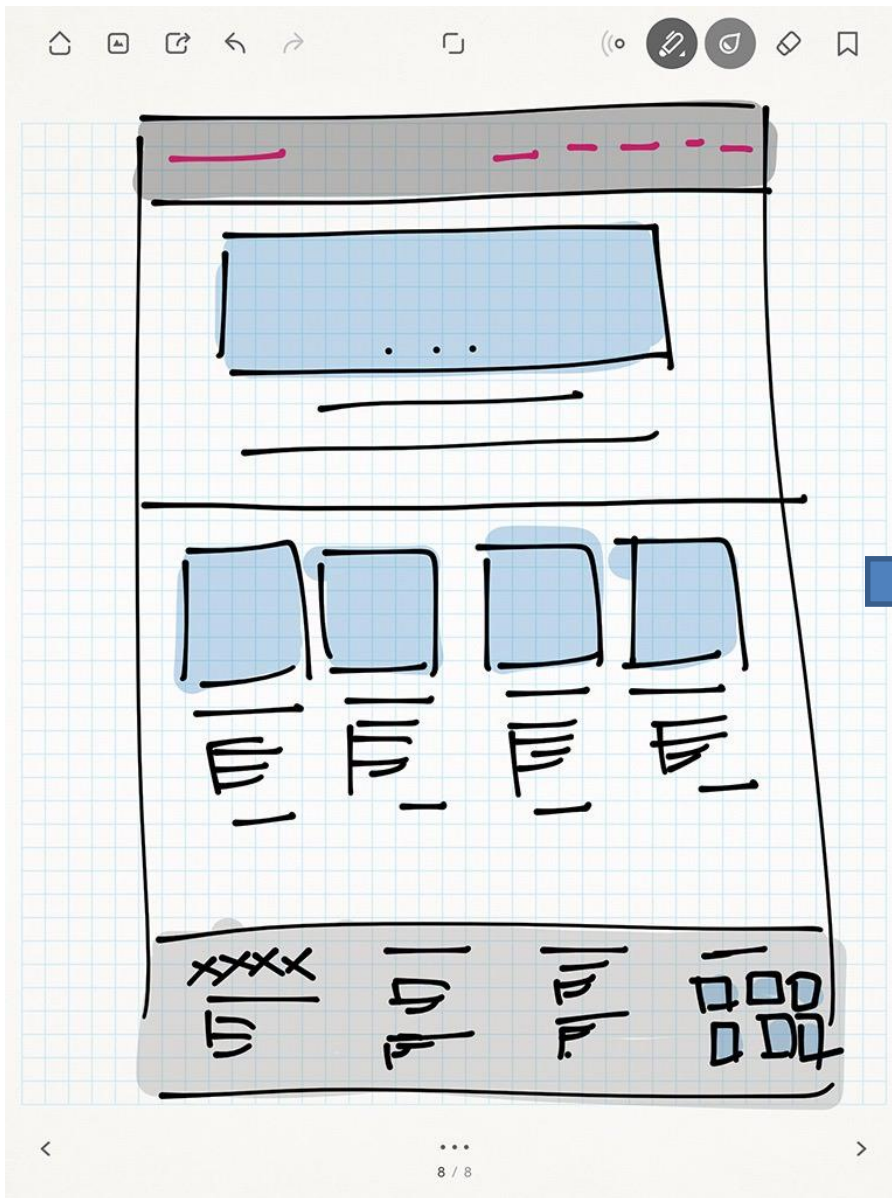
visuals - juxtaposed to aid understanding

Wireframe (cont.)



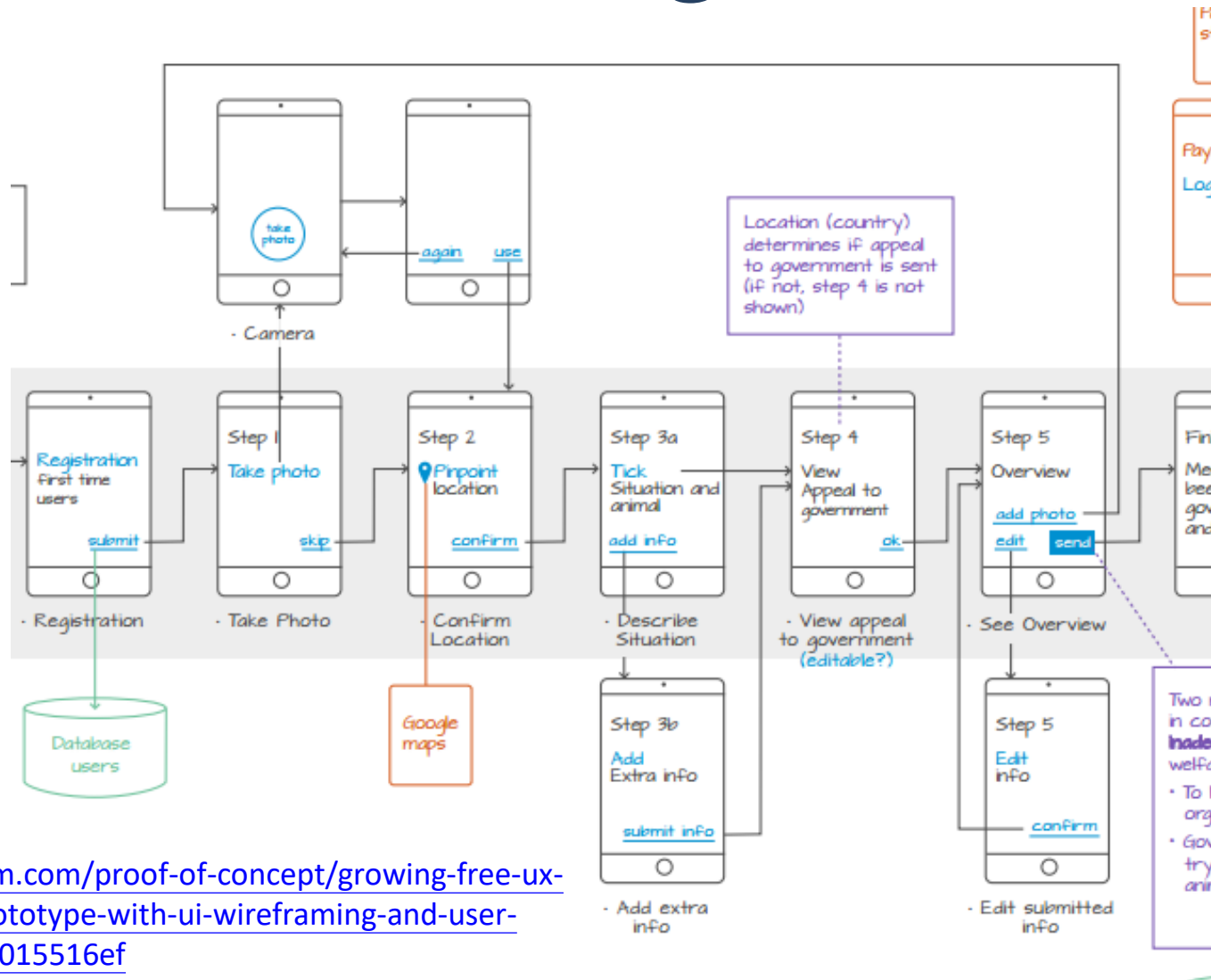
Mock-up (實物模型)

- 達到最佳產品功能為目標的視覺呈現
- 相較於線框圖，更加注重細節



出處：<https://blog.akanelee.me/posts/190997-a-beginners-mockup/>

UI Scenario using Wireframes

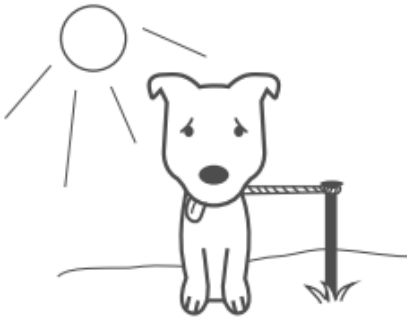


<https://medium.com/proof-of-concept/growing-free-ux-design-tool-prototype-with-ui-wireframing-and-user-scenarios-f2b0015516ef>

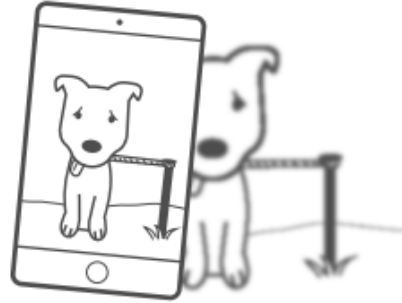


(how does it work?)

ANIMAL ALERT INTERNATIONAL



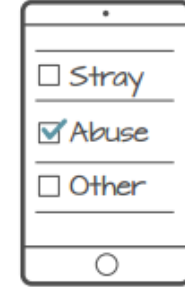
- 1 You encounter an animal in need



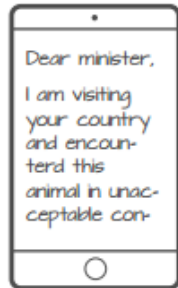
- 2 You take a picture using the app



- 3 The app finds your location



- 4 You tick the problem



- 5 A standard text - which can be modified - is attached to the message



- 6 Your report is sent to:
- A. a national or local animal welfare organisation

B. the government concerned.



- 7 You receive a picture of happy animals to compensate the negative feeling



- 8 Option for a donation to a local animal shelter, or if non-existing, a national or international organisation.

<https://medium.com/proof-of-concept/growing-free-ux-design-tool-prototype-with-ui-wireframing-and-user-scenarios-f2b0015516ef>

雛形法 (cont.)

- 優勢

- 增進溝通，客戶對於實體的事物比較容易想像
- 可以與訪談以及觀察法並用
- 降低風險，避免做出用戶不想要的東西

- 劣勢

- 僅適用於與使用者行為相關的需求獲取
- 雛形的開發成本

Summary Quiz

- 需求工程的步驟依序為何？
- 需求獲取可能遭遇的問題為何？
- 獲取需求大致上有哪些手段？

Summary

- 需求獲取與分析、需求規格、需求驗證
- Stakeholders 並不知道自己想要什麼、開發者缺乏領域知識因此難以與 Stakeholders 溝通、組織與政治因素等.....
- 訪談、觀察、雛形

需求規格

何為需求規格

- 將需求訴諸文字
- 使用者需求規格、系統需求規格
- 使用者需求規格撰寫上要令 Stakeholders 也能理解
- 系統需求規格包含較多技術細節
- 做為合約的一部分，其完整性非常重要

以自然語言撰寫需求規格

- 因為其直覺、通用同時具備較佳的表現力
- ATM 系統的需求描述範例
 - 客戶必須要能用他的提款卡從與該提款卡相對應的帳戶中提領現金
 - 在客戶將現金由提款機取走之前，此提款的動作必需要經過銀行批准 (Approval)
 - 顧客提領的現金必須為 1,000 元的倍數
 - ATM 提供千元鈔予客戶提領

以結構化的自然語言撰寫規格

- 格式化需求規格
 - 功能定義
 - 輸入與輸出
 - 需要執行的動作
 - 前置與後置條件 (Pre- & post-condition)
 - 可能的副作用

以格式化自然語言描述需求規格

功能: 提領現金

輸入: 客戶的提款卡、客戶的密碼、
客戶提款的金額 (1,000的倍數)

輸出: 客戶提領的現金

前置條件:

客戶提領前的帳戶餘額 > 其欲提領之金額

ATM 內現金存量 > 客戶欲提領之金額

後置條件:

客戶提領後的帳戶餘額 =

客戶提領前的帳戶餘額 – 其所提領的金額

以格式化自然語言描述需求規格

動作:

客戶將提款卡插入 ATM 的讀卡機

ATM 螢幕提示客戶輸入密碼

客戶從鍵盤輸入密碼，按下確認

ATM 將客戶提款卡所代表的帳號與加密後的密碼傳送到銀行伺服器

銀行伺服器確認該帳號之密碼與使用者輸入之密碼吻合

ATM 螢幕提示客戶選擇欲進行之功能

客戶從鍵盤輸入 1 (表示欲提領現金)

ATM 螢幕提示客戶輸入欲提領之金額

客戶從鍵盤輸入欲提領金額 (金額為 1,000 的倍數)

ATM 將客戶之帳號及欲提領之金額傳送給銀行伺服器

銀行伺服器確認該帳號戶頭仍有足夠金額可供提領

ATM 吐鈔機吐出客戶與其欲提領金額相同之鈔票

客戶取走鈔票

ATM 讀卡機將客戶的提款卡退出

客戶取走提款卡

以結構化的自然語言撰寫規格

- 表格式需求規格
 - 輔助以自然語言寫成的需求規格
 - 清楚表達許多的控制條件

Condition	Action
從伺服器收到密碼不符之訊息，且輸入密碼次數尚未超過 3 次	螢幕提示請重新輸入密碼
從伺服器收到密碼不符之訊息，且輸入密碼次數已經超過 3 次	螢幕提示，密碼錯誤次數過多，請聯絡專人處理
從伺服器收到帳戶餘額不足之訊息	(1) 螢幕提示帳戶餘額不足，請重新操作 (2) 讀卡機退回提款卡
使用者欲提領之金額 > ATM 現鈔存量	(1) 螢幕提示 ATM 內鈔票不足，請至其他 ATM 或櫃臺提領 (2) 讀卡機退回提款卡

Exercise

- 根據先前選課系統的需求描述，撰寫選課系統的系統規格

需求驗證

驗證些什麼？

- 正確性 (Validity)
 - 規格所描述的系統，是否能夠良好地滿足客戶的需要
- 一致性 (Consistency)
 - 需求間是否互相衝突
- 完整性 (Completeness)
 - 是否所有客戶想要的功能都已經被涵蓋在內
- 可實現性 (Realism)
 - 需求所描述的系統，是否可以在現有的預算與技術內完成
- 可驗證性 (Verifiability)
 - 是否可以確認需求被正確的達成

驗證方法

- 複審 (Review)
- 製作雛形 (Prototyping)
- 測試案例 (Test Case)

複審

- 可驗證性 (Verifiability)
 - 需求是否可以被驗證？如何驗證？
- 可理解性 (Comprehensibility)
 - 需求是否被正確的理解
- 可追蹤性 (Traceability)
 - 是否清楚明白該需求來自何處？
- 可適應性 (Adaptability)
 - 當該需求改變，對於其他需求將帶來多大的衝擊

測試案例

使用案例: 提款

測試功能: 系統會確認客戶帳戶餘額大於欲提領金額才進行交易

系統初始狀態: ATM 螢幕提示請客戶輸入提款金額

測試動作: 客戶輸入一大於其帳戶餘額的金額

預期結果: (1) ATM 螢幕提示帳戶餘額不足，交易取消
(2) 讀卡機吐出客戶的提款卡

Summary

- 需求規格是訴諸於具體文字、圖形之需求
- 開發者以及任何相關的 Stakeholders 都應該參與需求規格的編纂
- 需求規格的正確性、一致性、完整性、可實現性與可驗證性都需在需求驗證階段仔細討論並撰寫相對應之 (驗收) 測試案例