

淺談 Startup 公司

軟體開發流程 (v2)

<https://ihower.tw>

2016/3/24



Who am I?

- 張文鈿 a.k.a. ihower
- <https://ihower.tw>
- Instructor at ALPHA Camp
- <https://alphacamp.co>
- Web developer since 2002



Ruby on Rails 實戰聖經

使用 Rails 4.2 及 Ruby 2.1

 SEARCH

電子書和簡體版本準備中。如果您有任何意見、鼓勵或勘誤，歡迎來信給我，謝謝。

我是 ihower，本書介紹 Ruby on Rails 這套開放原始碼的網站開發框架。如果您對這本書有任何意見或勘誤指教，歡迎來信和我聯絡。

[關於本書](#)[回首頁](#)

Part 1: 入門實作

1. Ruby on Rails 簡介
2. 安裝 Rails 開發環境
3. Rails 起步走
4. Ruby 程式語言入門
5. 手工打造 CRUD 應用程式
6. RESTful 應用程式
7. Active Record 基本操作與關聯設計
8. RESTful 綜合應用

Part 2: 深度剖析

1. 環境設定與 Bundler
2. 路由 (Routing)
3. Action Controller: 控制 HTTP 流程
4. Active Record: 資料表操作
5. Active Record: 資料庫遷移 (Migration)
6. Active Record: 資料表關聯
7. Active Record: 資料驗證及回呼
8. Active Record: 進階功能
9. Action View: 樣板設計
10. Action View: Helpers 方法
11. Ajax 應用程式
12. Assets Pipeline

網站開發工程師實戰營

FULL TIME | 每週五天

十週 Ruby on Rails 課程實戰訓練，帶你學會 Rails、HTML、CSS 和 JavaScript。

打通前端、後端開發任督二脈，成為全球炙手可熱的網路全端工程師 (Full Stack Developer)，或親手實現創業的 idea！



瞭解更多

十週內打造全方位的 FULL STACK 開發能力

你將會學到

- ✓ Ruby 程式語言
- ✓ Rails 網站開發框架
- ✓ 前端網頁設計
- ✓ 版本控制開發流程
- ✓ 測試驅動開發
- ✓ Web API 設計
- ✓ 網站效能調校
- ✓ 網站安全

軟體(產品)開發流程

今天的主題

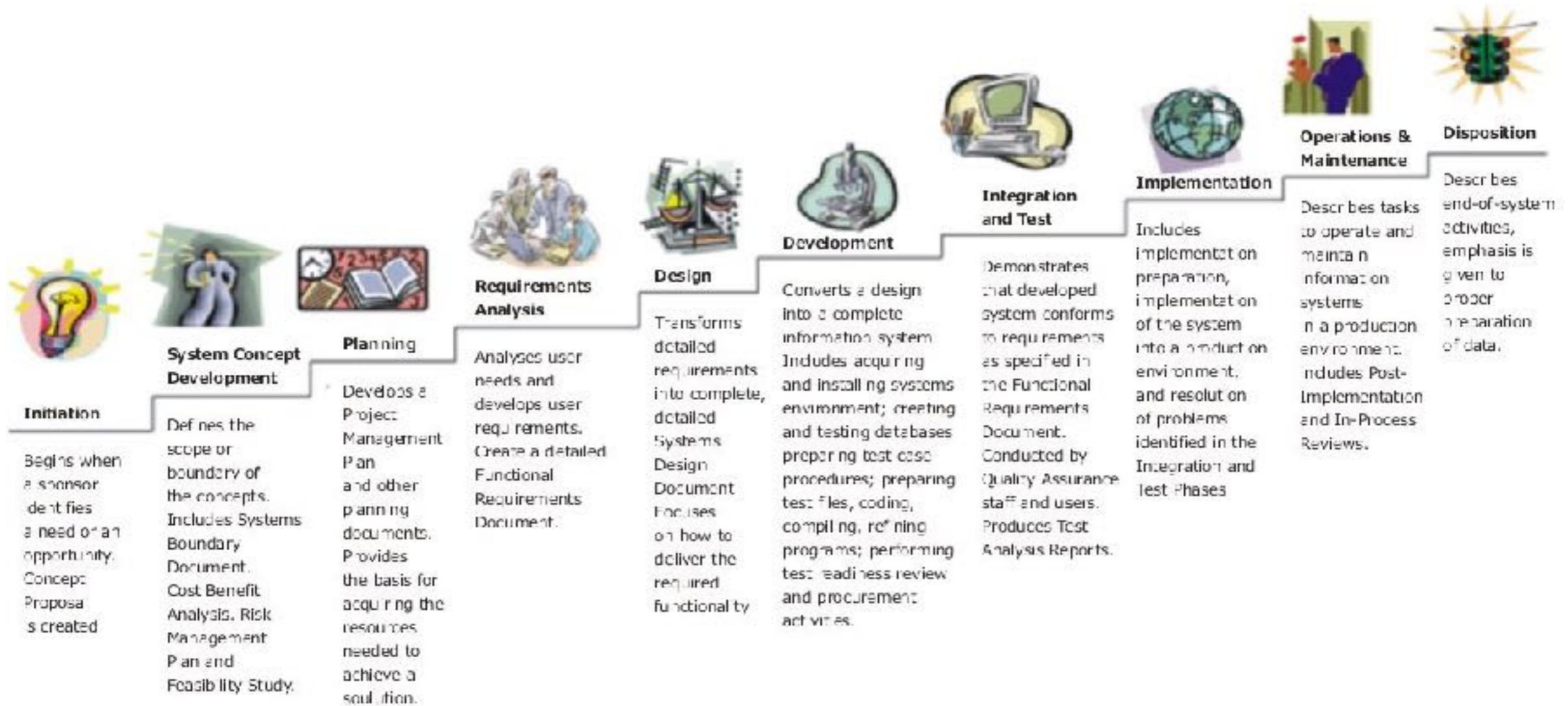
Warning: 主觀意見

請自行判斷哪些是我個人意見，哪些是事實

(理論版本)

Systems Development Life Cycle (SDLC)

Life-Cycle Phases



https://en.wikipedia.org/wiki/Systems_development_life_cycle

(今天我的版本)

software development life cycle

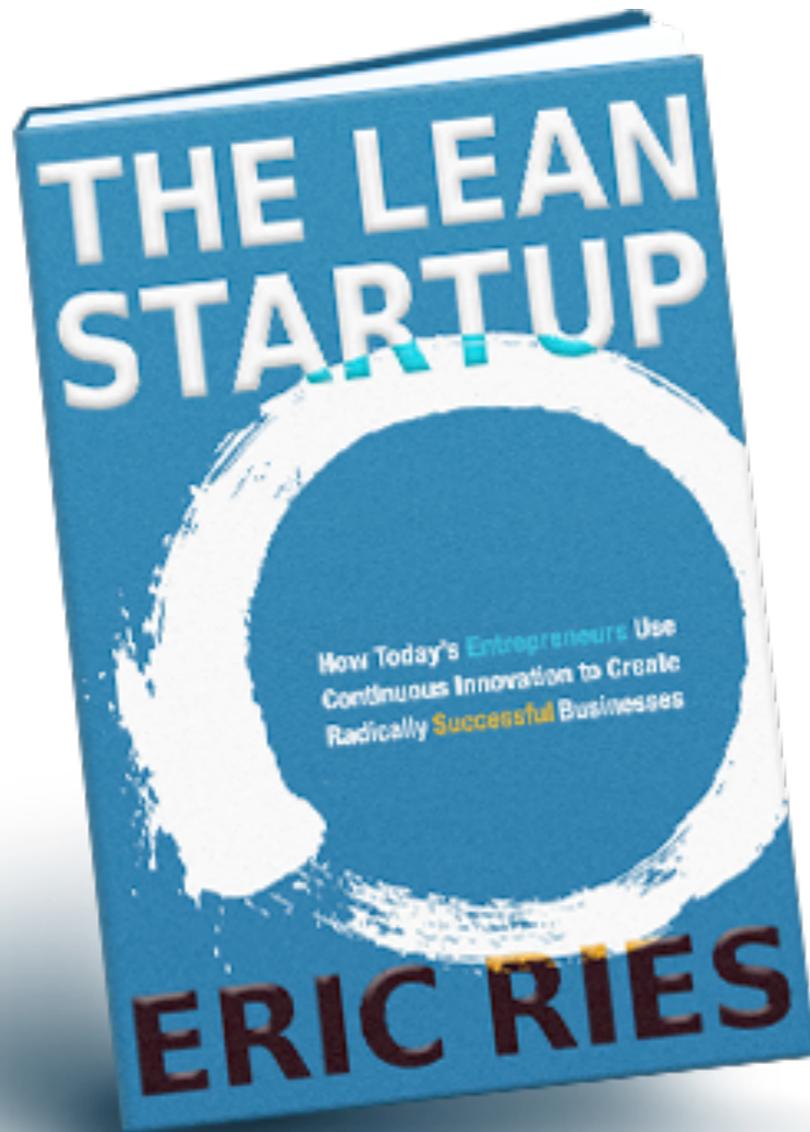
- 需求收集 Requirement
- 實作開發 Implementation
- 部署上架 Deployment
- 營運成長 Operations

1. 需求收集

一個產品的敏捷開發故事

- 老闆有一個 idea，PO 整理需求
- 採用 Scrum，開始跑 iterations 6個月
- 釋出 beta 版，進行早期用戶訪談
- 得到更多 feedback，需要做更多功能，這些用戶也很開心
- 努力工作，每次 iteration 都很完美，陸續完成這些功能，產品功能非常強大
- 增加不同營利模式：加廣告、BD合作....etc
- 媒體發布，下 fb 廣告，註冊用戶數極速增加
- 幾個月後，註冊量下跌，一不打廣告，就沒人註冊
- 活躍用戶呈現一攤死水

Lean Startup



精 實

〈用小實驗玩出大事業〉

創 The Lean Startup 業

How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses

艾瑞克·萊斯
Eric Ries

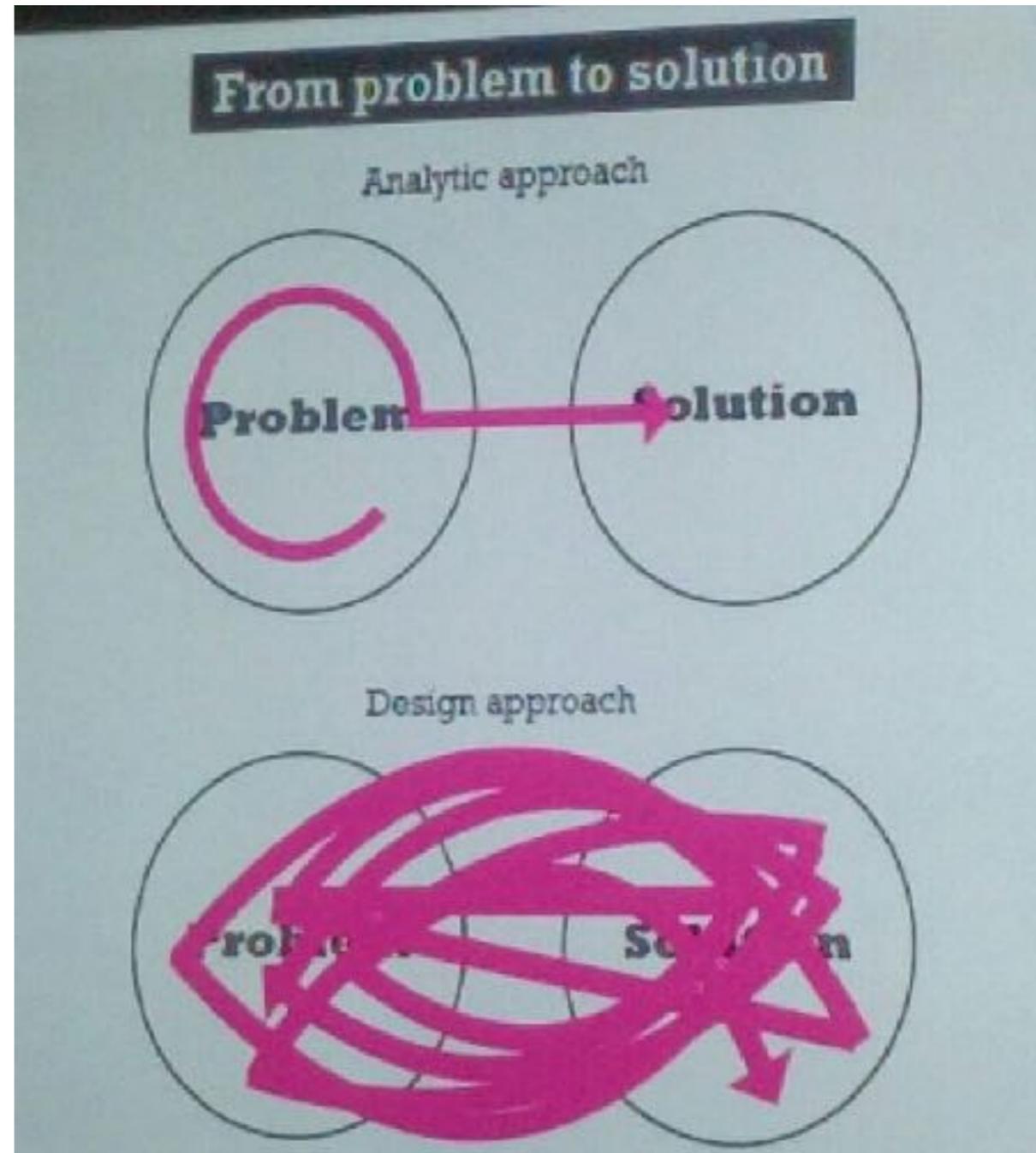


Lean Startup 101

- 一套管理創業過程的方法
- 透過 Build -> Measure -> Learn 的循環過程，快速、低成本的科學實驗方法，不斷調整來驗證產品跟市場的契合度
- 開發任何功能前，都必須先有假設，然後當做實驗來驗證學習。任何和學習無關的功能開發都是浪費。
- 直到找出 Business Model 營利模式，才開始進行規模擴張。



From Problem to Solution

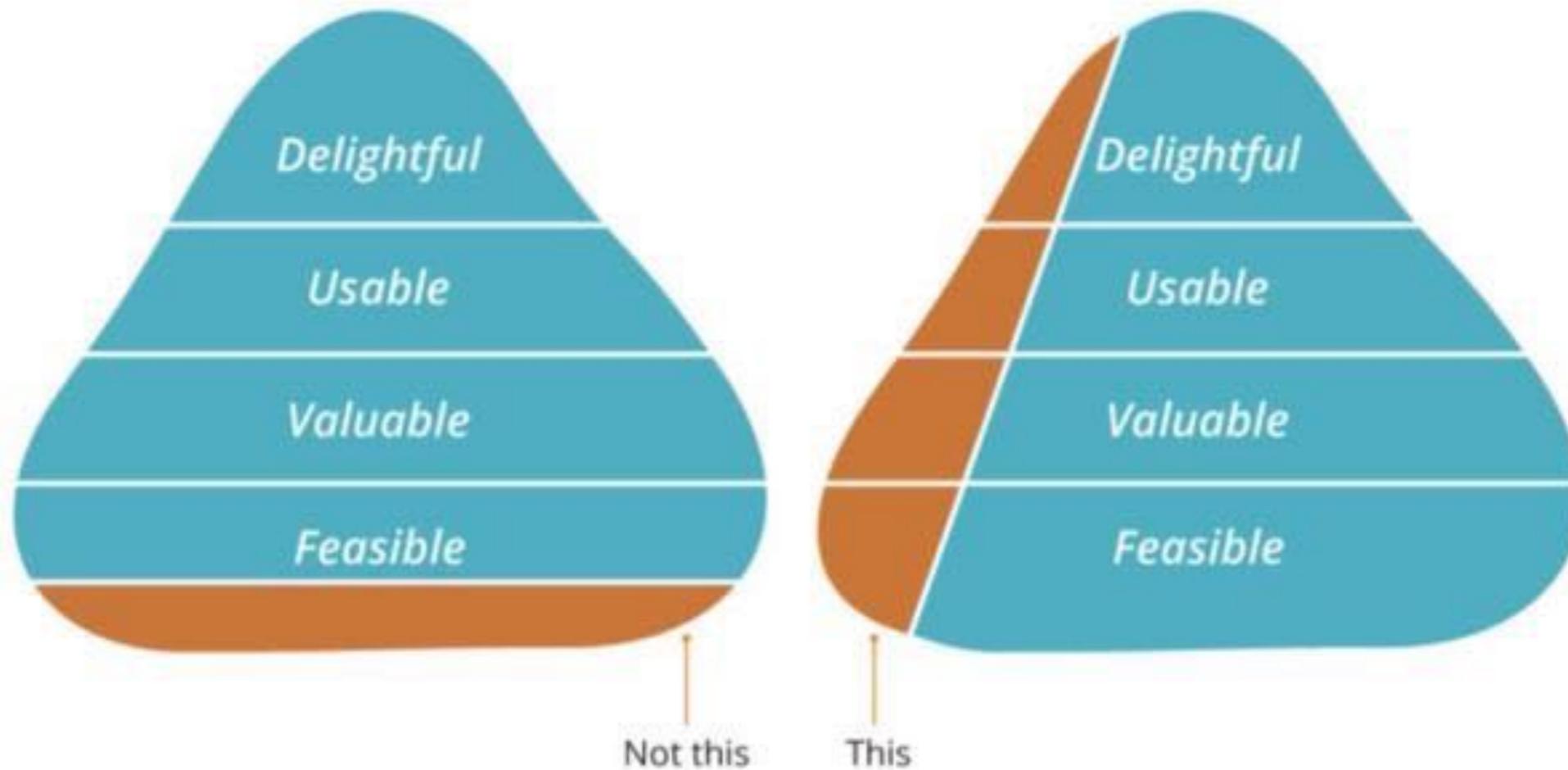


<https://twitter.com/leanplc/status/702910036243320832>

Lean Startup (cont.)

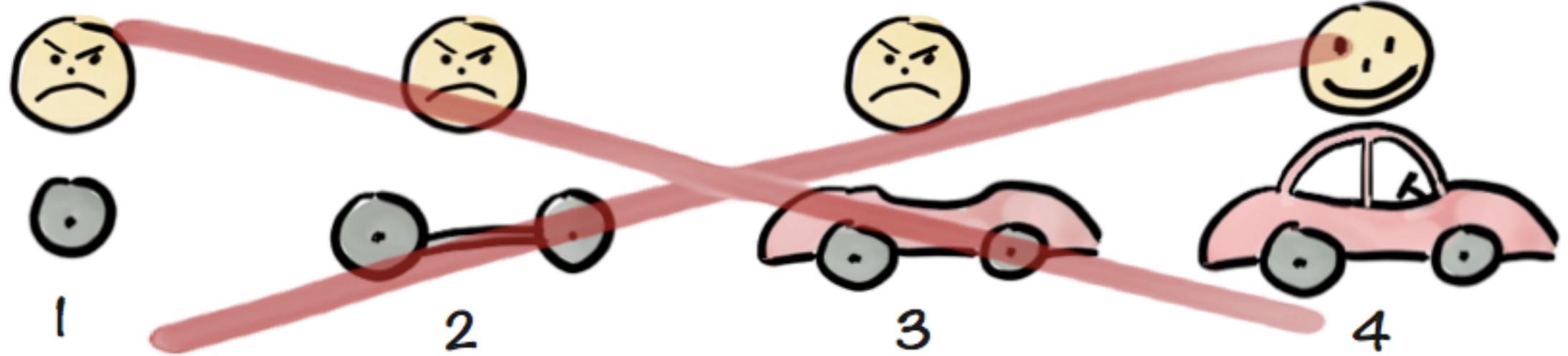
- 敏捷和精實開發強調開發有效率、不浪費
- 但是最大的浪費是做出沒人用的東西
- Lean Startup != 創業才能用，只要開發新產品，就值得一看

MINIMUM VIABLE PRODUCT

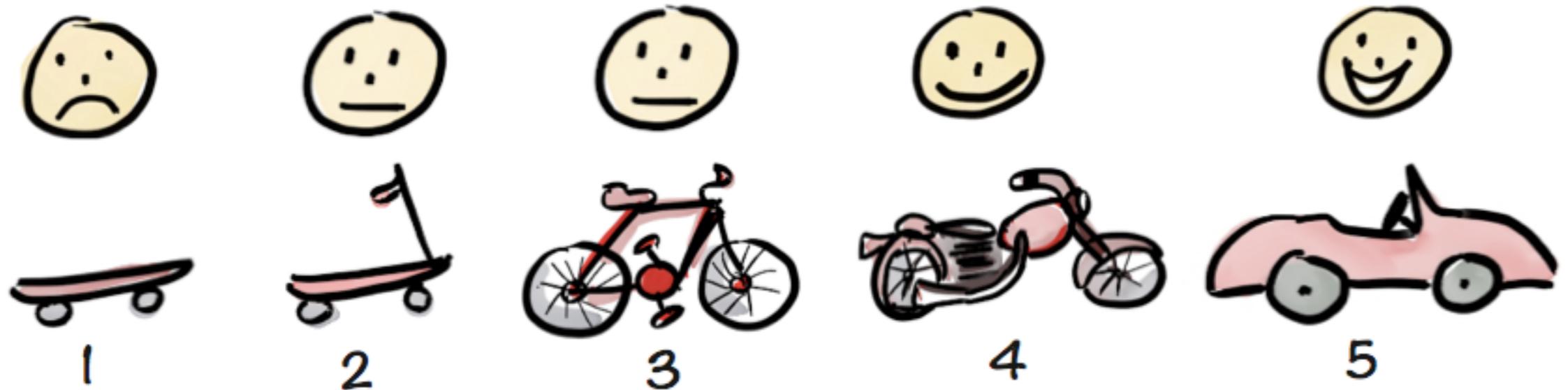


優先開發最核心的功能
1~3 個月內就第一次上架部署

Not like this....



Like this!



by Henrik Kniberg

Life's too short to build
something nobody wants

– Ash Maurya, Running Lean 作者

使用 User Stories 撰寫文件

- User Story(使用者敘述)是一段簡單的功能敘述，以客戶或使用者的觀點撰寫下有價值的功能 (functionality/feature)

<https://ihower.tw/blog/archives/2090>

範例

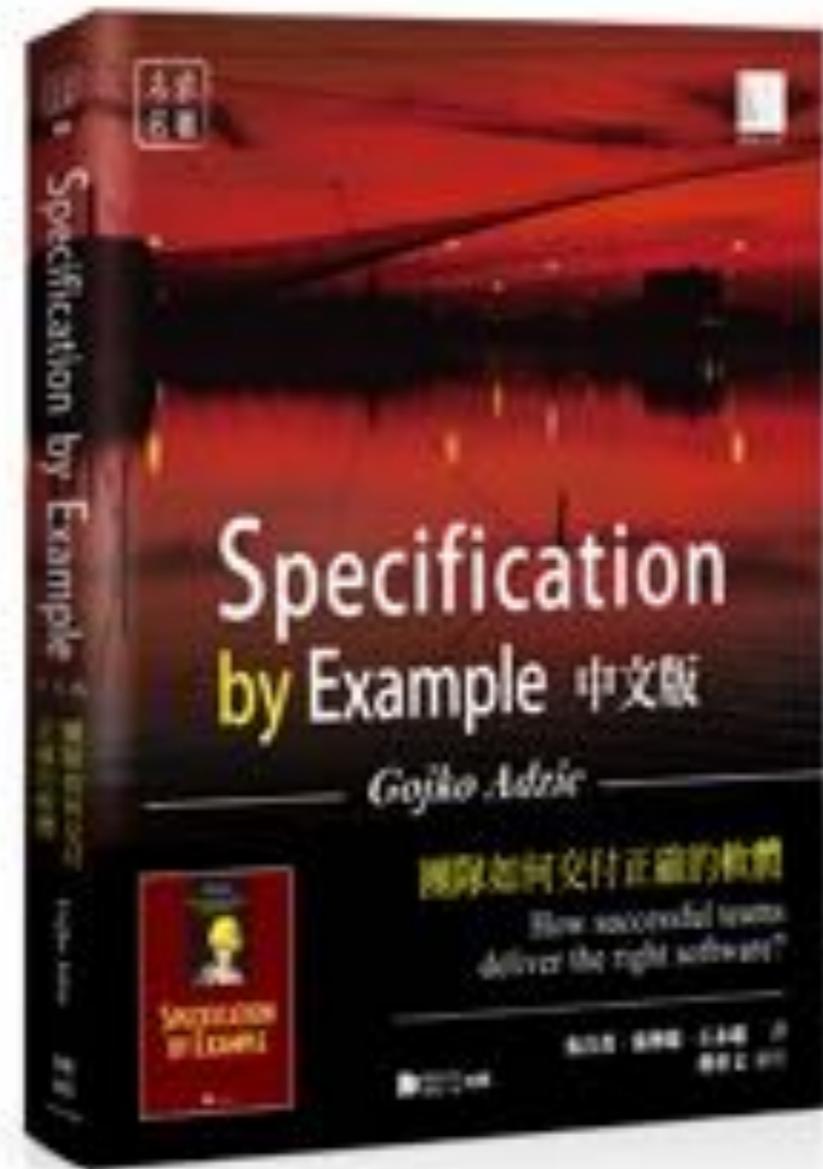
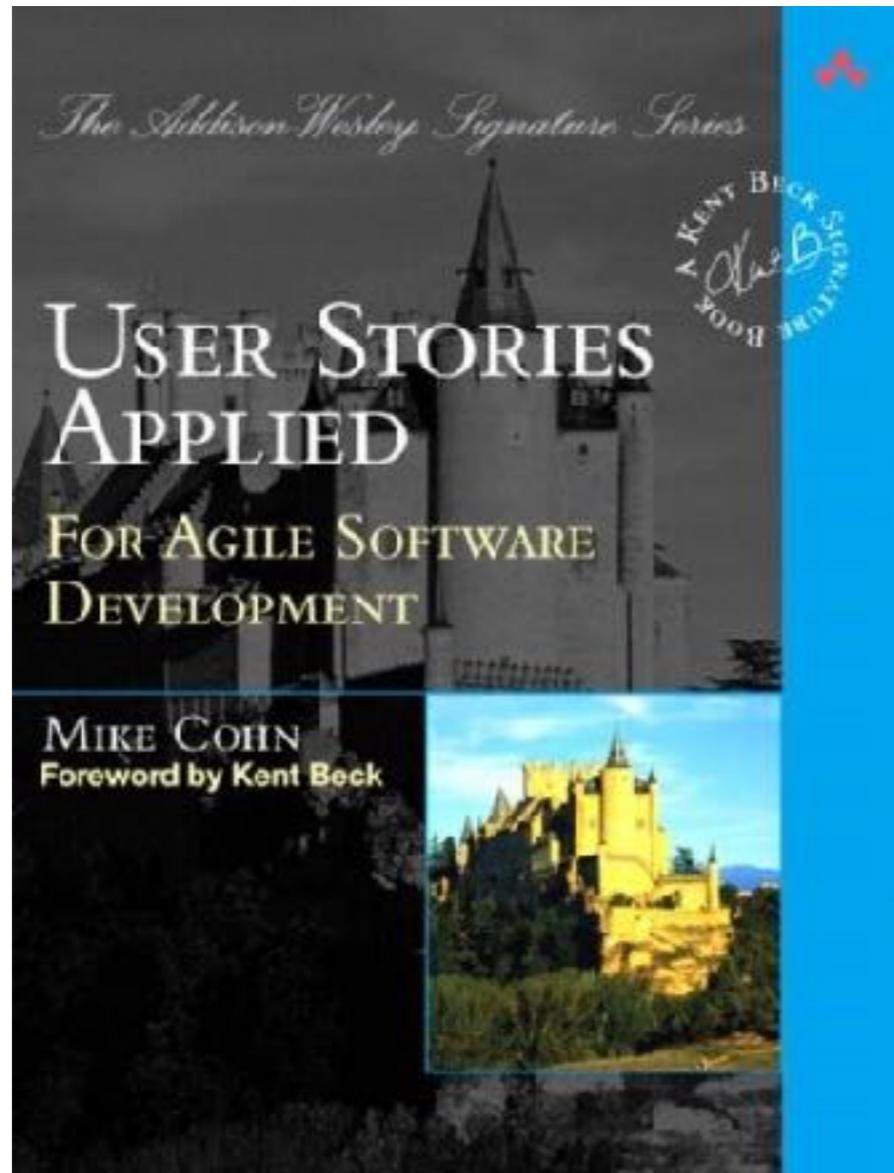
- 使用者可以在網站上張貼履歷
- 使用者可以搜尋有哪些工作
- 公司可以張貼新工作
- 使用者可以限制誰可以看到他的履歷

樣板

- 作為一個 (某個角色) 使用者，我可以做 (某個功能) 事情，如此可以有 (某個商業價值) 的好處。

只描述對使用者 有價值的功能

- 反例：這個軟體由 Java 撰寫 或 這隻程式將透過 connection pool 與資料庫連結
- 重新思考：這些技術規格帶來的商業價值是什麼，然後重新用 user stories 不帶技術詞彙表達出來



善用工具寫文件

- Hackpad 或 Quip
- 即時多人協作
- 寫文件必備，讓你文思泉湧



Create and communicate in one place.

Quip is the modern productivity suite that helps your team get work done faster.

Work email address

Sign Up or Sign In



live demo

小結

- Lean Startup: MVP
- 用 User Stories 描述
- 善用線上工具 Quip 或 Hackpad

2. 實作開發

專案管理

- 將 User Stories 分派給開發人員，並追蹤開發進度
- 在有限的時間跟資源下，讓團隊可以高效率開發出成果

專案管理 (cont.)

- 先講目標：
 - Frequent, early, small release 盡早、頻繁的小規模釋出可以運作的軟體
 - 因為透過經常性地建構出可以實際使用的軟體，我們持續得到使用者 feedback，才可以確保方向正確
 - Fail often fail early 是最有效率降低發佈風險的辦法

假設你有這樣的陣容



開發流程的演進

- Milestone-based , Waterfall 式流程
- Scrum , Agile 式流程
- Kanban , Lean 式流程

Milestone-based

The screenshot shows a web browser window with the URL <https://p.passionbean.com/versions/6>. The page is titled "App-iOS-1.0" and features a navigation bar with buttons for "網站首頁", "帳戶首頁", "專案清單", "網站管理", and "說明". The user is logged in as "ihower".

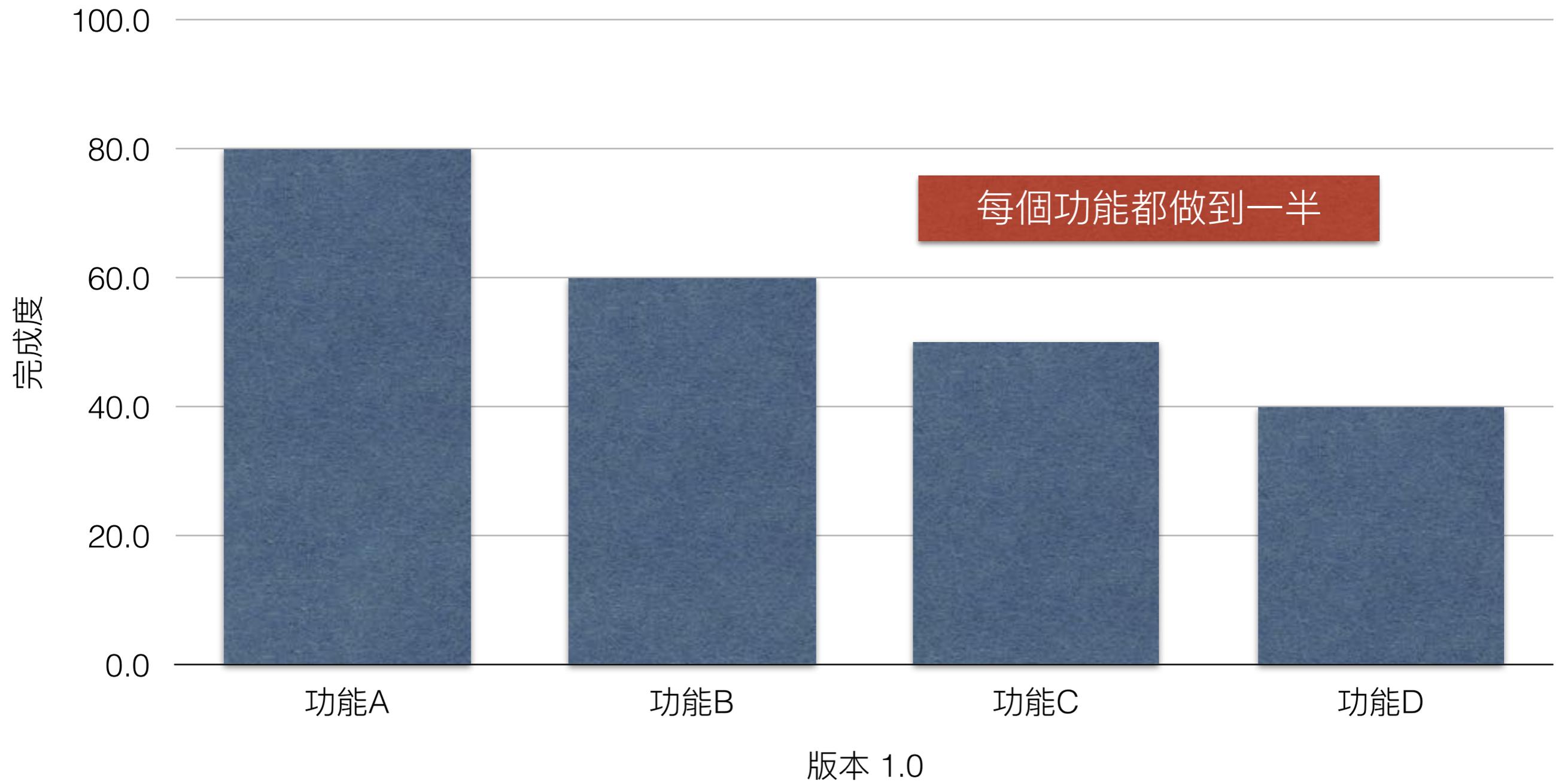
The main content area displays a progress bar for "App-iOS-1.0" with a completion rate of 97%. A red box highlights the text "306 個任務...". Below the progress bar, there is a list of related issues (Story-Bug) with titles such as "Story-Feature-功能-#1055: 動態和運動後資料加狀態" and "Story-Bug-臭蟲-#617: 照片上傳後沒有發佈到 Feed".

On the right side, there is a "工時追蹤" (Time Tracking) section showing "預估工時" (Estimated Time) and "耗用工時" (Used Time) as 0.00 小時. Below this is a "問題按" (Issue Filter) section with a "登入標籤" (Login Tag) and "分組顯示" (Group Display) options. The filter shows a list of categories with their respective counts: Story-Feature 功能 (1/1), Story-Bug 臭蟲 (31/31), Task-Coding 開發 (250/262), Task-UI 設計 (5/5), 對外合作 (1/1), 任務 (1/1), and 文件 (5/5).

Waterfall 的問題：產品 Feedback 要等很久

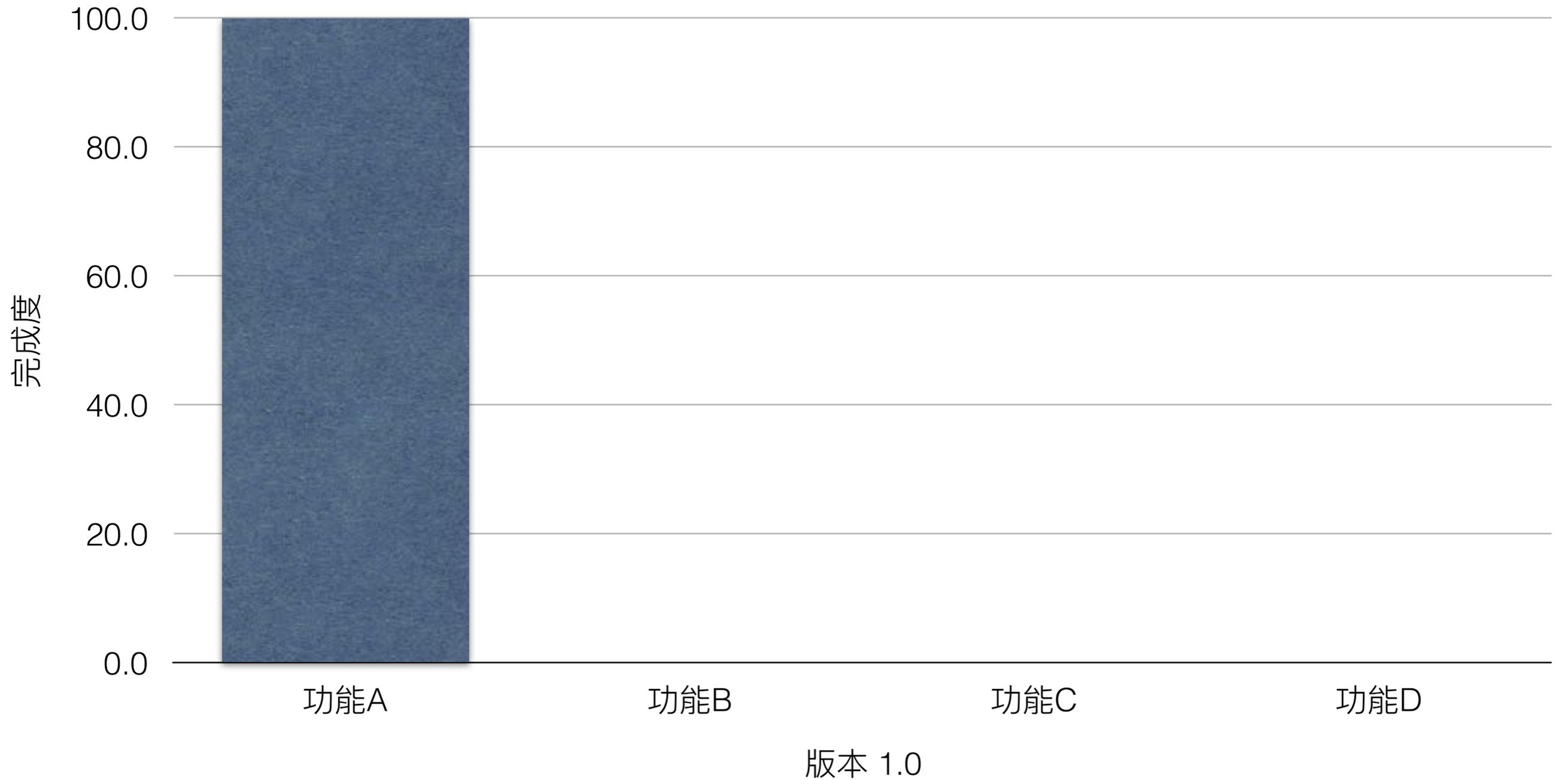
- 需要耗時數個月，才能產出可以讓使用者使用的產品
- 總是無法在預計的時程內完成計畫的事情
- 浪費太多時間在寫很少用到的功能
- 專案進度不明朗

要進步，先承認你就是 Waterfall 吧

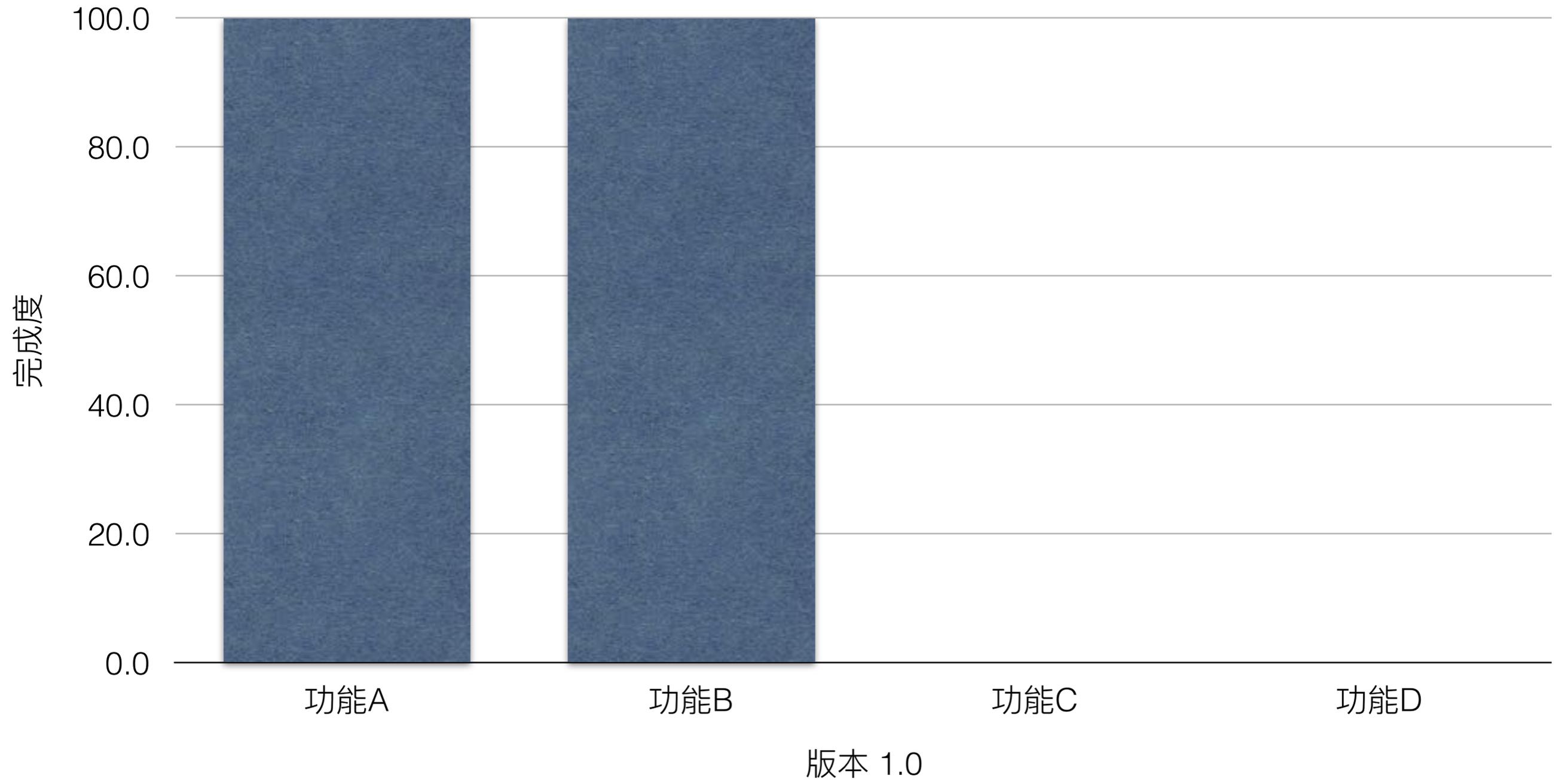


等全部功能完成才能拿到 1.0
給使用者 feedback 太慢了，
聽說敏捷軟體開發可以快速回應需求？

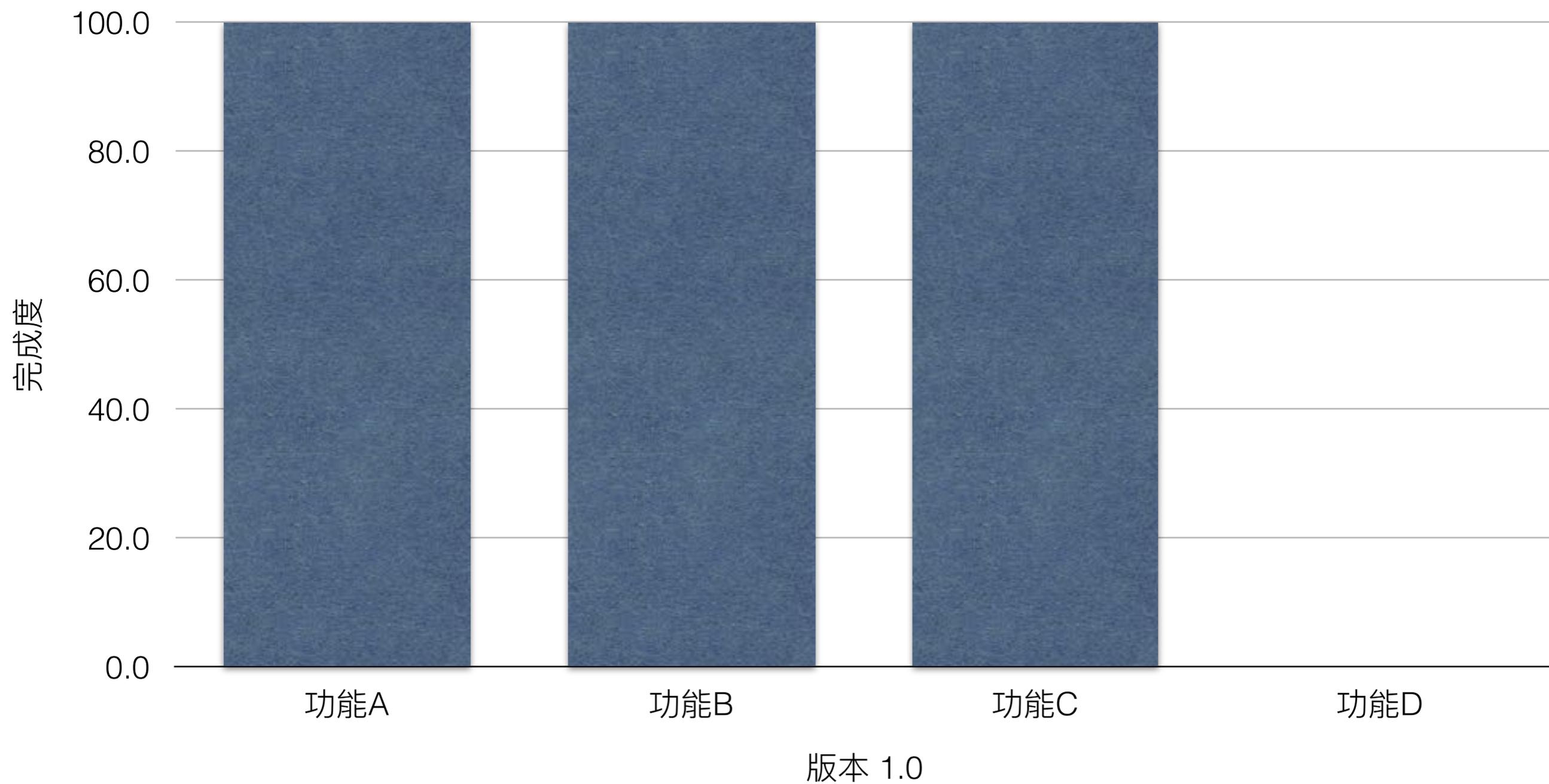
敏捷開發第一個週期釋出



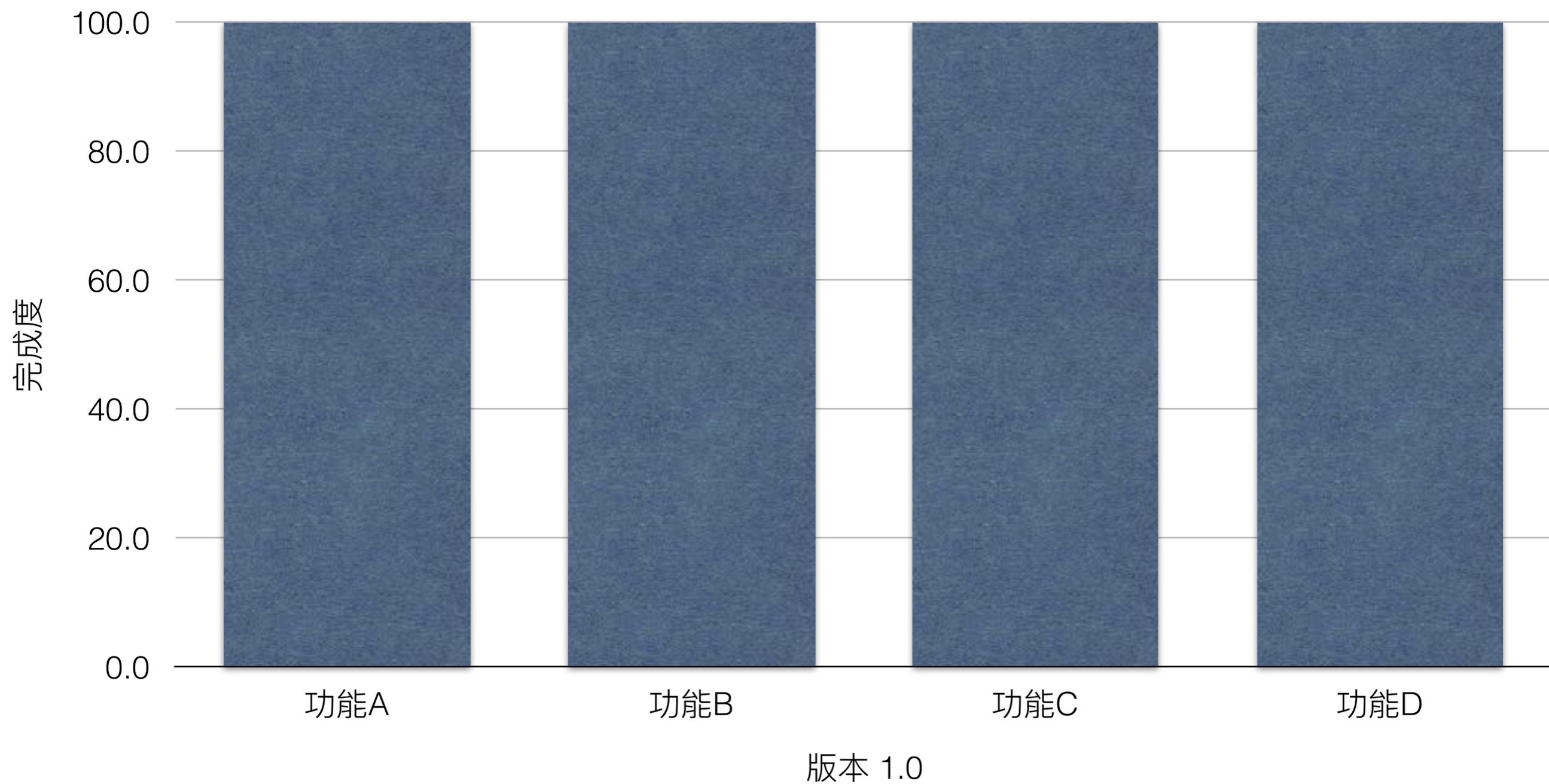
敏捷開發第二個週期釋出



敏捷開發第三個週期釋出



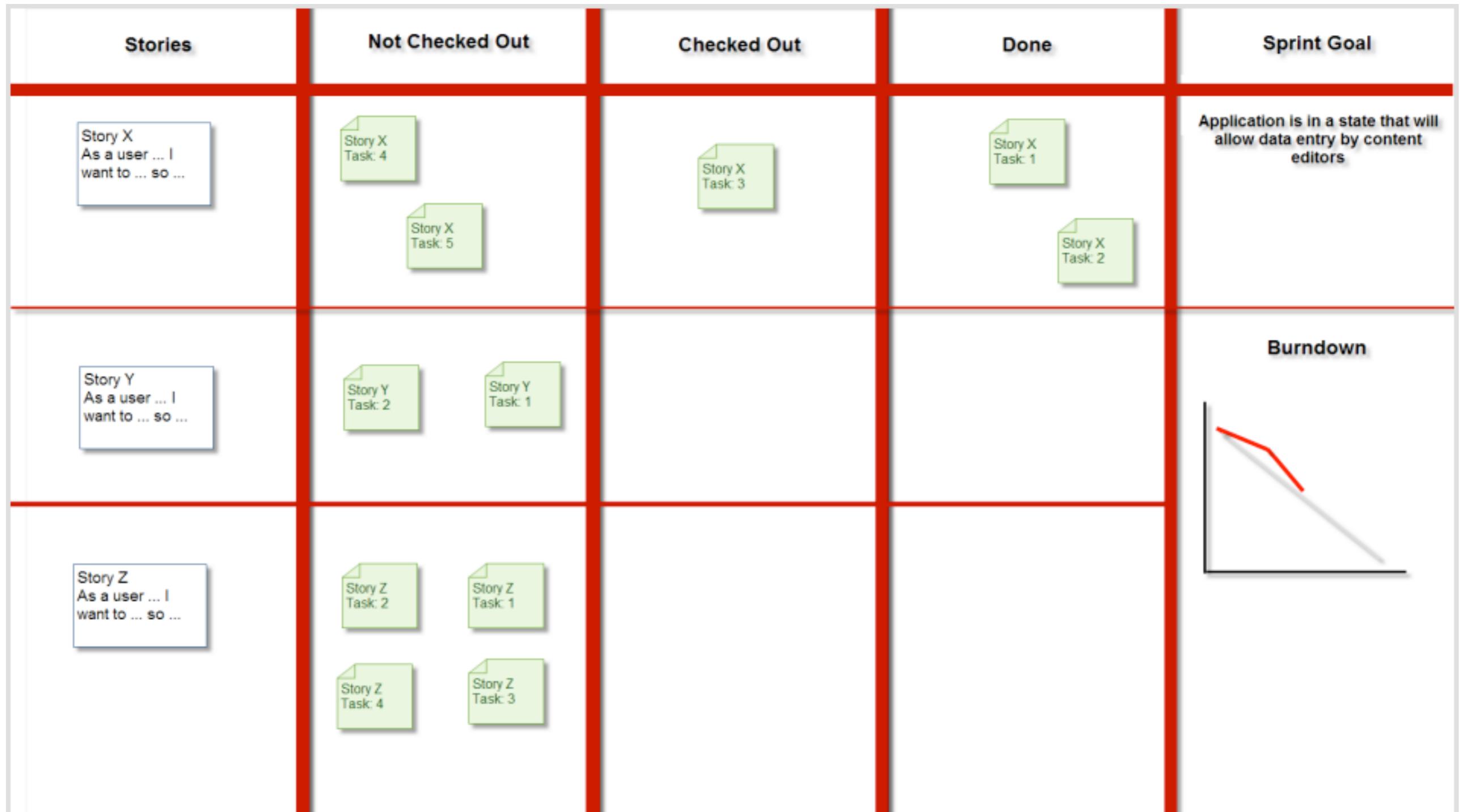
敏捷開發第四個週期釋出



導入 Scrum 敏捷流程

- 團隊角色區分成 Product Owner、Scrum Master 和 Team member
- 固定開發週期，每1~4週釋出一個可以運作的產品
- 每個週期前進行 planning meeting
- 每個週期結束進行 demo 和 retrospective meeting
- 每個週期都可以得到產品 Feedback

開發週期的 Task Board



teddysoft.tw

 **泰迪軟體** 敏捷開發、實務、課程、顧問服務 

[敏捷開發系列課程](#) [企業服務](#) [C.C. Agile 社群聚會](#) [特色商品](#) [聯絡我們](#)



從遊戲中學習看板核心精神

看板方法與圖書開發實作坊

[查看詳情](#)

 **敏捷顧問導入服務**
深入現場對症下藥
[企業導入服務](#)

 **敏捷開發系列課程**
理論 + 實作，歡笑中動手學習。
[查詢最新課表](#)

 **C. C. Agile 每月聚會**
一起來聊聊軟體開發的甘苦談。
[活動精華報導](#)

關於泰迪軟體

泰迪軟體是一家專門提供「敏捷開發顧問服務」與「教育訓練課程」的公司。若您的專案、團隊、或是個人有以下症狀，歡迎洽詢。

2016年度 [快速查詢](#)

[敏捷開發系列課表](#)

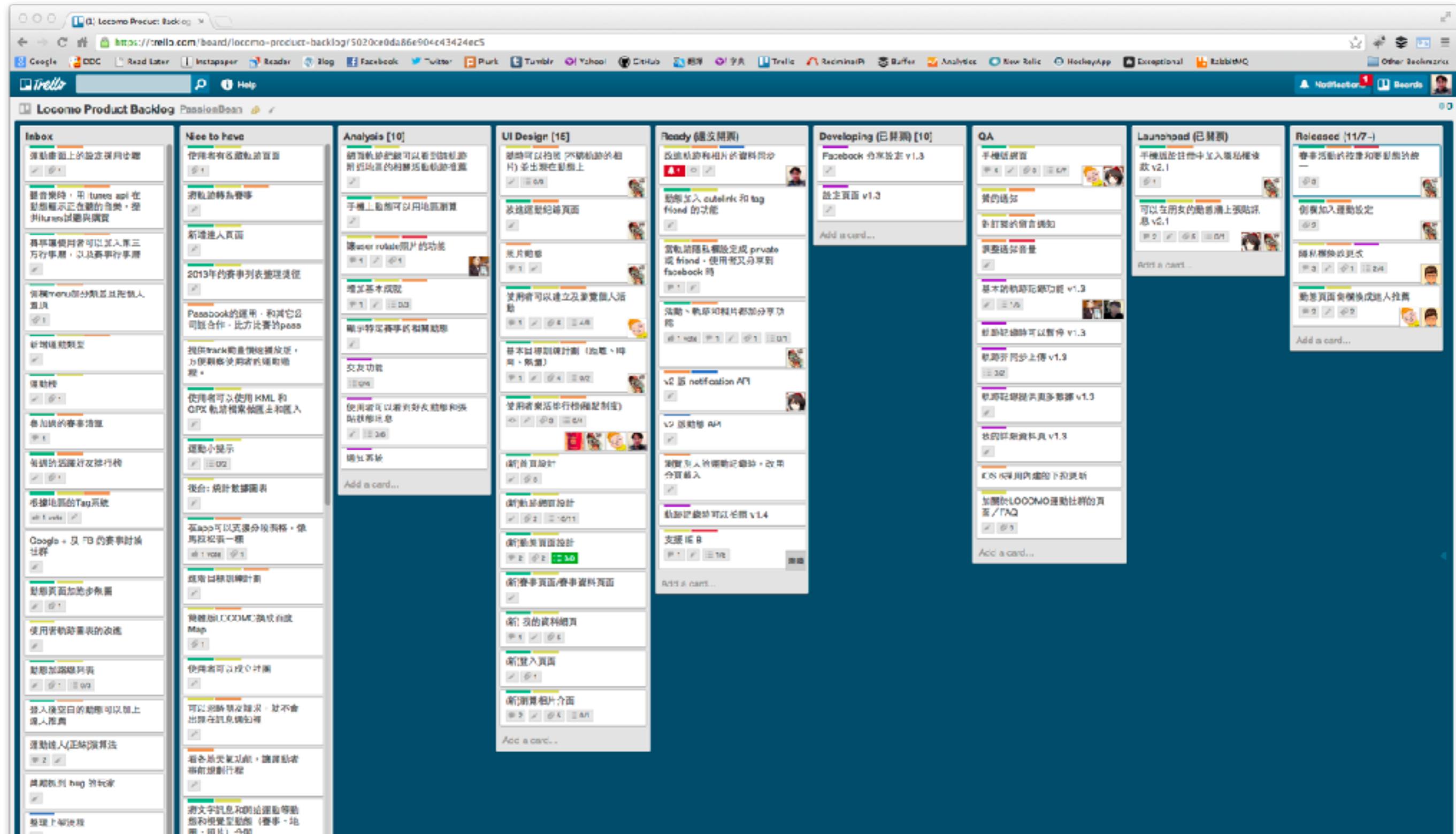
固定開發週期的問題

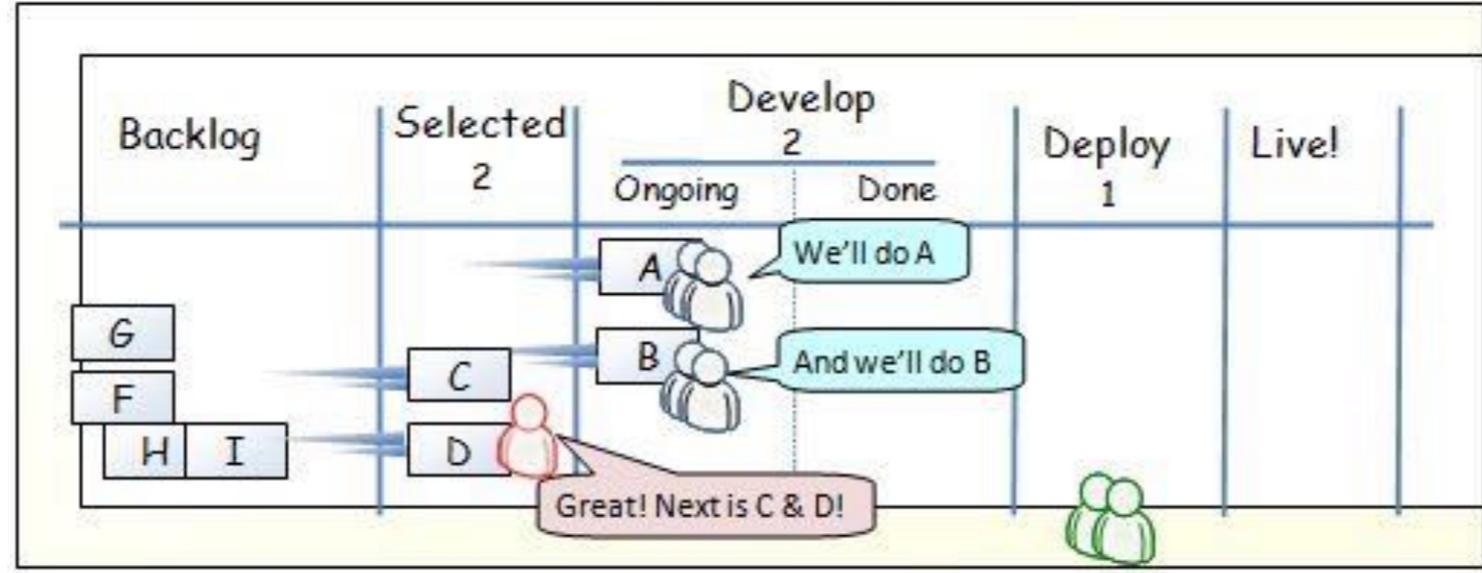
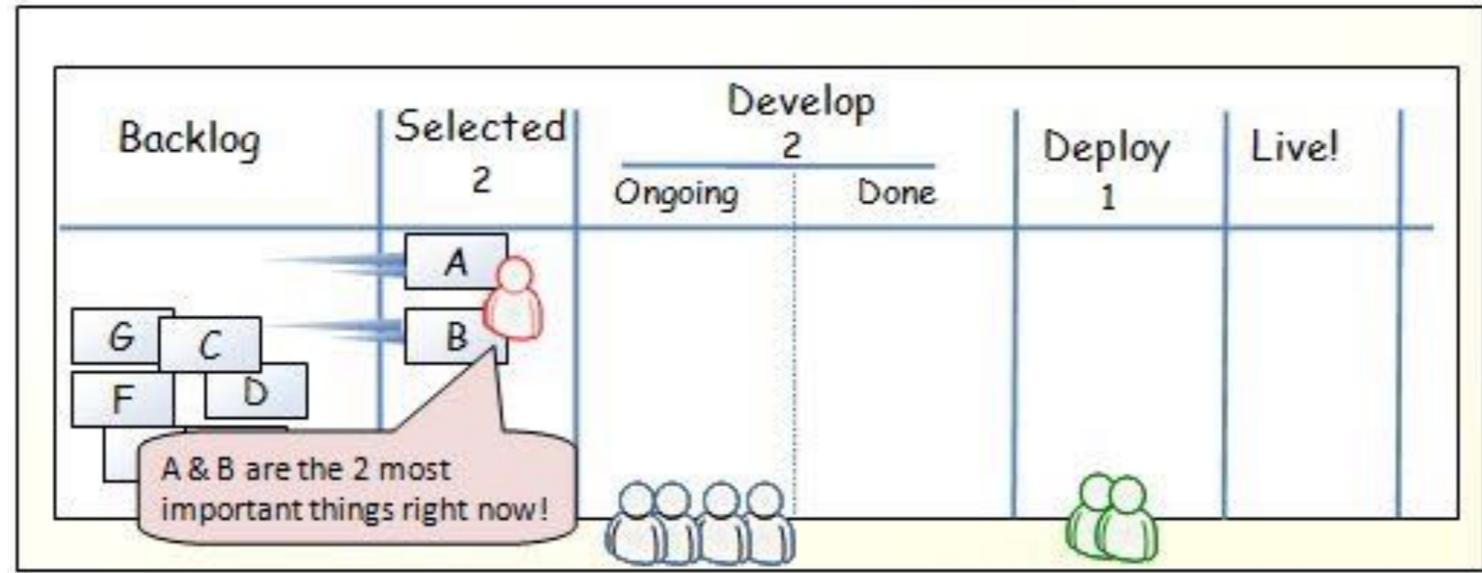
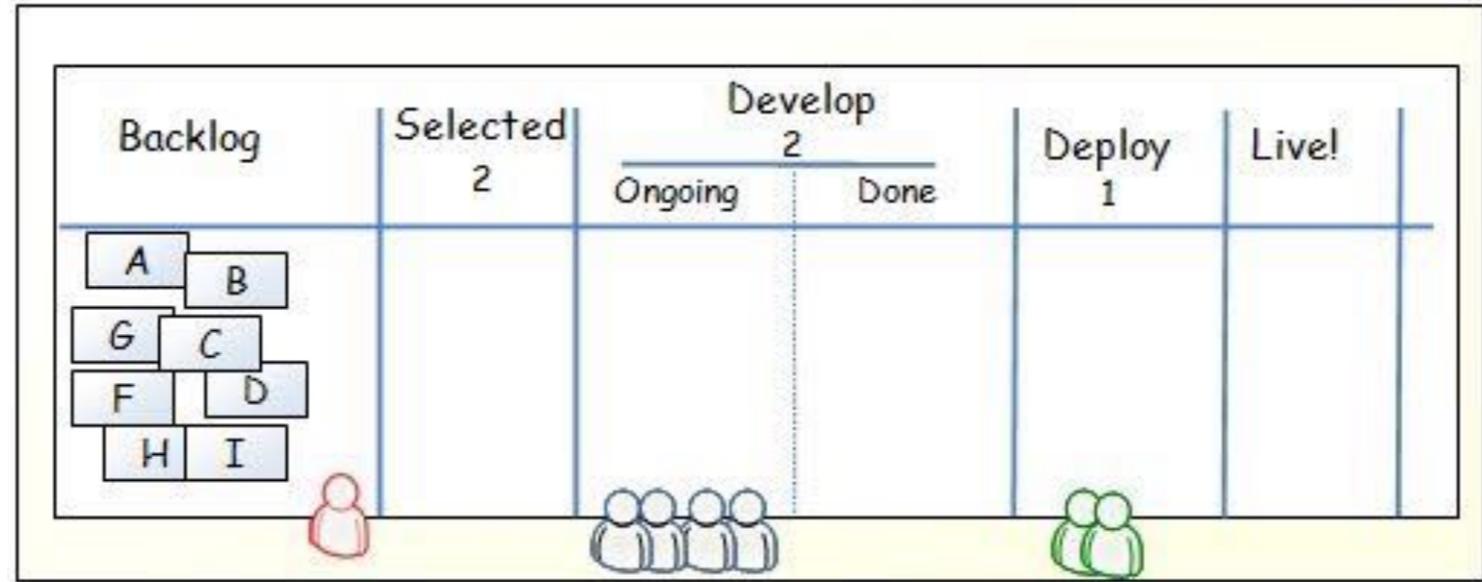
- 產品上線之後的營運，是一種 Event-driven 的模式，每天都有問題需有優先處理，不適合開發人員與營運人員合一的小型公司
 - 無法事先計畫不被打擾的週期
 - 太短的週期也不行，有的任務會超過
- UI 設計跟程式設計的開發週期搭配問題
- 不同平台(iOS, Android, Web)開發週期搭配的問題
- App Store 審核時間不一定
- 兩週的開發週期 Feedback 仍嫌長，可不可能更快？

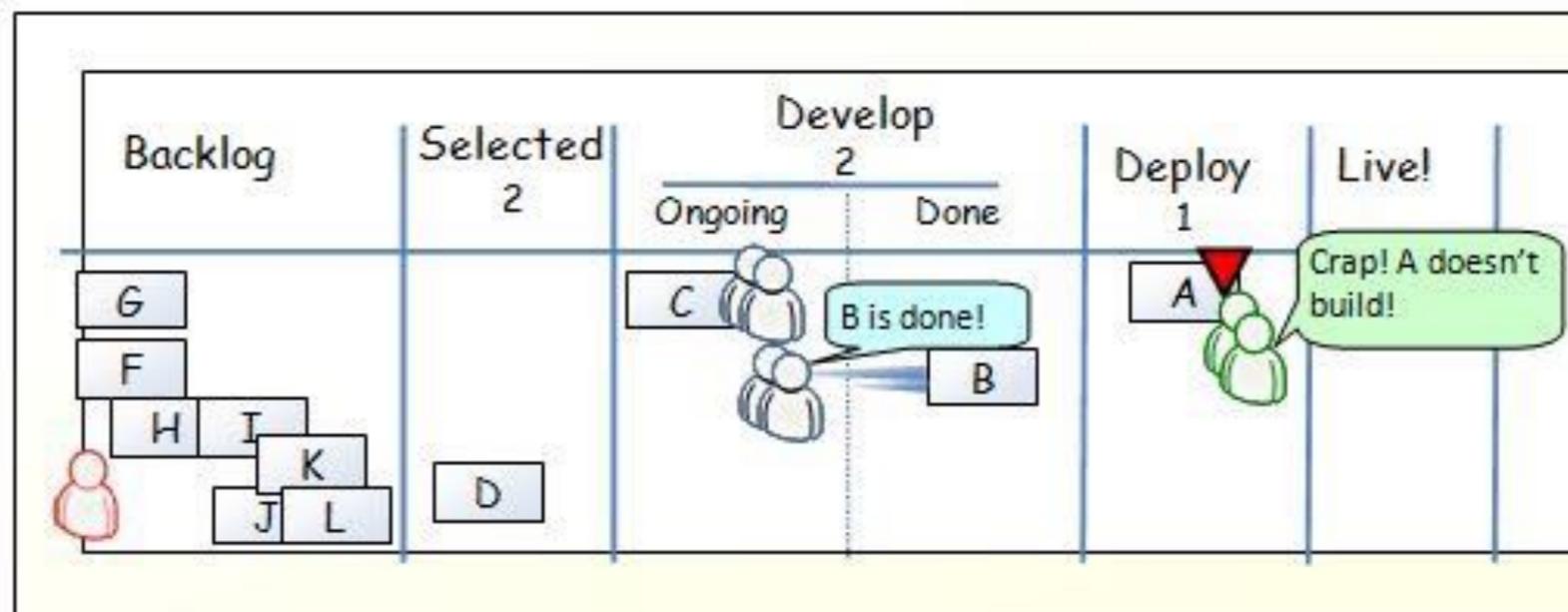
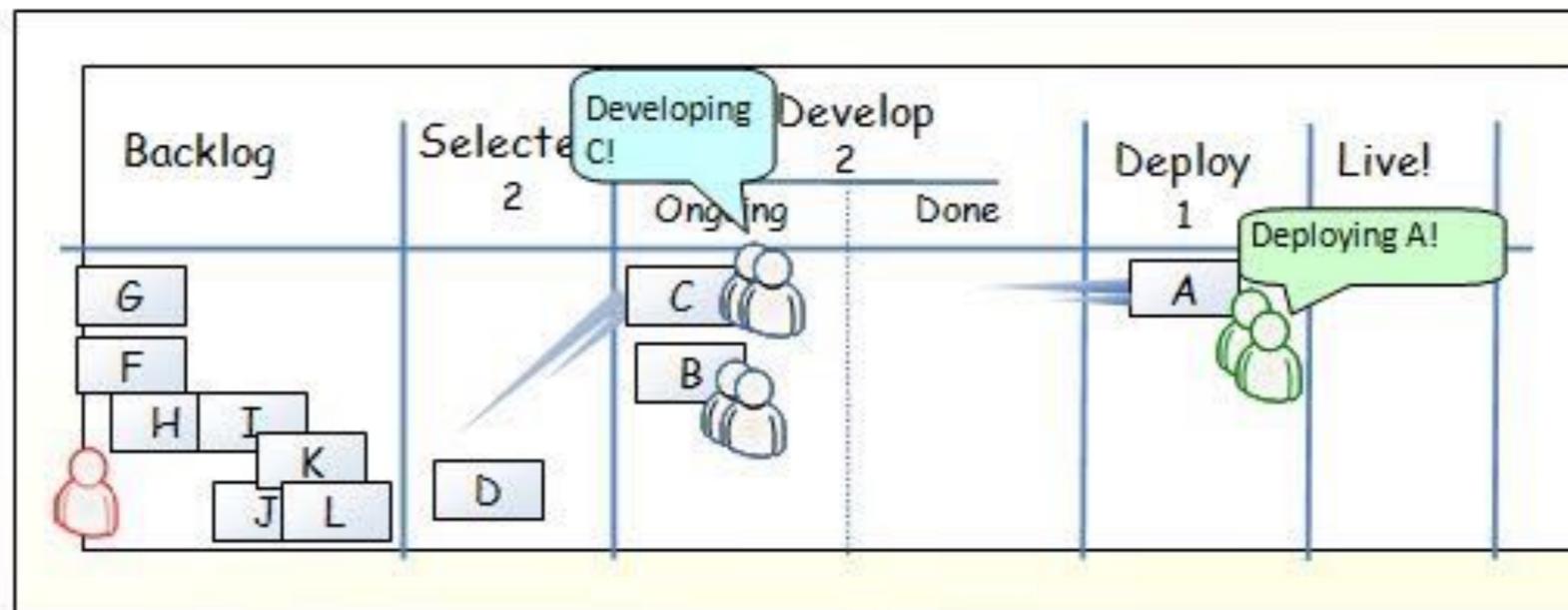
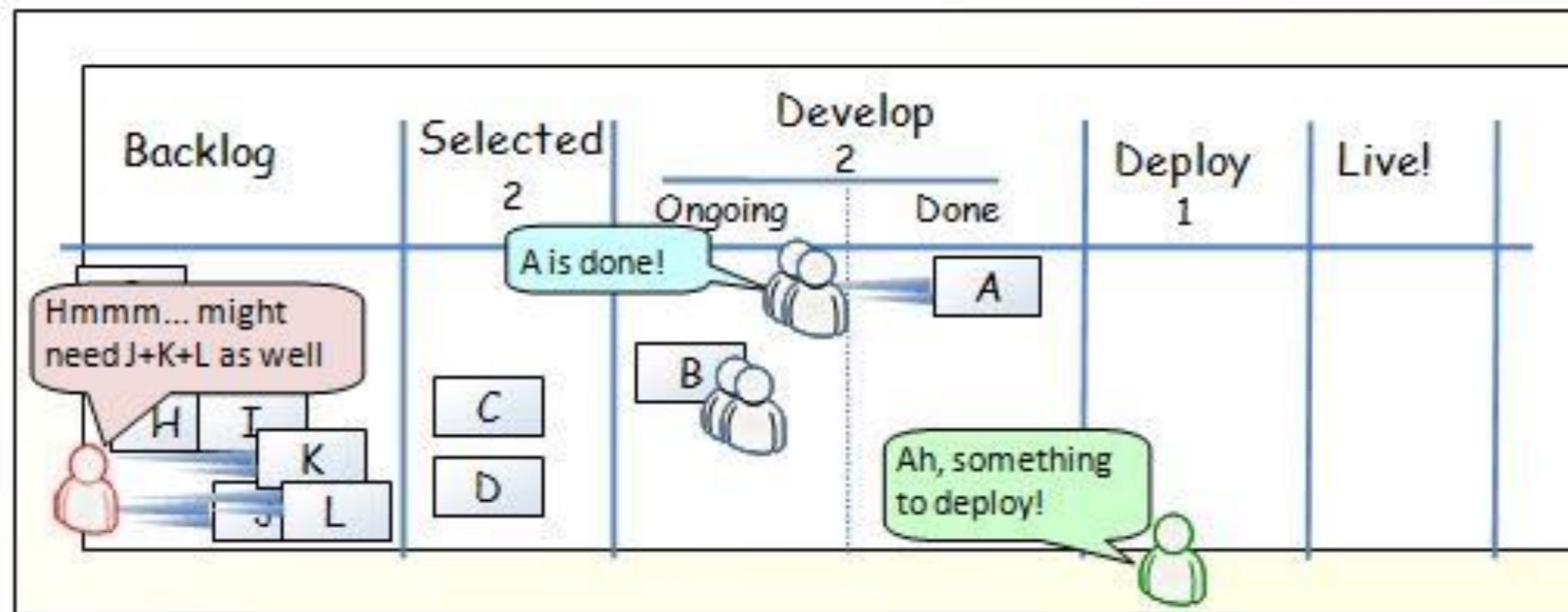
Kanban 看板系統

- 最輕量的流程管理方法
- WIP (Work in Process) 限制 (TOC限制理論)
 - 限制每個狀態的最多項目
- 關注 Cycle Time (平均每個項目的完成時間)

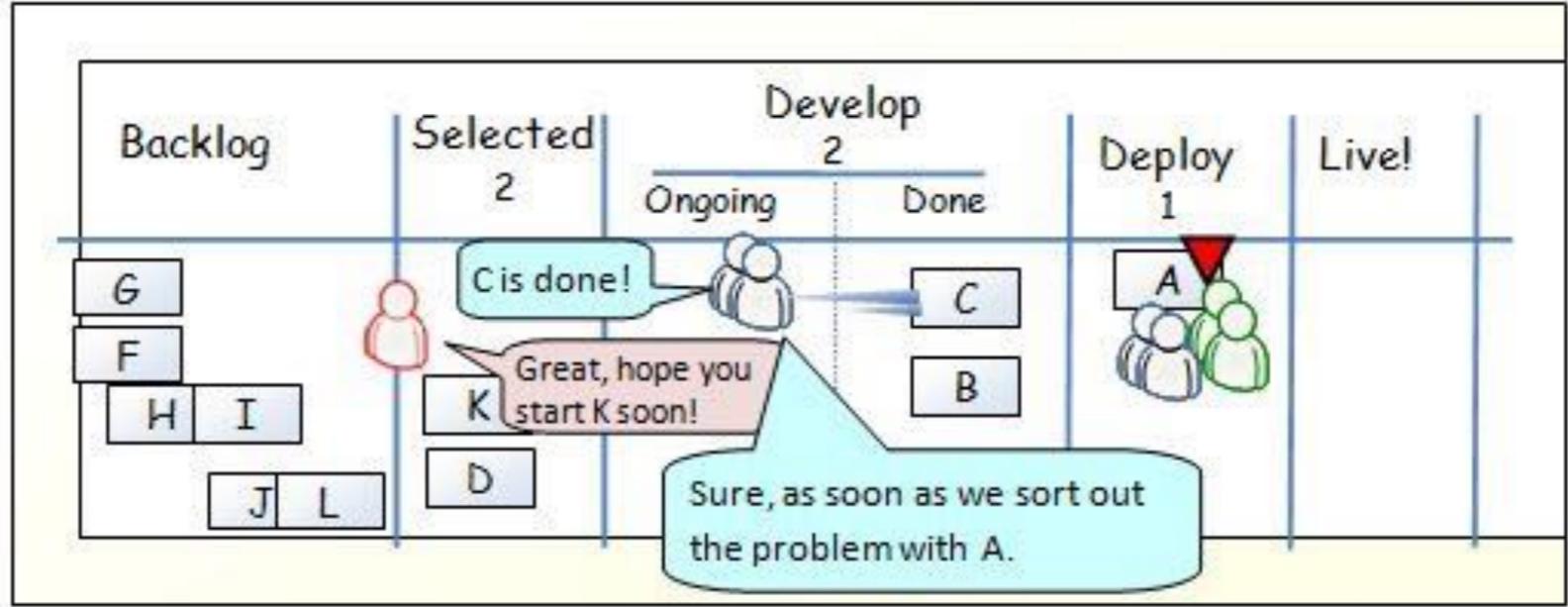
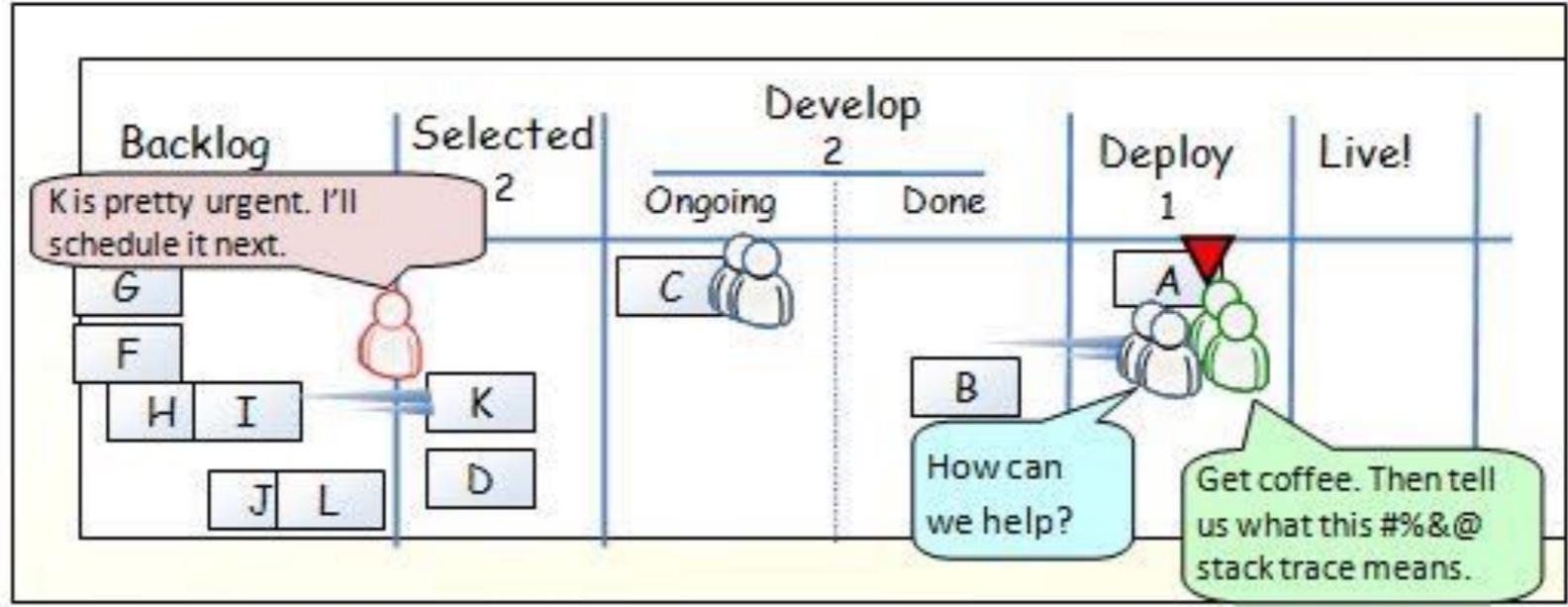
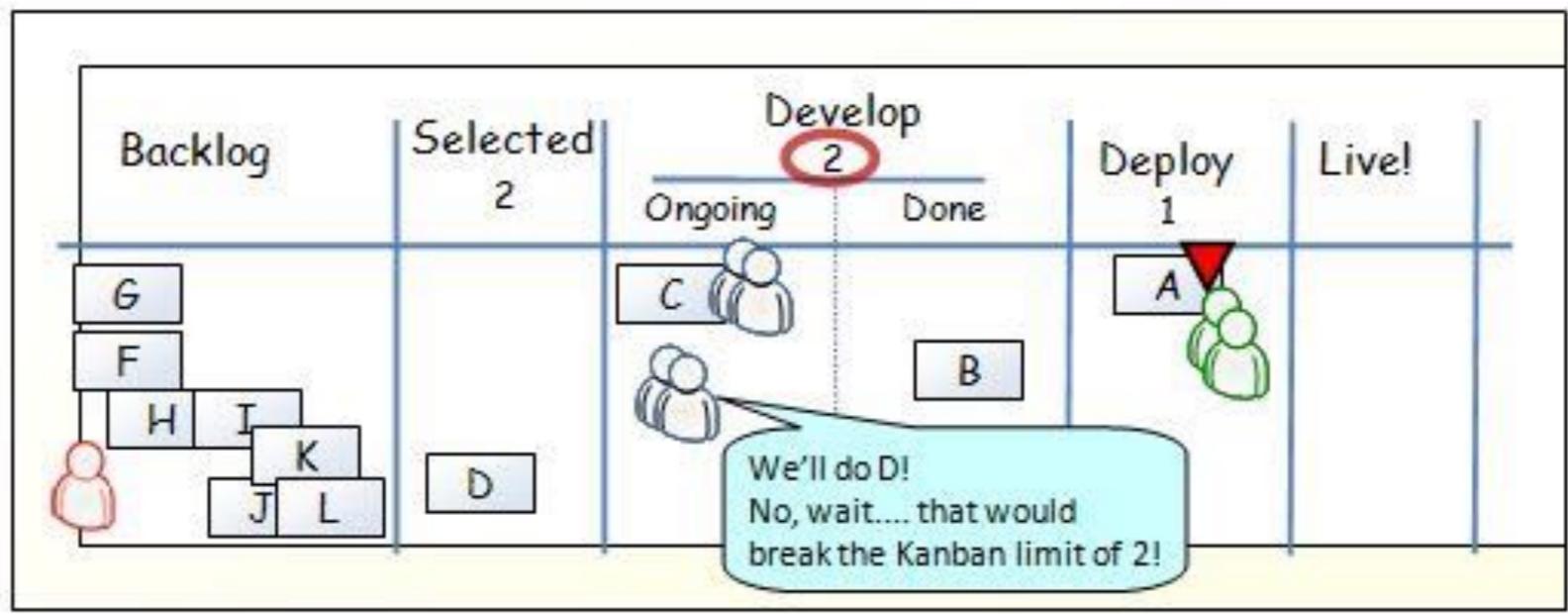
Kanban 視覺化看板系統

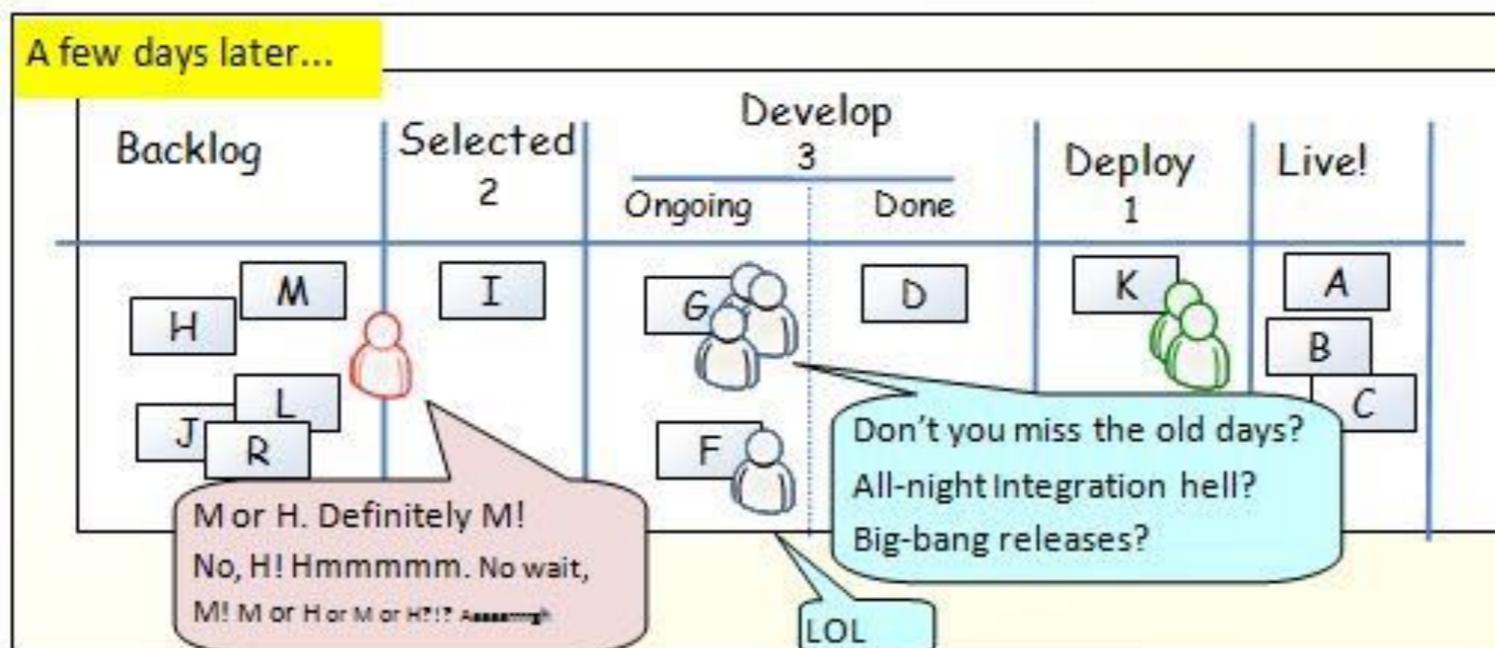
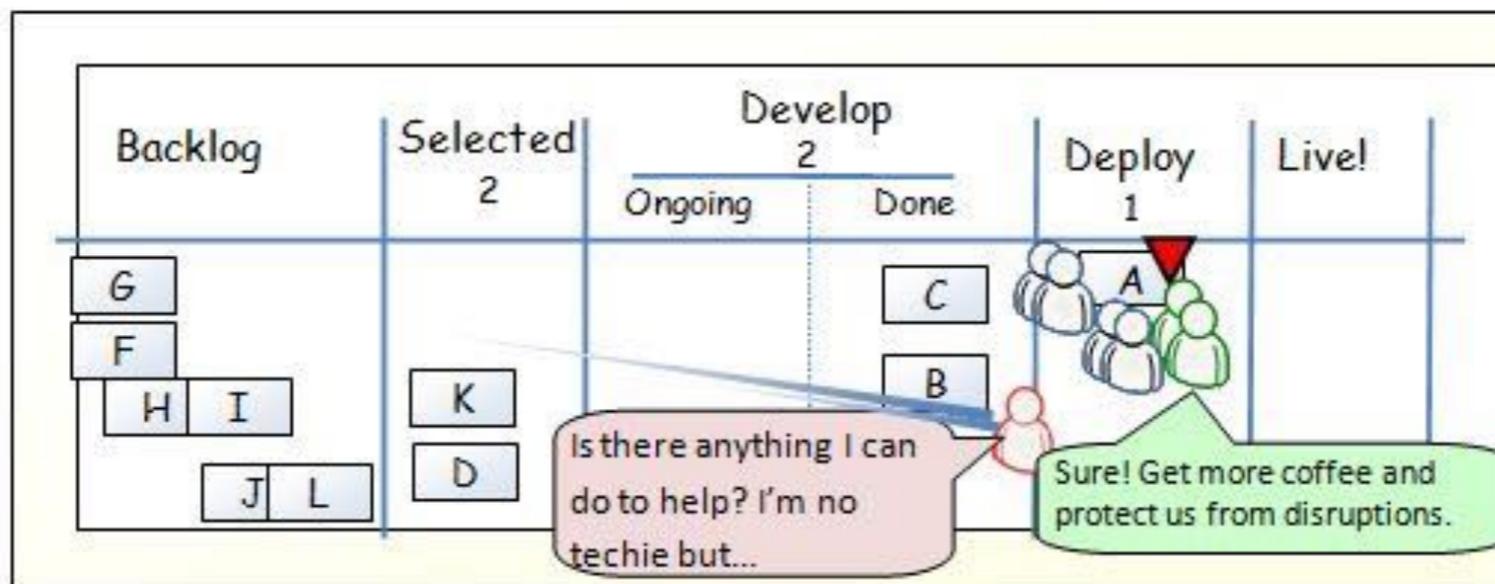
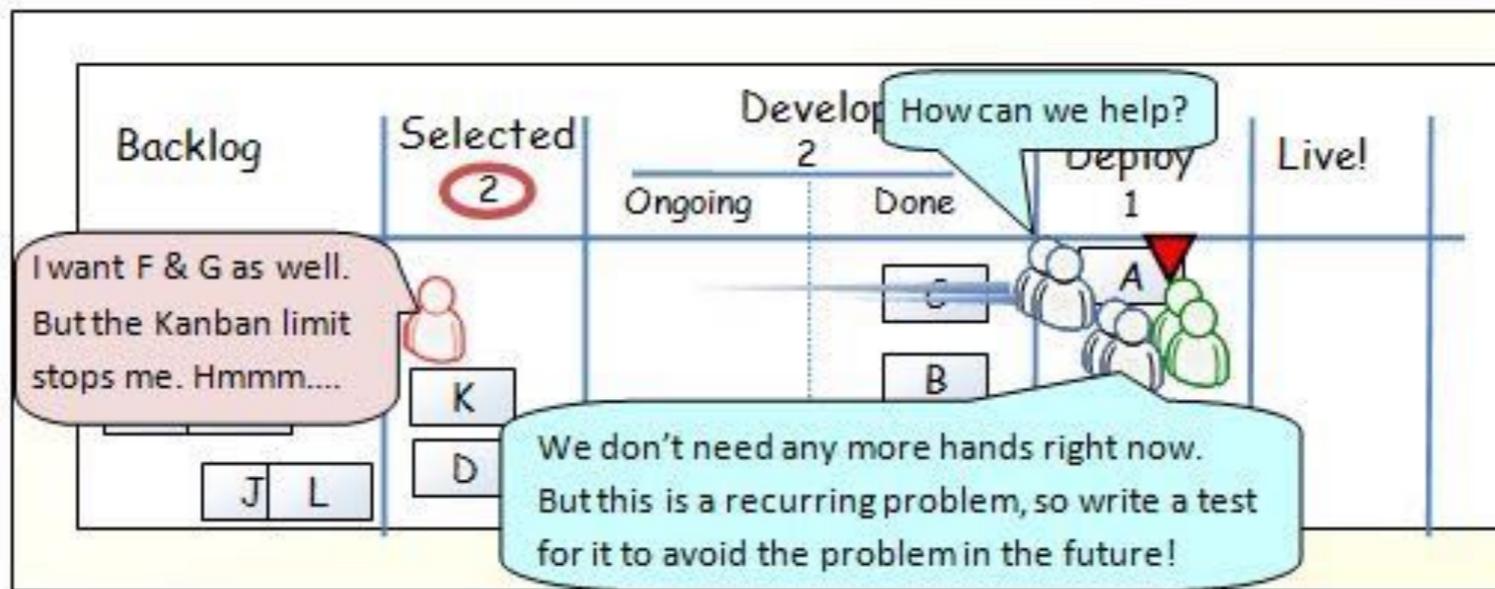


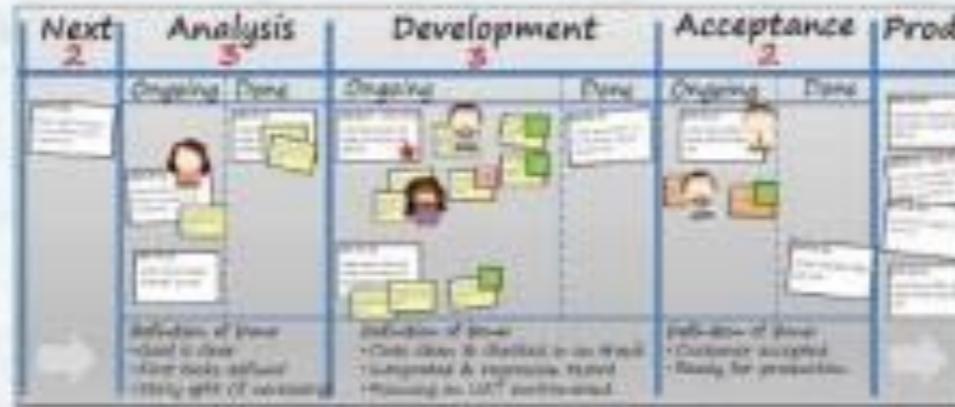




可惡，A 有問題
不能佈署







Kanban and Scrum making the most of both

Henrik Kniberg & Mattias Skarin

Forewords by Mary Poppendieck and David Anderson

ENTERPRISE SOFTWARE
DEVELOPMENT SERIES

InfoQ

<http://www.infoq.com/minibooks/kanban-scrum-minibook>

Continuous Deliver

(或 Continuous Deployment)

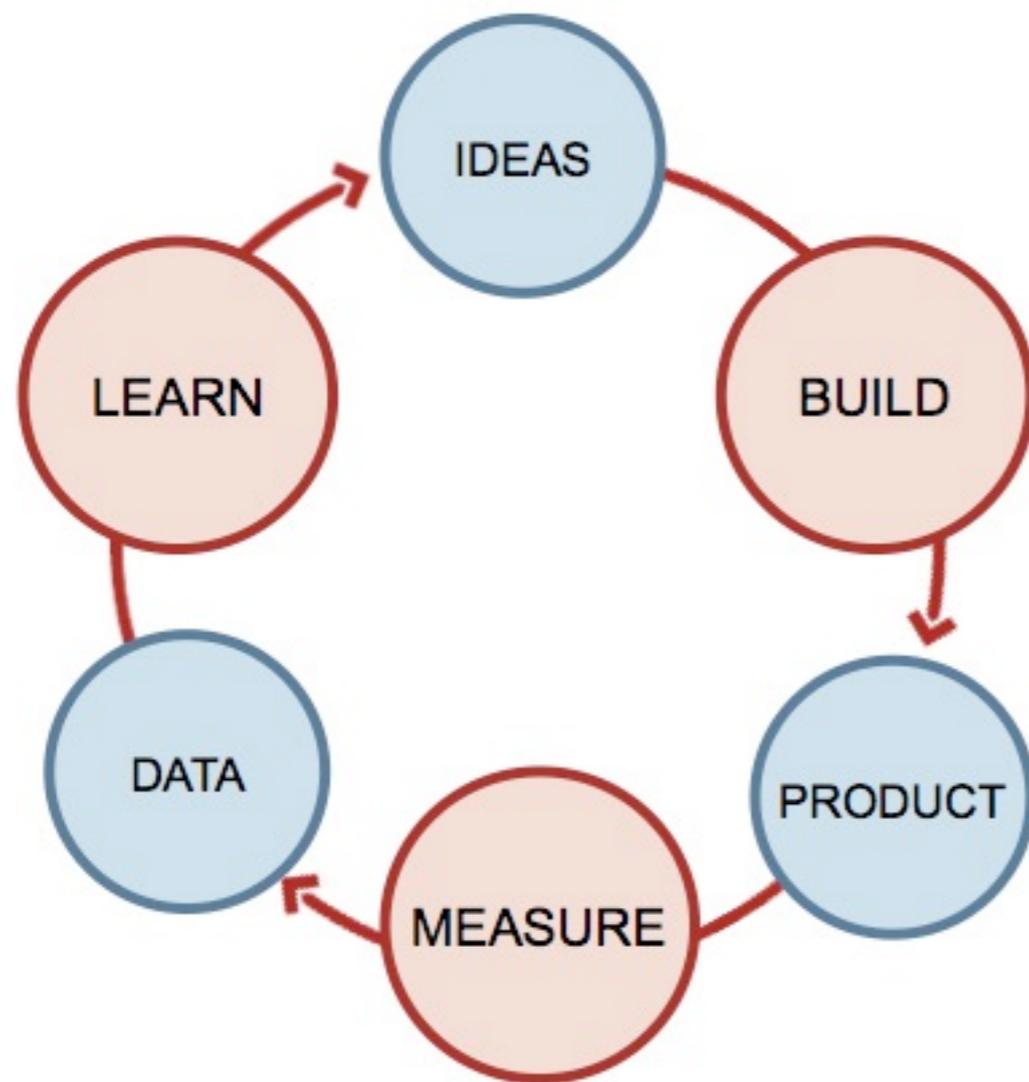
- 連續發佈：從兩週的開發和釋出週期，縮短到每天，極致是每一個 code commit 就發佈一次
- iOS 需等待審核，但可以讓每次上架後，都有一個準備好
- (well-tested) 的版本可以馬上送審。
- Fail often fail early 才是降低發佈風險最有效率的方法
- 要能 repeatable 和 reliable 的發佈，是需要技術配套的
 - 高度自動化測試、自動化的發佈程序、監控和 alert 系統

從幾個月才能得到
Feedback，縮短到固定兩週，
最後是每天。

為什麼縮短開發週期這
麼重要？

Lean Startup Cycle

你可以多快完成一個發想、實作、學習的循環?
你有一個點子，你多快可以驗證客戶買不買單?



錢燒光前可以完成越多循環，
成功的機會越大。

Speed of iteration beats
Quality of iteration

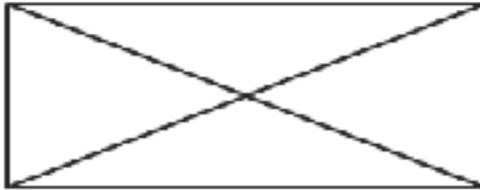
“
在十倍速的時代
不是大的打敗小的
不是強的打敗弱的
是快的打敗慢的
”

柯文哲



UX/UI 設計

- Wireframe 草圖
 - 線上工具 <https://moqups.com/>
- Mockup (plus 視覺設計)
 - 線上工具 <http://www.invisionapp.com/>
- Prototyping (plus 互動設計)
 - 刻 HTML (例如用 bootstrap)



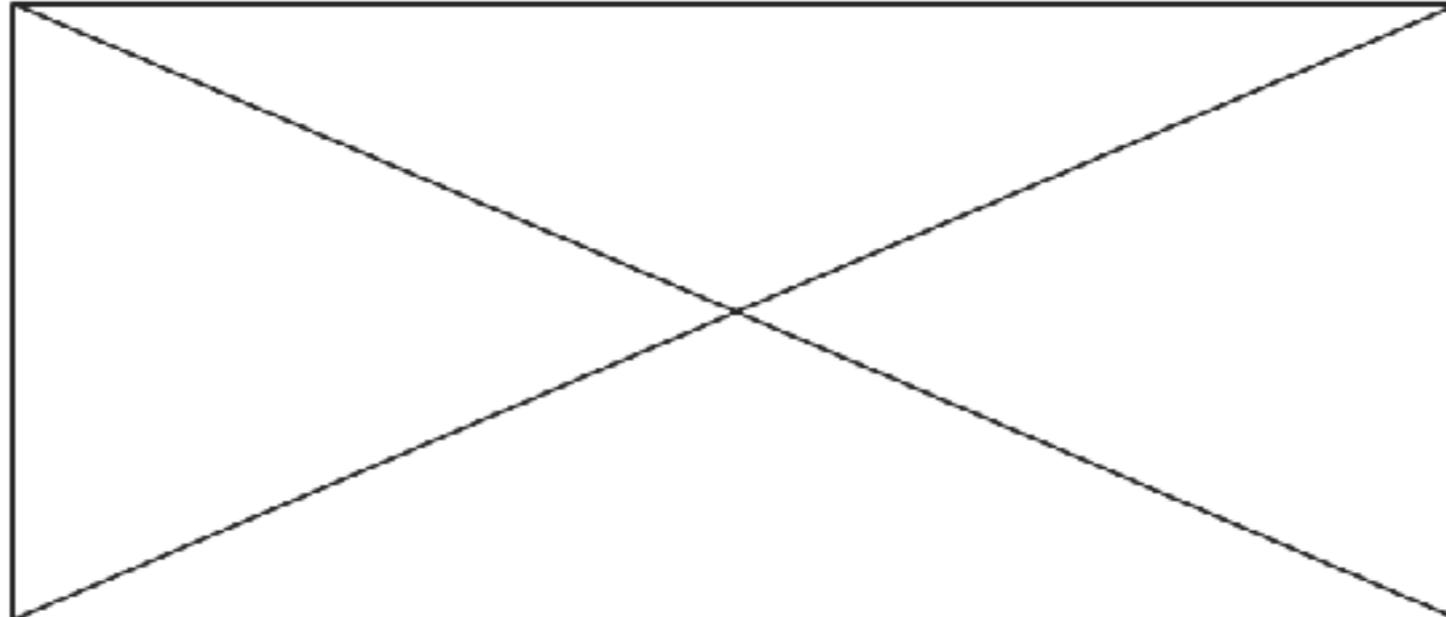
[View Cart](#) [Order History](#)

Products, Parts, Keywords Search

[Home](#) | [About Us](#) | [Products](#) | [Blog](#) | [News](#) | [Support](#) | [Contact](#)

800-888-8888

Rainfall Sensors
Wind Anemometer
Weather Stations
Controllers
Indicators/Displays
Temp/Humidity Sensors
Barometric Pressure
Solar Radiation
Replacement Parts



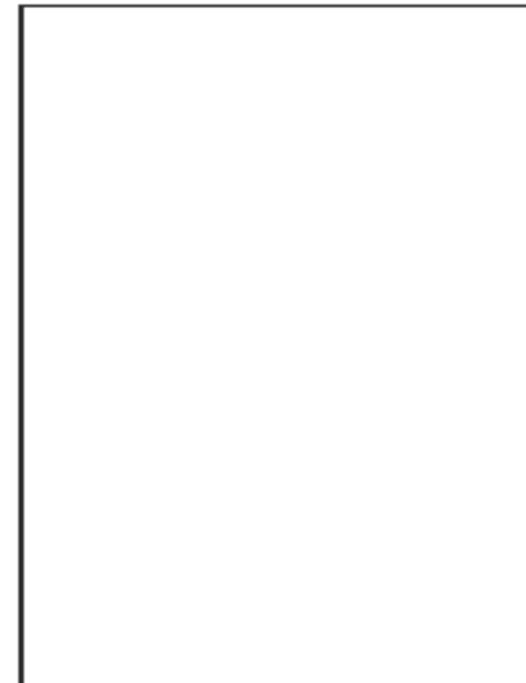
Featured Products

**Product Title**
SKU
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

**Product Title**
SKU
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

**Product Title**
SKU
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

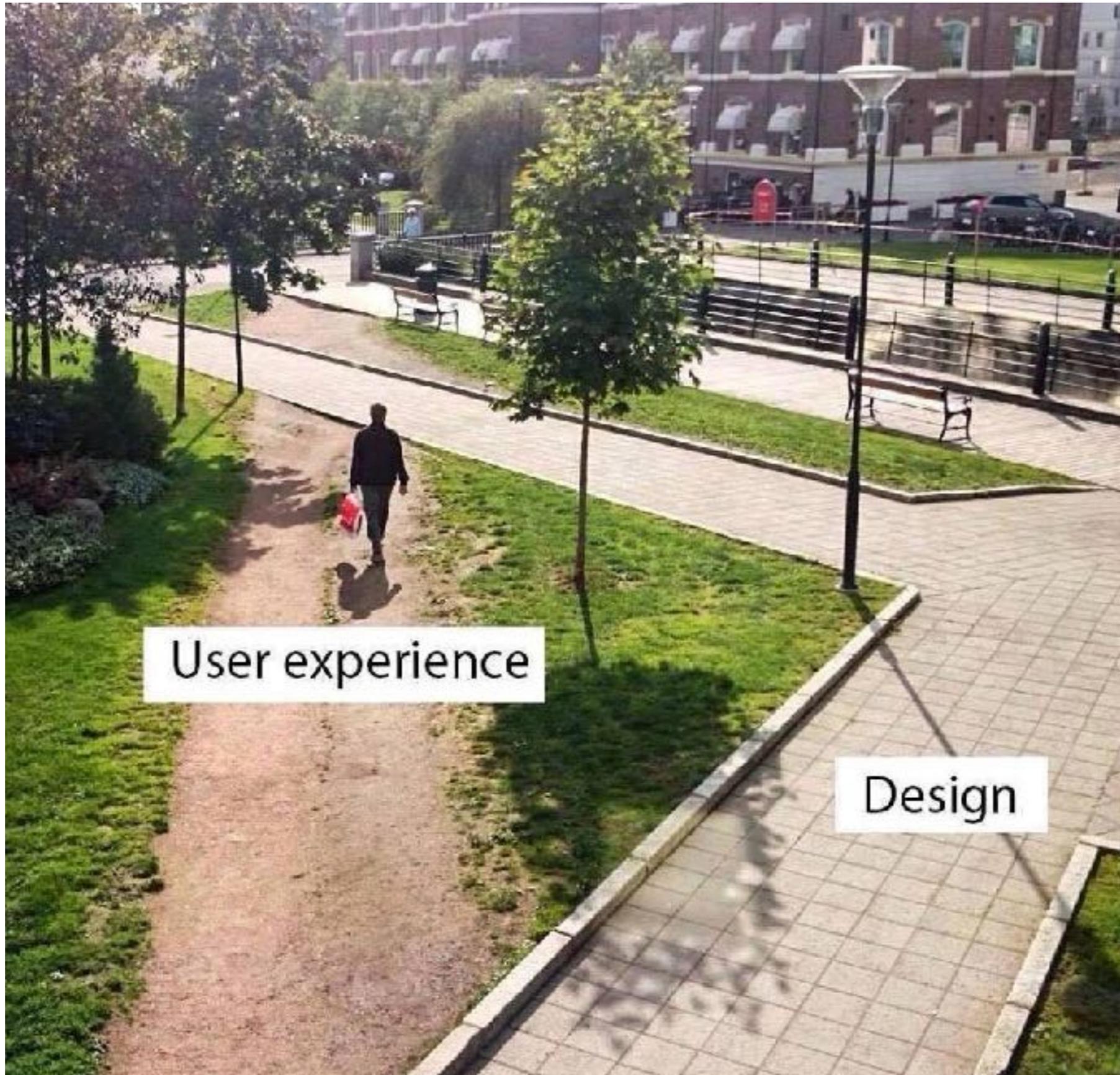
**Product Title**
SKU
Lorem ipsum dolor sit amet, consectetur adipiscing elit.





<https://dribbble.com/shots/1107830-From-Wireframe-to-Interface>

使用者經驗 vs. 漂亮的視覺設計

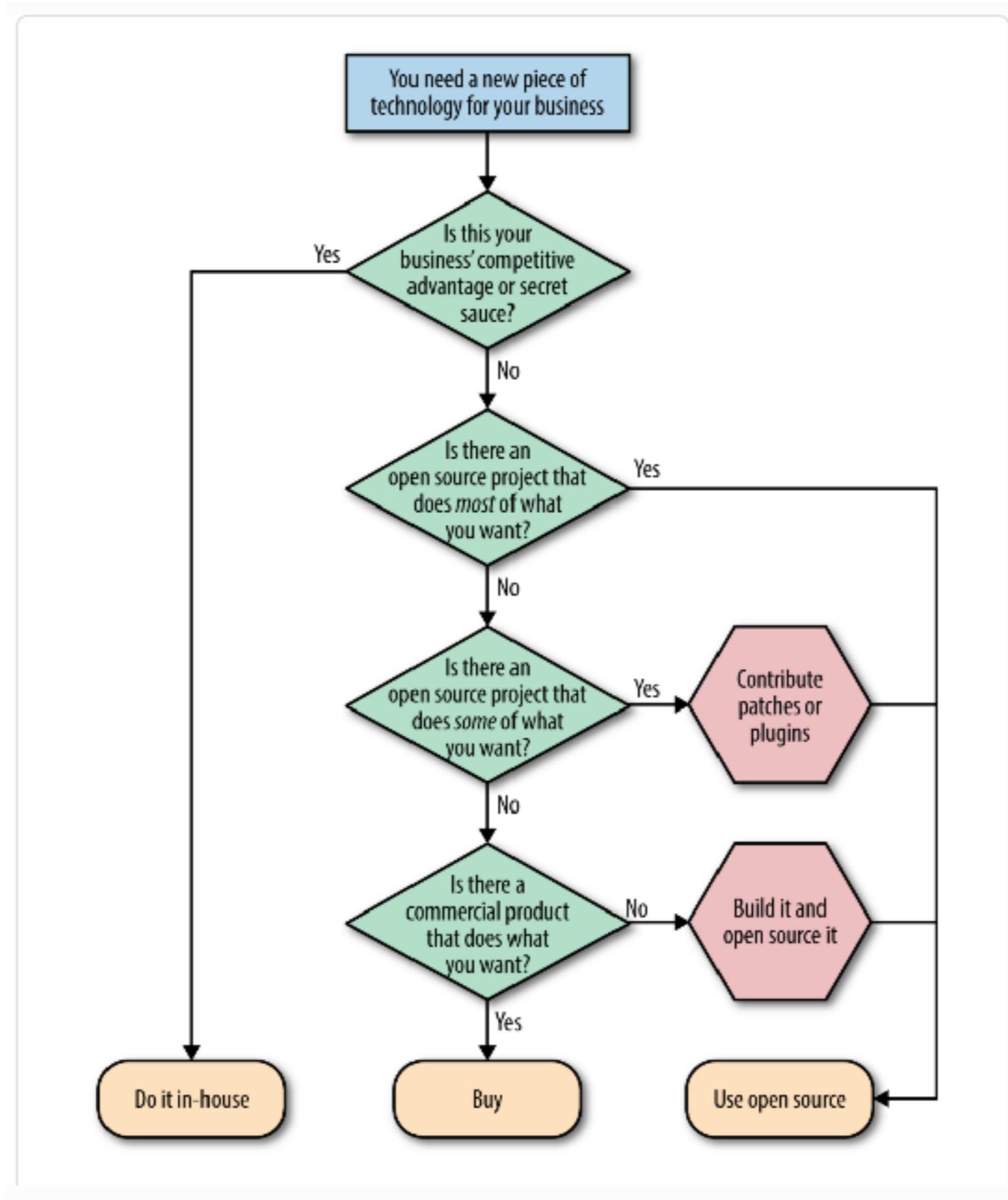


開發 Development

- 技術選擇
- 寫自動化測試，但要注意讓測試碼發揮效用
- 用版本控制系統 (強烈推薦用 Git) 貫穿開發流程
 - Pull request and Code Review
 - Continuous Integration (CI)

技術用現成的還是自己開發？

- in-house
- commercial
- open source



出自 Hello Startup, O'Reilly 一書

Don't reinvent the
wheel

這些東西不要自己蓋

- Operating systems
- Programming languages
- CS 101: basic data structures (map, list, set), sorting algorithms
- Web technologies: HTTP servers, server-side and client-side frameworks
- Data systems: databases, NoSQL stores, caches, message queues
- Software delivery: version control, build systems, deployment automation
- Libraries for common data formats: XML, HTML, CSV, JSON, URLs
- Utility libraries: date/time manipulation, string manipulation, logging
- Secure Libraries: cryptography, password storage, credit card storage
- Operations/Office software: SaaS (google, gmail, slack, quip, dropbox...etc)

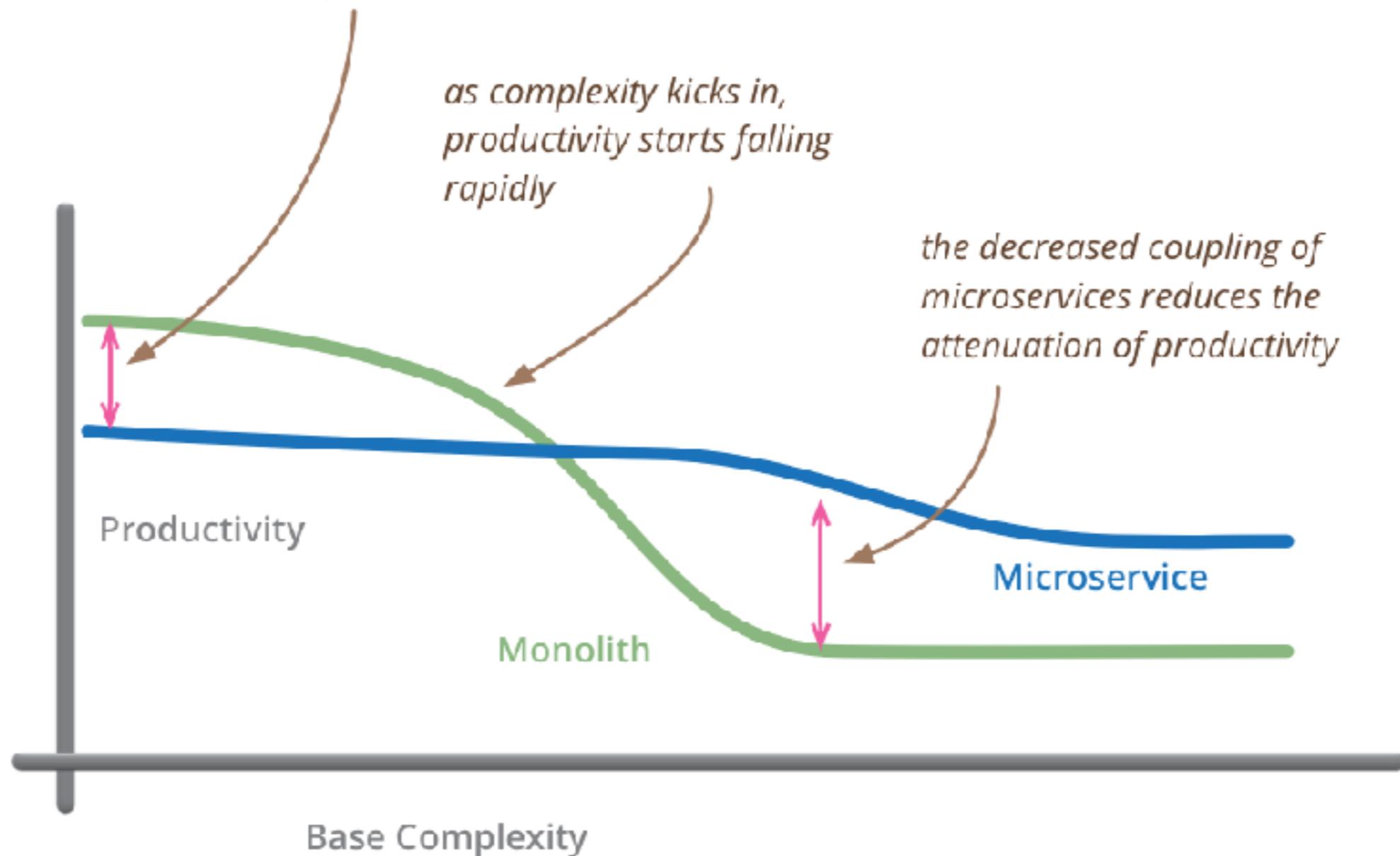
善用現成的 Library 和 Framework

API就是皮卡丘 提供 鋼鐵尾巴 跟 雷電 兩種技能給你呼叫 基本上 你不用研究皮卡丘為什麼會發電 也不用研究為什麼尾巴會變鋼鐵 反正你只要說: 上吧皮卡丘 使用雷電!!

選擇 Boring 的技術

- 選擇 boring 穩定的技術
- 用關聯式資料庫 MySQL 或 PostgreSQL，不要用 MongoDB
- 寫 iOS, Android 就乖乖用 native，不要用 Web 技術如 React Native
- 寫 Web 就用整套 Web framework 例如 Rails、Django
 - 不要搞什麼 microservice 或 macro framework
 - 用 Server-Side Rendering 技術，只有前台可能有需要搞 SPA，自己後台不需要
 - Isomorphic javascript 開發還不成熟，不要考慮
- 不要 overengineering：用 monolith 架構
 - 不要 SOA。SOA 是配合組織人員(單team10人)，不是技術問題。為下週需要的架構思考，不要為明年思考。
 - 康威定律：<http://www.infoq.com/cn/articles/every-architect-should-study-conway-law>

for less-complex systems, the extra baggage required to manage microservices reduces productivity



but remember the skill of the team will outweigh any monolith/microservice choice

Use Cloud Hosting

- 已經 2015 年了，沒有 startup 網路公司在買伺服器硬體的
- PaaS: Heroku
- IaaS
 - Amazon Web Services
 - Google
 - Microsoft Azure
 - Linode or Digital Ocean

寫自動化測試

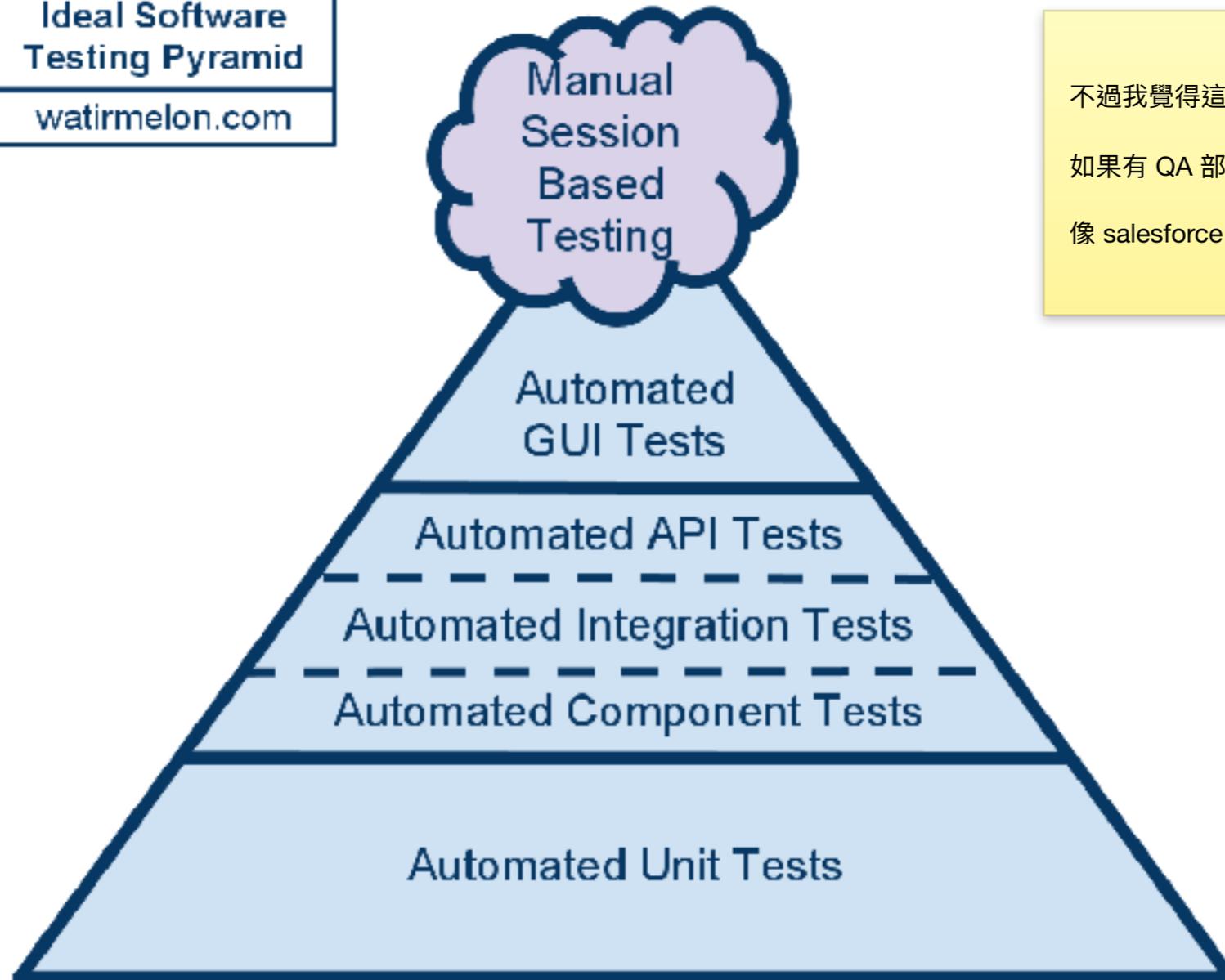
- Why?
 - 當程式邏輯比較困難時: 寫測試節省開發時間
 - 當你的程式不是用過即丟時: 回歸測試和重構，是 continuous delivery 很重要的信心來源
 - 透過 TDD 思維可以設計出可測試更好用的 API 介面

誰寫？如何極大效用

- 不是 100% coverage 就是好的，太浪費時間
- 開發人員自己寫自己的 Unit Test 單元測試
- GUI 驗收測試只寫 Happy Path，另外分配任務由不同開發者撰寫

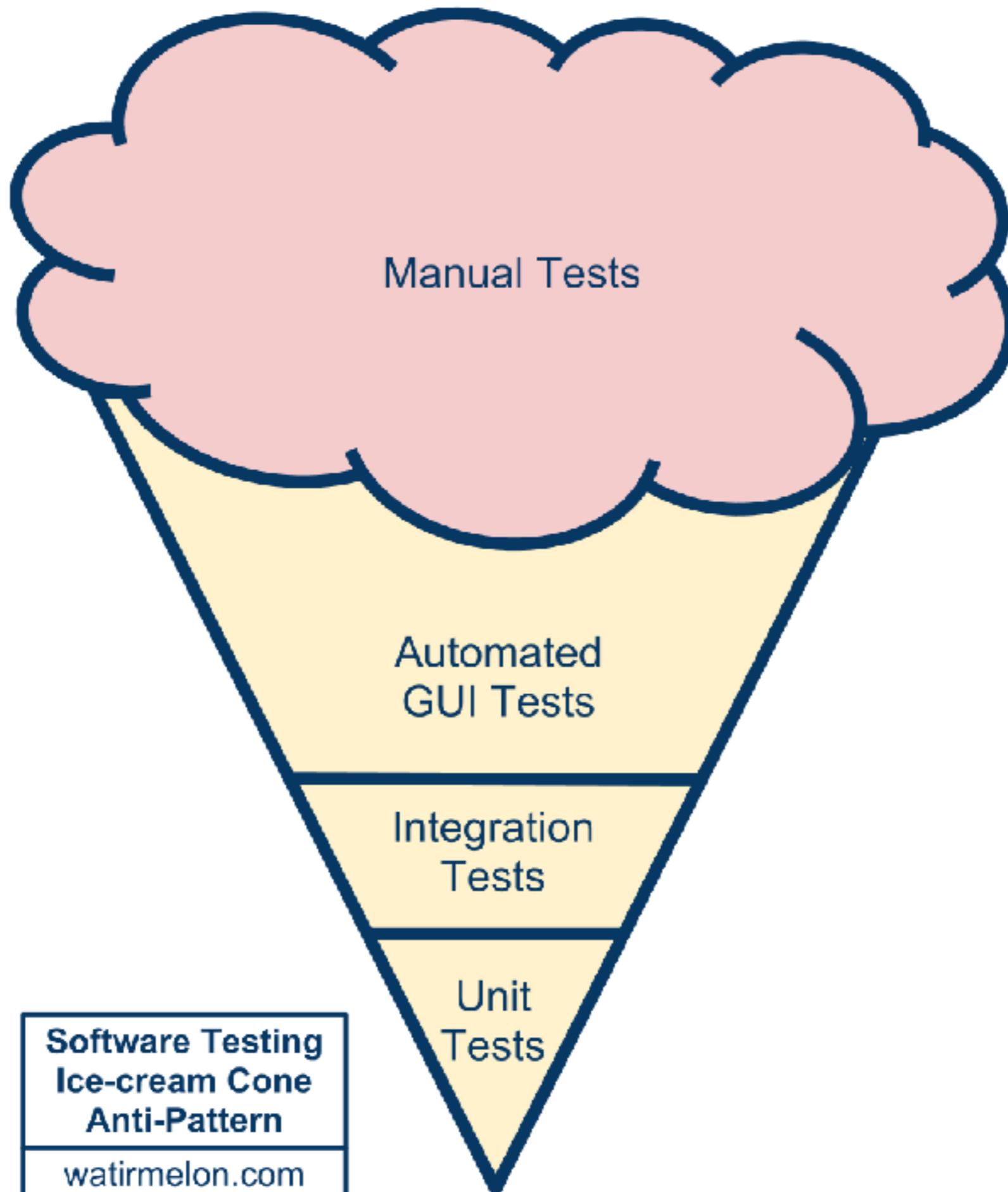
Test Pyramid

Ideal Software
Testing Pyramid
watirmelon.com



不過我覺得這是針對 developer 講的
如果有 QA 部門，那還是可以做 XD
像 salesforce ?

<http://watirmelon.com/2012/01/31/introducing-the-software-testing-ice-cream-cone/>
<http://martinfowler.com/bliki/TestPyramid.html>



Manual Tests

Automated
GUI Tests

Integration
Tests

Unit
Tests

**Software Testing
Ice-cream Cone
Anti-Pattern**

watirmelon.com

Why?

- 越 low-level 的測試，測試跑起來快，出錯時也容易 debug。
- 越 high-level 的測試，需要跑越久，而且出錯時也很難 debug。
 - 測試較脆弱，常 view 和流程一改，測試就要大改
 - 因此不建議 developer 投資太多，除非投資專門的 QA 人員

DHH Way

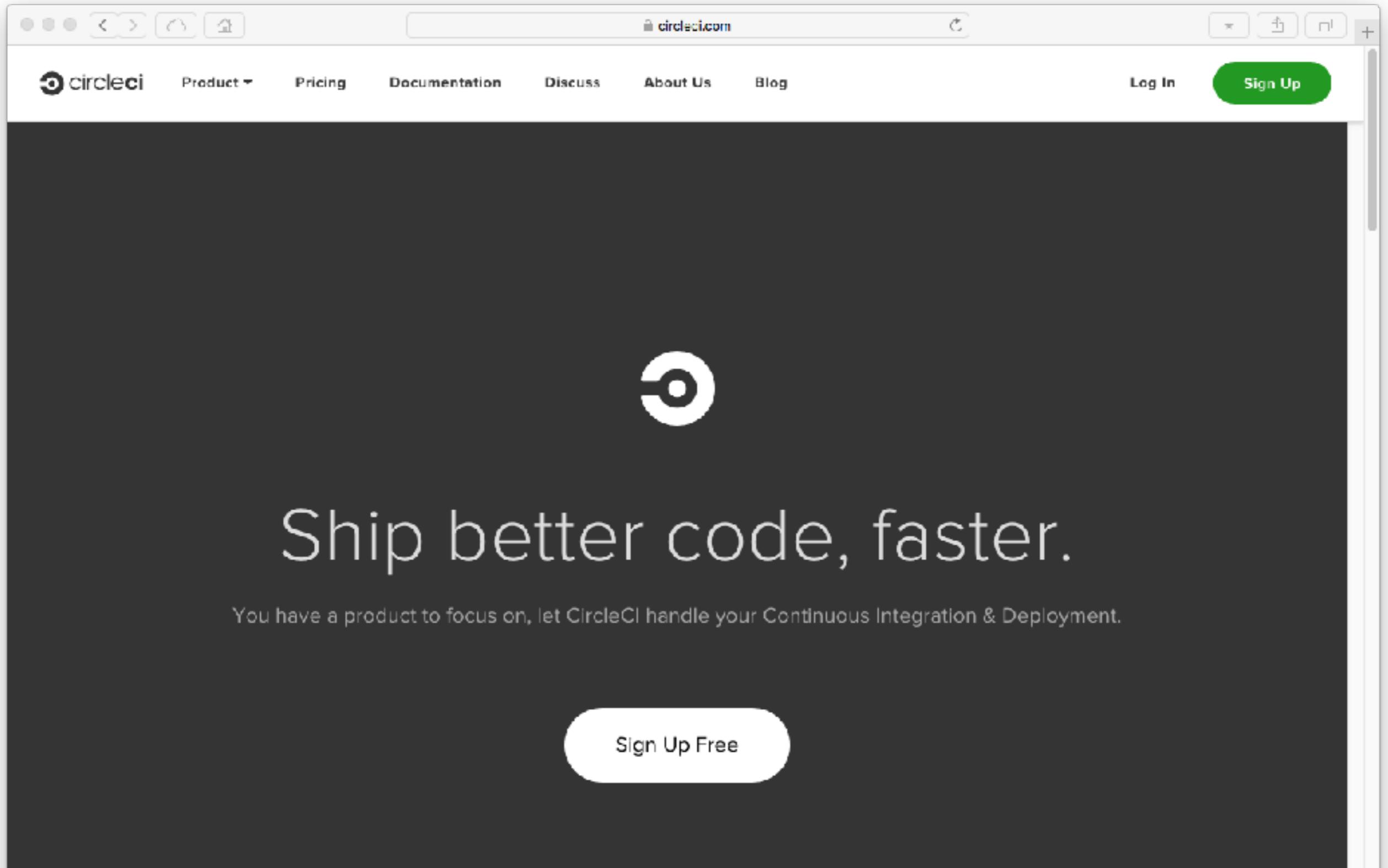


- 100% test coverage 不是目標
- Code-to-test 比例超過 1:2 要注意，1:3 表示你寫太多測試了
- 不超過 1/3 的時間寫測試，超過一半時間你一定搞錯了
- 不要測試標準的 Active Record associations, validations, or scopes.
- 使用整合測試 (但不要測試 Unit Test 已經測到的東西)
- 不要用 Cucumber，別幻想了
- 不要強迫自己用 TDD (DHH 大概只有 20% 用 TDD)
- Model 測試不需要隔離 DB、建議用 Fixtures 執行速度比較快
- Controller 測試用整合測試的方式
- Views 測試用 system/browser testing

用 Git 貫穿開發流程

- 都 2015 年了，還有人沒有用版本控制系統嗎？
- 這個流程包括：
 - 開 Feature branch
 - 每個 commit 跑 CI 連續整合
 - 做 Code Review
 - 開 Release branch 給 beta user 人工測試 (staging server)
 - 合併進主幹，進行 Production 部署

CI 目前也有很多 3-rd party 工具



CircleCI https://circleci.com hower

By Repo Mine All Bulbs

ALPHACamp/dojo

Build	Revision	Branch	Author	Log	Started	Length	Status
feature/apply_v3	✓						
master	✓						
staging	✓						
ALPHACamp/dojo #415	6cea461	feature...bs_options_v2	John Chang	Create condition for admin/staff to see out of data jobs	18 hours ago	01:07	Success
ALPHACamp/dojo #414	c7f3213	staging	John Chang	Fix conflicts	73 hours ago	02:59	Success
ALPHACamp/dojo #413	14d088a	feature...bs_options_v2	John Chang	fix some wording in HR.yml file	24 hours ago	01:05	Success
ALPHACamp/dojo #412	f0e4bae	feature...bs_options_v2	John Chang	Add more options for job creation	1 day ago	01:05	Success
ALPHACamp/dojo #411	c7293ff	master	Wan-Tien Chang	Merge branch 'feature/set_deadline_for_opportunity_v2'	1 day ago	01:23	Success
ALPHACamp/dojo #410	c4836ae	feature...ppportunity_v2	John Chang	optimized coding style	1 day ago	01:01	Success
ALPHACamp/dojo #409	c4836ae	staging	John Chang	optimized coding style	1 day ago	04:34	Success
ALPHACamp/dojo #408	afc8d1d	feature/set_jobs_options	John Chang	Add more options for job creation	1 day ago	02:45	Success
ALPHACamp/dojo #407	3ee74b0	feature/set_jobs_options	John Chang	Add more options for job creation	1 day ago	02:53	Success
ALPHACamp/dojo #406	7010b07	master	Wan-Tien Chang	Merge pull request #10 from ALPHACamp/hotfix/fix_dev_fake	3 days ago	02:00	Success
ALPHACamp/dojo #405	d62a0e0	hotfix/fix_dev_fake	Yang Li-Yi	Change teacher_id to teacher	3 days ago	01:05	Success
ALPHACamp/dojo #404	4962ee0	hotfix/fix_dev_fake	Yang Li-Yi	Fix devfake bug	3 days ago	01:03	Success
ALPHACamp/dojo #403	75c153c	feature...ppportunity_v2	John Chang	Solve migration pending issue	4 days ago	01:10	Fixed
ALPHACamp/dojo #402	99d4c2b	feature...ppportunity_v2	John Chang	Optimized coding style	4 days ago	01:15	Failed
ALPHACamp/dojo #401	1c884b3	feature...ppportunity_v2	John Chang	Optimized coding style	4 days ago	01:02	Success
ALPHACamp/dojo #400	f2eb07f	feature...ppportunity_v2	John Chang	Create deadline selection for jobs	4 days ago	01:04	Success
ALPHACamp/dojo #399	0055f94	master	Wan-Tien Chang	Fix nil error	6 days ago	01:07	Success
ALPHACamp/dojo #398	93d129d	master	Wan-Tien Chang	Fix nil error	6 days ago	01:07	Success
ALPHACamp/dojo #397	8ad4aee	master	Wan-Tien Chang	Fix date_select input	7 days ago	01:05	Success
ALPHACamp/dojo #396	f125a45	feature/apply_v3	Wan-Tien Chang	Add application and interview result emails, public email	7 days ago	01:05	Fixed

travis-ci.org

Travis CI Home Blog Status Help We're Hiring! Travis CI for Private Repositories Sign in with GitHub

Search all repositories

Recent

- Kotti/Kotti 501
 - 3 sec
- msabramo/Kotti 15
 - 6 sec
- docker/machine 607
 - 29 sec
- BCS-lo/letting 2724
 - 49 sec
- strongloop/fsevents 8
 - 12 sec
- supercollider/supercollider 265
 - 46 sec
- hyperoslo/HYPForms 577
 - 10 sec
- mozilla/bedrock 1270
 - 52 sec

Kotti/Kotti build passing Settings

Current Build History Pull Requests Branch Summary

master - README.rst: Add instructions for Docker

#501 started

- running for 3 sec
- Commit ab54e90
- #386: README.rst: Add instru...

Marc Abramowitz authored and committed

Build Matrix

Job	Duration	Finished	Python	ENV	OS
501.1	-	-	2.6	KOTTI_TEST_DB_STRING=postgresql://postgres@localhost:5432/kotti_testing	linux
501.2	3 sec	-	2.6	KOTTI_TEST_DB_STRING=mysql+oursql://root@localhost:3306/kotti_testing	linux
501.3	-	-	2.6	KOTTI_TEST_DB_STRING=sqlite://	linux
501.4	-	-	2.7	KOTTI_TEST_DB_STRING=postgresql://postgres@localhost:5432/kotti_testing	linux
501.5	-	-	2.7	KOTTI_TEST_DB_STRING=mysql+oursql://root@localhost:3306/kotti_testing	linux
501.6	-	-	2.7	KOTTI_TEST_DB_STRING=sqlite://	linux

Code review 用 Github 的 Pull Request

The screenshot shows a GitHub Pull Request for the Rails project. The browser address bar displays the commit hash: `https://github.com/rails/rails/commit/c834a751d2acbd55b580cbba2e96dd29c5d9a452`. The page title is "Commit c834a751d2acbd55b580cbba2e96dd29c5d9a452 to rails/rails - GitHub".

The repository is "rails / rails" with 7,862 stars and 1,352 forks. The commit message is "define_attr_method correctly defines methods with invalid identifiers" by spastorino, made about 16 hours ago. The commit details show the commit hash `c834a751d2acbd55b580cbba2e96dd29c5d9a452`, the tree hash `423e181f132e2977ac12`, and the parent hash `fda45f4fc493f5596375`.

Two files are shown as changed: `activemodel/lib/active_model/attribute_methods.rb` (5 changes) and `activemodel/test/cases/attribute_methods_test.rb` (6 changes). The selected file, `activemodel/lib/active_model/attribute_methods.rb`, shows a diff with the following code changes:

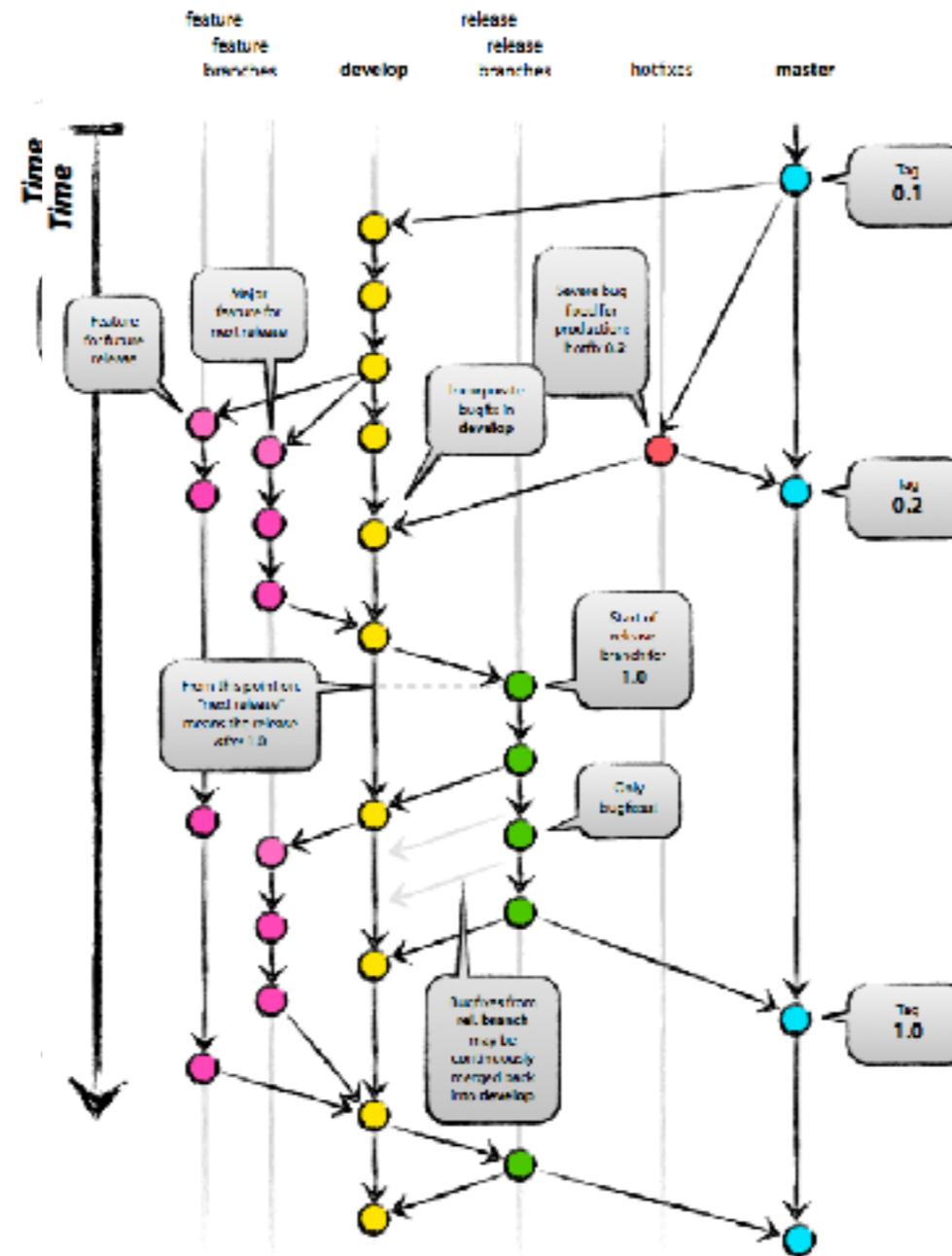
```
... -108,9 +108,8 @@ module ActiveSupport
108 108     else
109 109     # use eval instead of a block to work around a memory leak in dev
```

There are two comments on the diff:

- From dmitry (about 3 hours ago): "Forgot to cleanup comment?"
- From spastorino (about 3 hours ago): "dmitry, hey thanks for pointing me to this. I saw this yeah I have to try a few more things and do another commit."

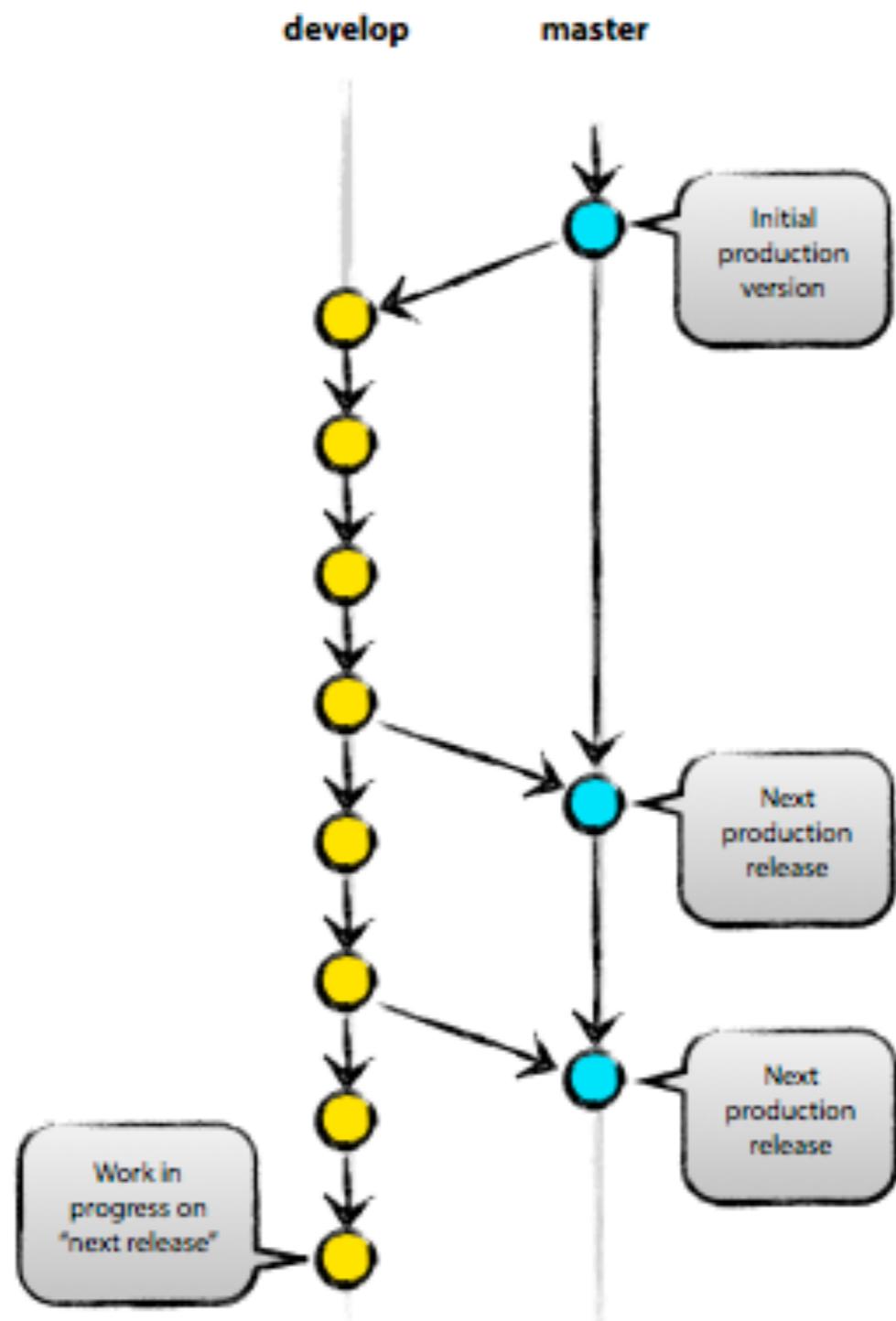
Git flow

<http://nvie.com/posts/a-successful-git-branching-model/>



兩個主要分支

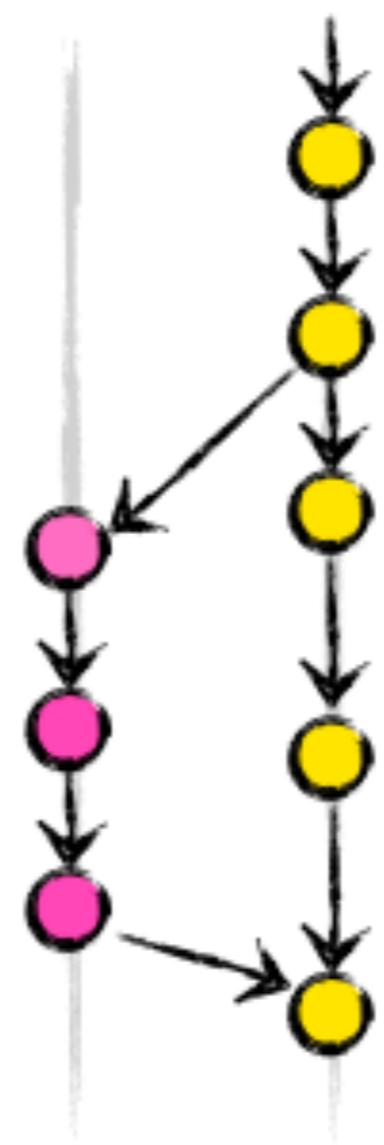
- master: 穩定的 production 版
- develop: 開發版本，從 master 分支出來

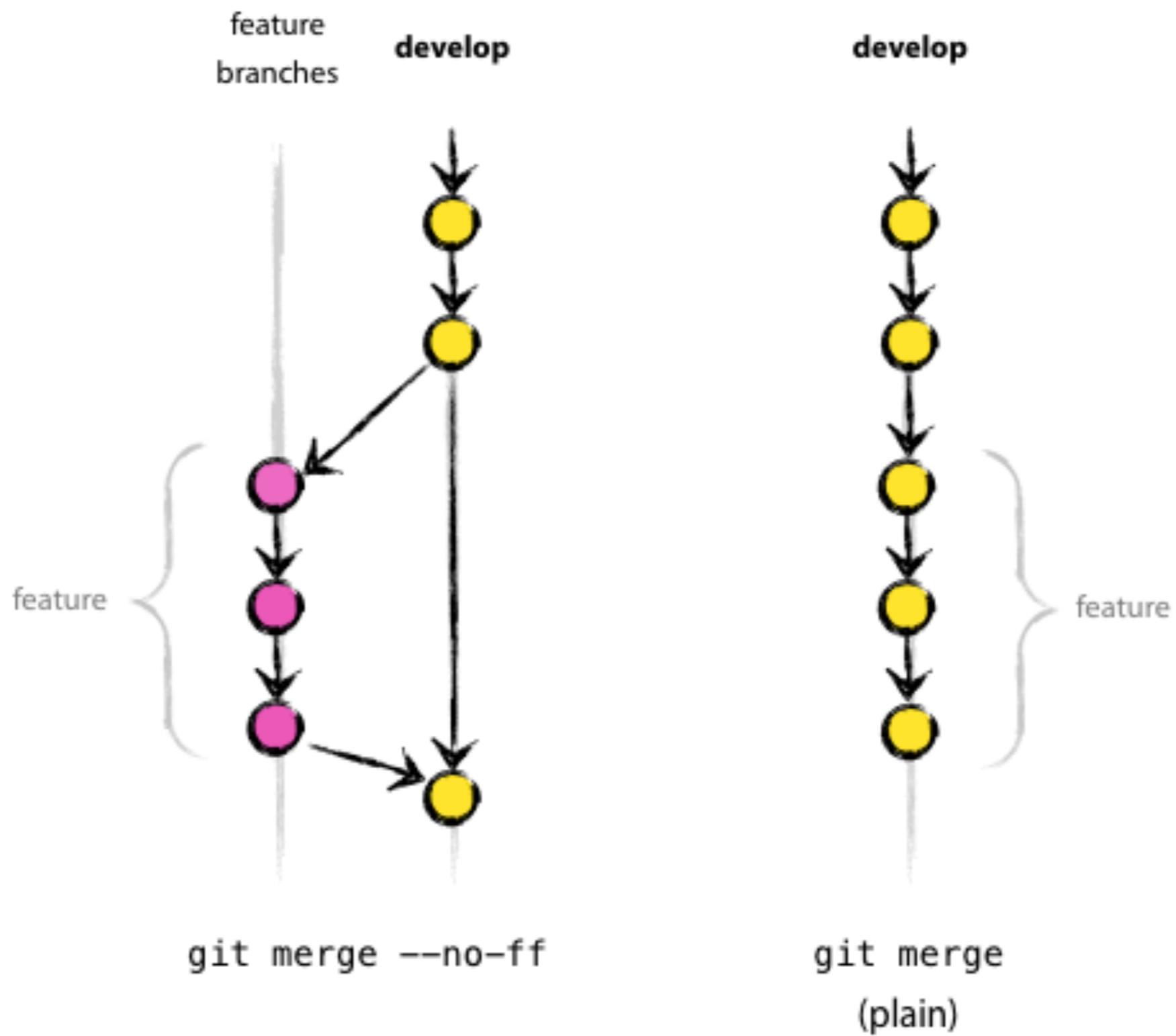


三種支援性分支(I)

- Feature branches
 - 開發新功能或修 bugs
 - 從 develop 分支出來
 - 完成後 merge 回 develop
 - 如果開發時間較長，則需定期同步 develop 主幹的程式(初學可用 merge，建議改用 rebase)，不然最後會合併不回去。

feature
branches **develop**



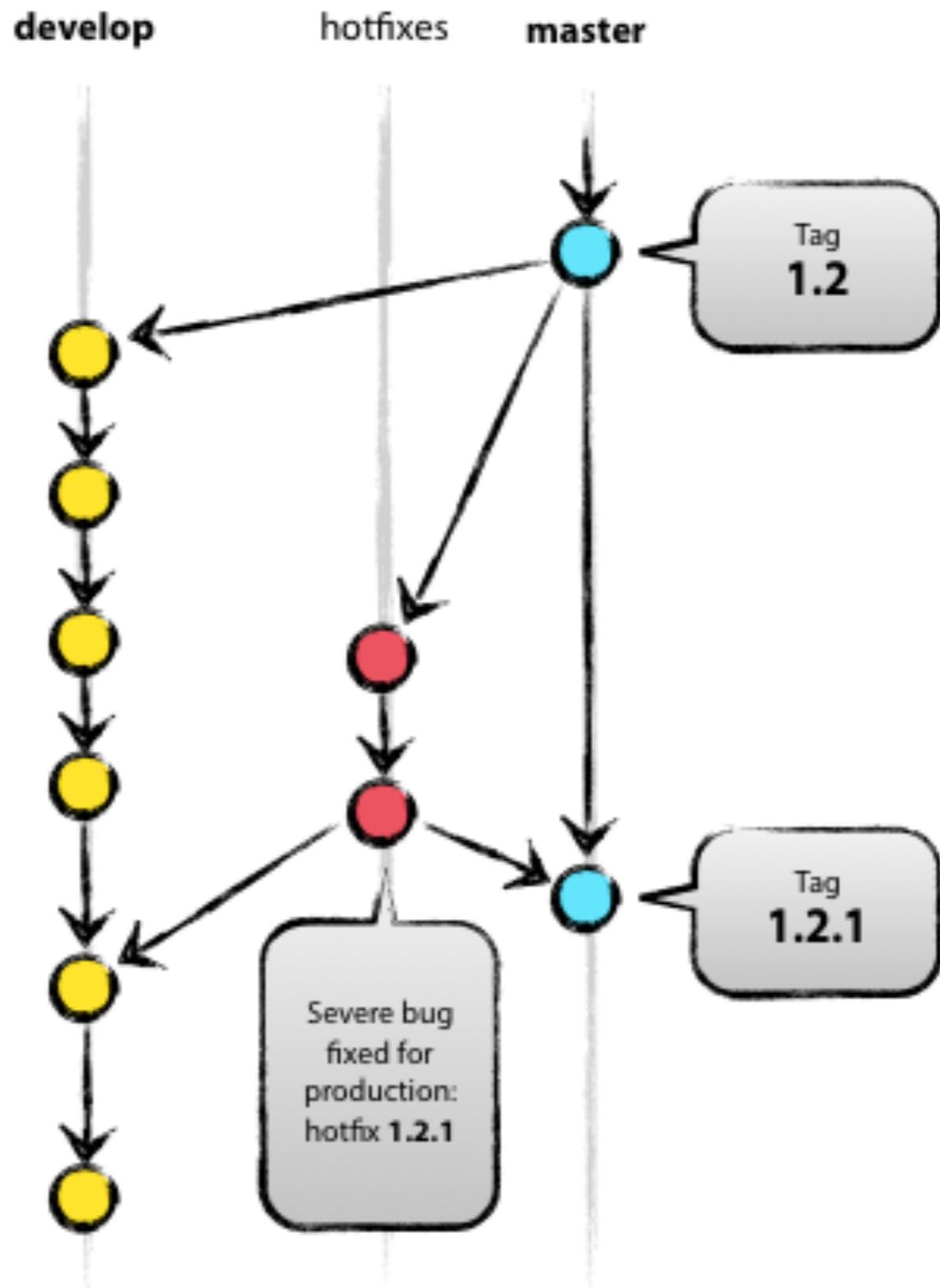


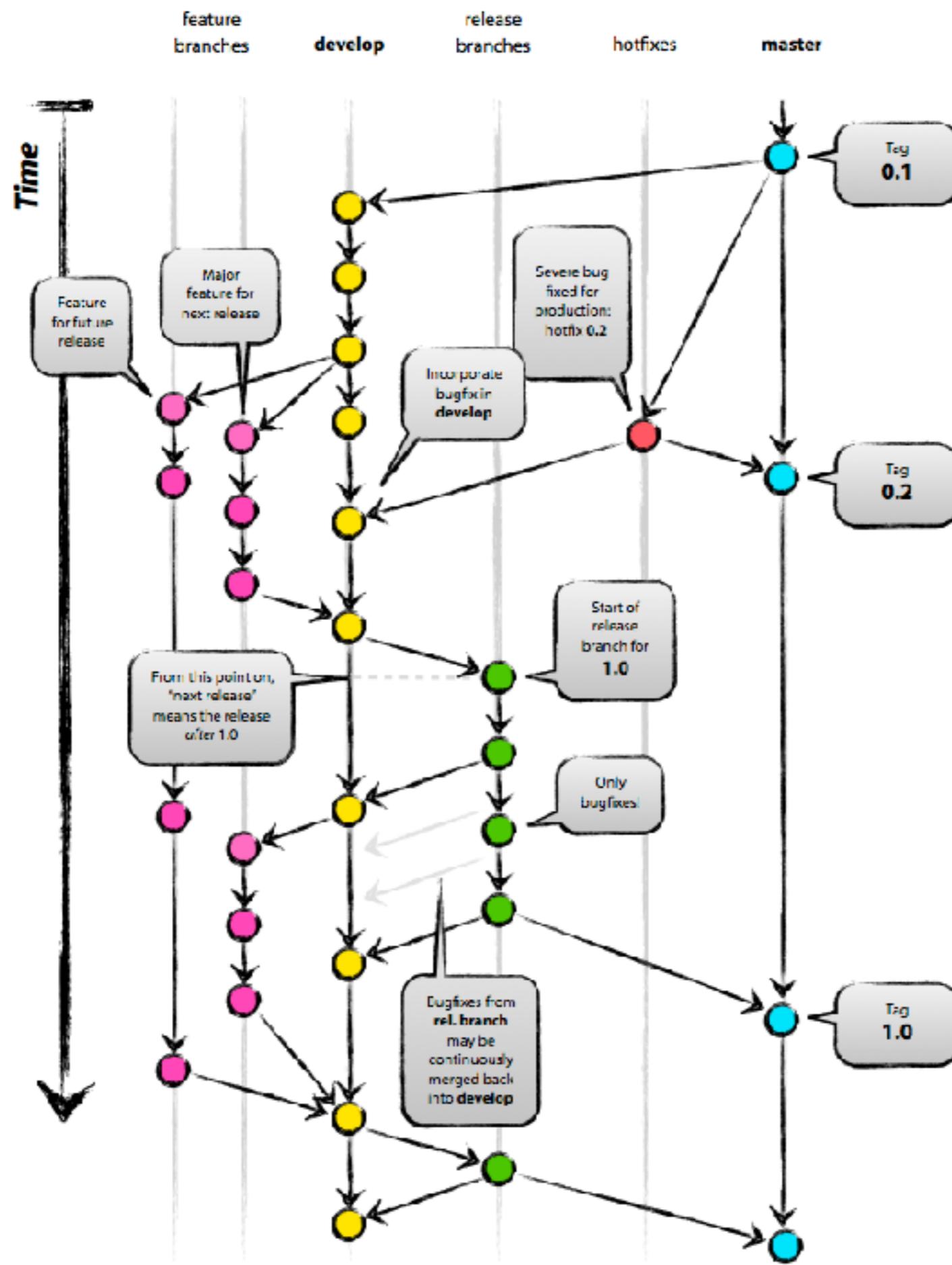
三種支援性分支(2)

- Release branches
 - 準備要 release 的版本，只修 bugs
 - 從 develop 分支出來
 - 完成後 merge 回 master 和 develop

三種支援性分支(3)

- Hotfix branches
 - 等不及 release 版本就必須馬上修 master 趕著上線
 - 會從 master 分支出來
 - 完成後 merge 回 master 和 develop





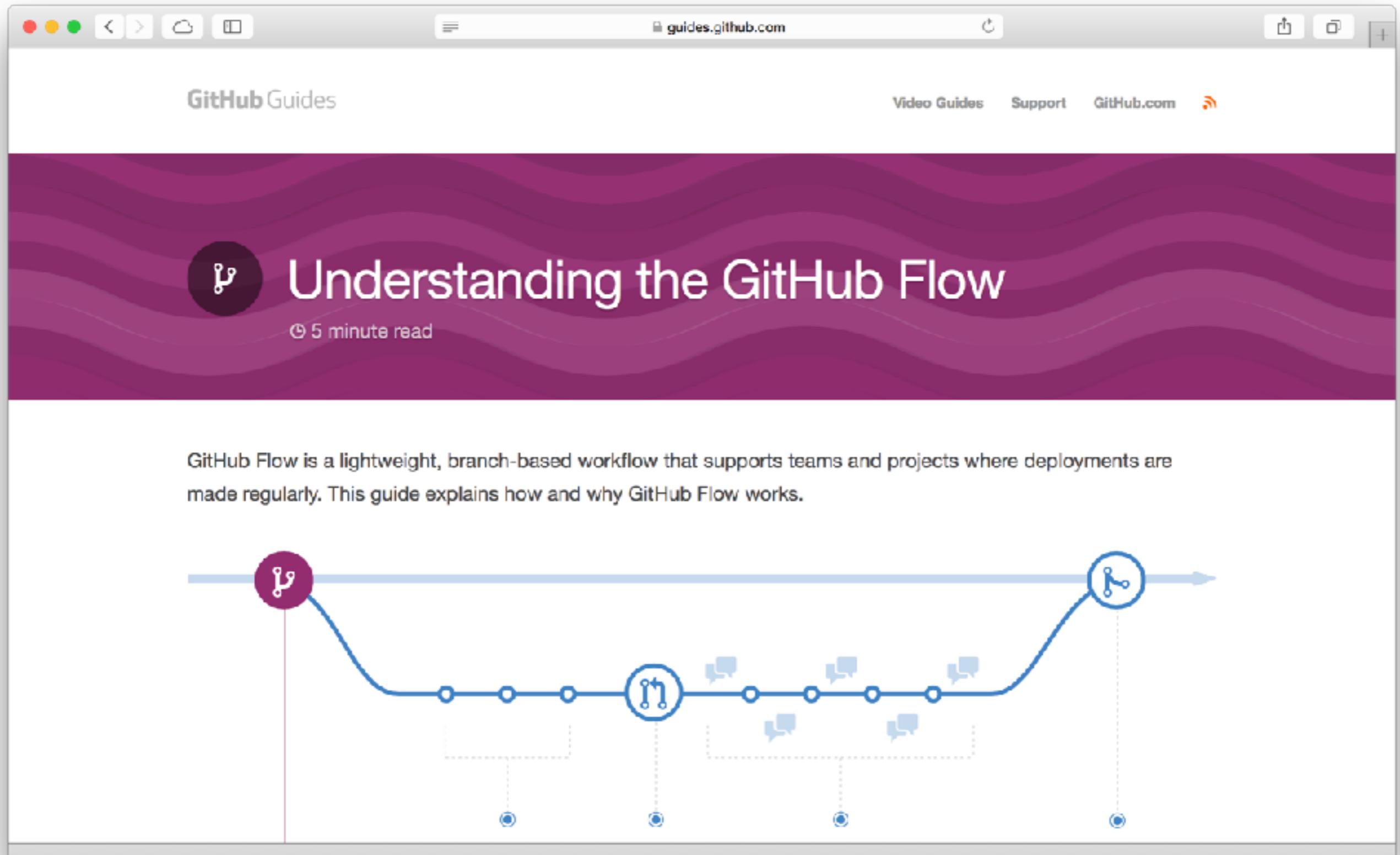
pros and cons

- 第一個出名的、有形式的 Git 分支工作流程
- 很清楚的追蹤 feature, release, hotfix 等 branches，確保釋出和審核流程
- 有點複雜，不適合小專案或 Open Source 專案

Github flow

- master 是 stable/production 可佈署的版本
- 任何開發從 master branch 分支出 feature branch
- 送 pull request 開始進行討論、code review 和測試
- 最後合併回 master 代表可以佈署了

<https://guides.github.com/introduction/flow/index.html>



The image shows a browser window displaying the GitHub Guides page for "Understanding the GitHub Flow". The browser's address bar shows the URL `guides.github.com`. The page header includes the "GitHub Guides" logo and navigation links for "Video Guides", "Support", and "GitHub.com". The main content area features a purple wavy background with a circular GitHub logo icon and the title "Understanding the GitHub Flow" in white text. Below the title, it indicates a "5 minute read". The introductory text states: "GitHub Flow is a lightweight, branch-based workflow that supports teams and projects where deployments are made regularly. This guide explains how and why GitHub Flow works." Below the text is a diagram illustrating the workflow. It shows a horizontal blue line representing the main branch, with a GitHub logo icon at the start and a deployment icon at the end. A branch is created from the main branch, shown as a lower horizontal line with several small circles representing commits. A central circle with a GitHub logo icon is highlighted on this branch. Dashed lines connect these commits to a vertical line below, representing the local repository state. Small speech bubble icons are placed above the branch line, indicating pull requests or discussions.

開發與佈署流程

<http://ihower.tw/blog/archives/7798>

- 分支流程不只與開發相關，也與測試和軟體部署(釋出)流程相關
- 什麼時候，哪個分支跑 CI 自動測試?
- 什麼時候，哪個分支佈署到 Staging Server 進行人工測試?
 - 使用 Github flow 的話，需要讓每個 feature branch 都可以上 CI 和 staging server 環境
 - 或是採用有個專門跑整合測試的 branch，開個 staging branch 專門用來整合跑 CI，這個 branch 不會合併回 master

與專案管理的搭配

- Scrum (有 iteration 開發週期)
 - 比較適合搭配有 release branch 的 Git flow
- Kanban (沒有 iteration 開發週期)
 - 比較適合 Github flow 流程

Git 流程小結論

<https://ihower.tw/git/>

- Github flow 或 Gitflow 二選一
- desktop/mobile software：用 Gitflow
- 想要嚴謹的流程：用 Gitflow
- 頻繁釋出的 Web app: 用 Github flow

Feature Toggle

- 功能完成，不代表業務面需要馬上上線
- 擔心 Feature branch 放太久臭掉?
- 建議先合併進主幹! 但是 UI 先藏起來即可!
- 甚至是設計權限，讓 admin 可以提早看到做 production 的線上測試

小結

- 專案管理: Scrum 或 Kanban
- 不重複發明輪子
- 用 Wireframe 做設計
- 寫自動化測試
- 用版本控制系統 Git 搭配 Github flow 或 git flow

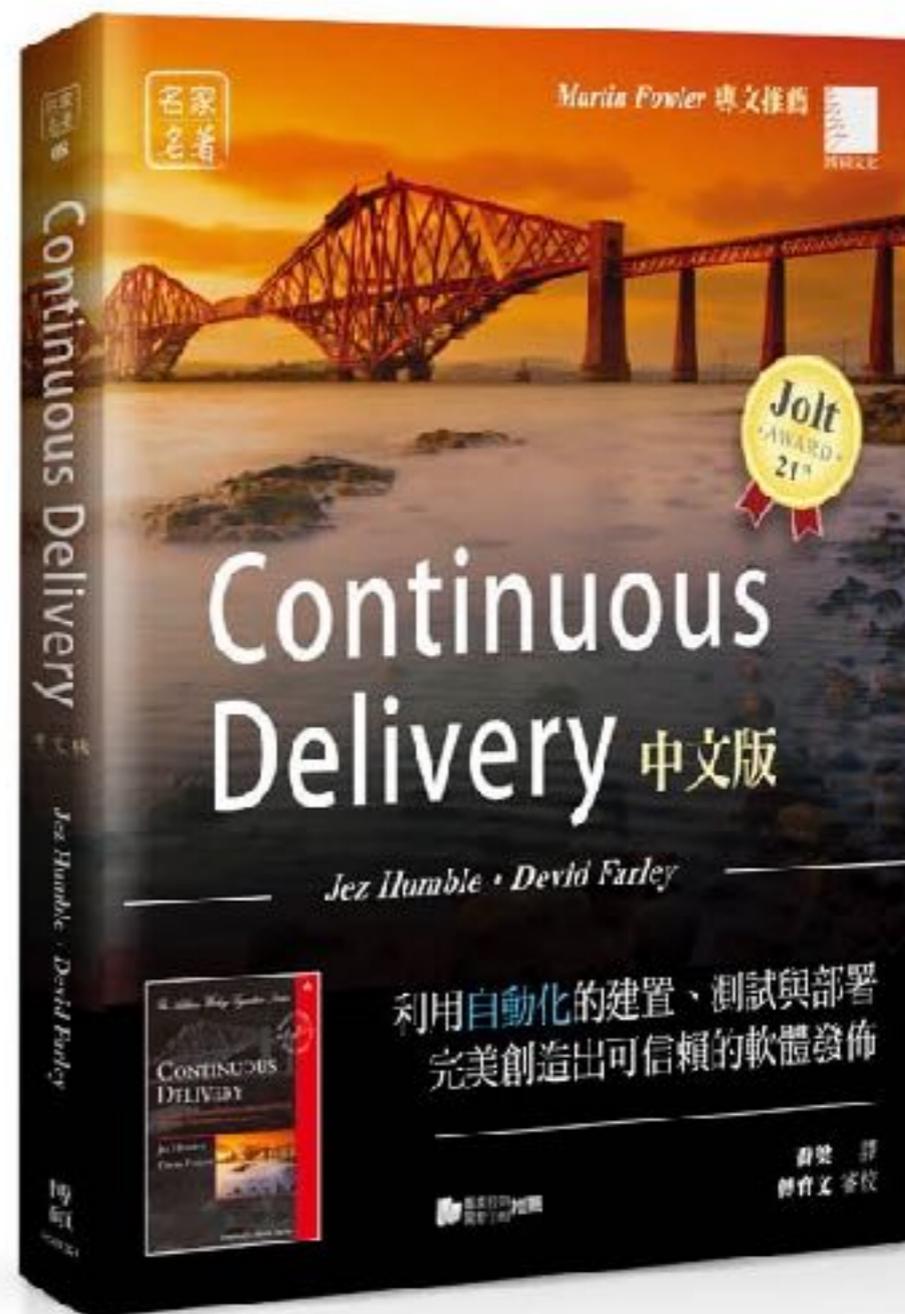
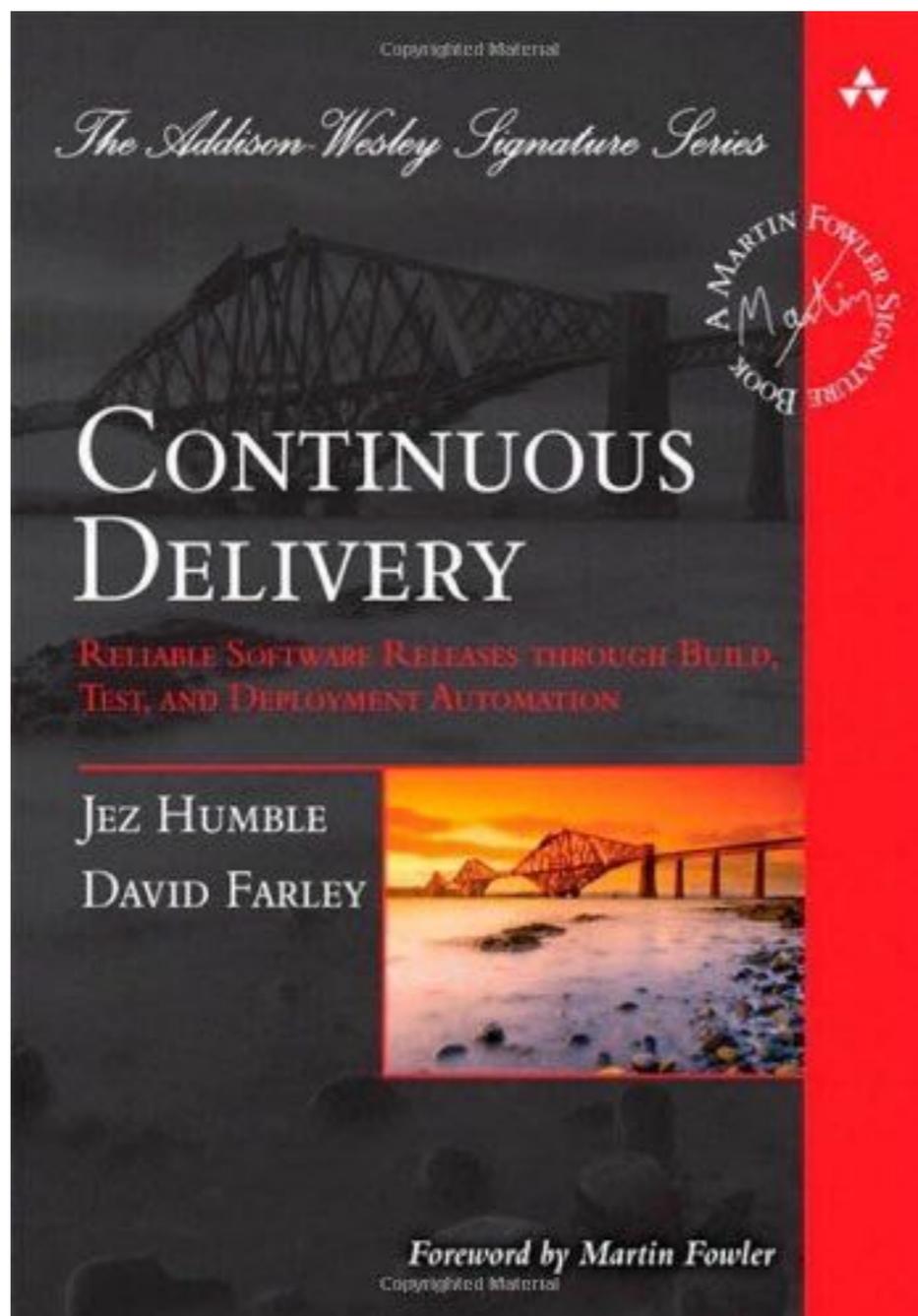
3. 部署上架

Shipping 才是軟體開始發揮價值 的時候



自動化部署程序

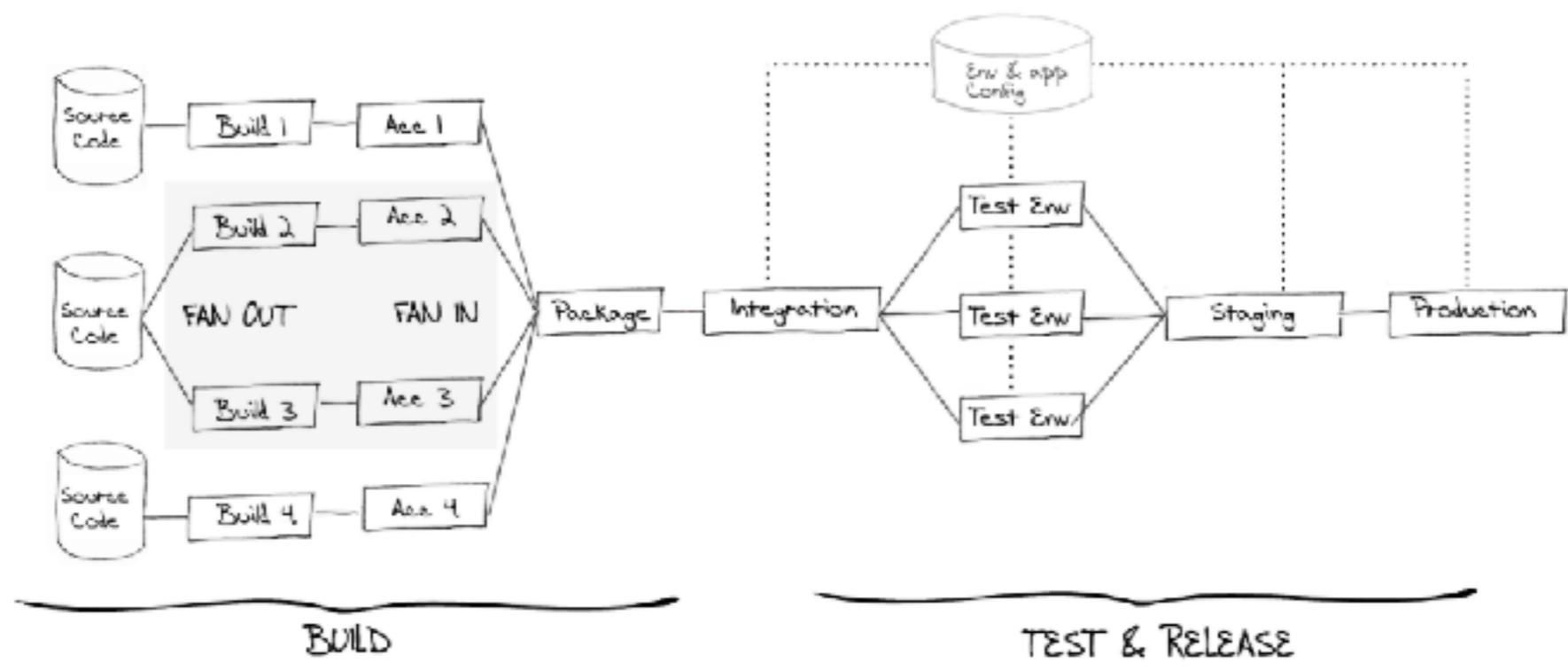
- 都 2015 年了，還有人用 FTP 上傳進行部署嗎？
- 與 Git 開發流程銜接，完成整套 Continuous Delivery 流程
 - 一鍵進行 Build、上 Staging 伺服器、上 Production 伺服器的部署動作
- 例如 Application Development 工具 <http://capistranorb.com/> (Ruby)



Go 15.2.0 now available!

Check out [What's new](#) in 15.2.0 or [Download](#) it.

Automate and streamline the build-test-release cycle for worry-free, continuous delivery of your product.



aws.amazon.com

Menu
amazon web services
AWS re:Invent Announcements
Products
More
English
My Account
Create an AWS Account

PRODUCTS & SERVICES

- AWS CodePipeline** >
- Product Details >
- Pricing >
- Getting Started >
- Product Integrations >
- FAQs >

RELATED LINKS

- Documentation
- Partner Solutions

AWS CodePipeline

AWS CodePipeline is a **continuous delivery** service for fast and reliable application updates. CodePipeline builds, tests, and deploys your code every time there is a code change, based on the release process models you define. This enables you to rapidly and reliably deliver features and updates. You can easily build out an end-to-end solution by using our pre-built plugins for popular third-party services like GitHub or integrating your own custom plugins into any stage of your release process. With AWS CodePipeline, you only pay for what you use. There are no upfront fees or long-term commitments.

Get Started With AWS for Free

[Create a Free Account](#)

Receive twelve months of access to the AWS Free Tier and enjoy AWS Basic Support features including, 24x7x365 customer service, support forums, and more.

[View AWS Free Tier Details >](#)

```

graph LR
    subgraph Pipeline [AWS CodePipeline]
        S[Source  
Developers commit changes] --> B[Build  
Changes are built]
        B --> ST[Staging  
Code is deployed and tested]
        ST --> P[Production  
Code is deployed to public servers]
    end
    P --> C[Customers]
    C --> IB[Ideas  
Requests Bugs]
    IB --> CH[Changes  
Updates Fixes]
    CH --> D[Developers]
    D --> S
  
```

Benefits

Rapid Delivery

AWS CodePipeline automates your software release process, allowing you to rapidly release new features to users. With CodePipeline, you can quickly iterate on feedback and

善用第三方伺服器 Monitor 工具

- Server 例外錯誤通知
- Server 倒站通知
- Server 效能監控
- Server Log 收集和查詢

You detected an outage. Now what?

Monitoring tools only get you part of the way to resolving critical incidents. Let PagerDuty unlock the full value of your monitoring stack.



Get Started with a Risk-Free 14-Day Trial

SIGN UP



Don't Stop at Detection

Detecting an issue doesn't mean resolving it. PagerDuty's incident resolution platform sits between your monitoring tools and your on-call responders to make sure your incidents are resolved quickly and efficiently.

TAKE A TOUR

Resolved 14 / All 239

Immediately after an incident

Images

Live Help: Online

Put errors in their place

Full-stack error tracking for all apps in any language

Sign up with GitHub

or

[Get started](#)

14 days free. Takes 5 minutes to set up. No credit card required.

The screenshot shows the Rollbar dashboard interface. On the left, there is a table titled "Top 10 items in last 24 hours" with columns for "24hr Trend", "Count", and "Title". The first item is a "NameError: undefined local variable or method 'this_variable_has_not_been_set'" with a count of 464. To the right of the table is a "view all" button. On the right side of the dashboard, there are two bar charts: "Hourly Error/Critical Occurrences" and "Daily Error/Critical Occurrences". The hourly chart shows a peak in error occurrences around 11:04 AM on 11/24/2014, with a tooltip indicating 99 occurrences.

24hr Trend	Count	Title
	464	#54 NameError: undefined local variable or method 'this_variable_has_not_been_set' fo...
	-	#3 Kill browser
	326	#56 ActionController::RoutingError: No route matches [GET] "/assets/404"
	410	#28 ReferenceError: noncalbrant_function is not defined
	269	#55 NoMethodError: undefined method '[]' for nil:NilClass
	335	#8 some log message

TRUSTED BY



“Rollbar has been a great tool for helping us track and fix bugs. We're able to fix issues before users even notice. We highly recommend Rollbar to all our developer friends.”

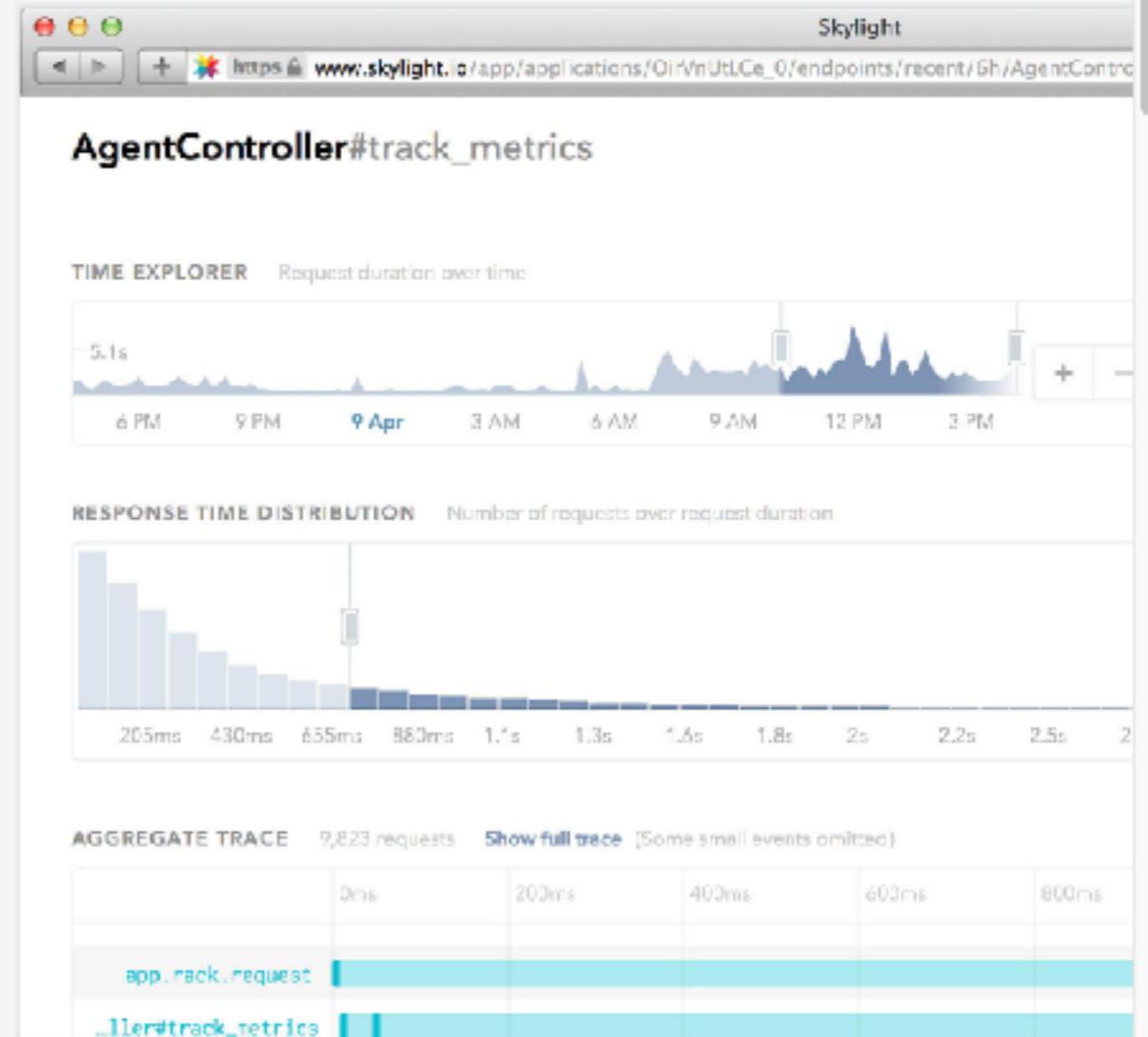


SKYLIGHT IS A
SMART PROFILER
CODE WHISPERER
DATA DETECTIVE
TRENDSPOTTER
FLAG RAISER
FOR RAILS APPS

CREATE AN ACCOUNT

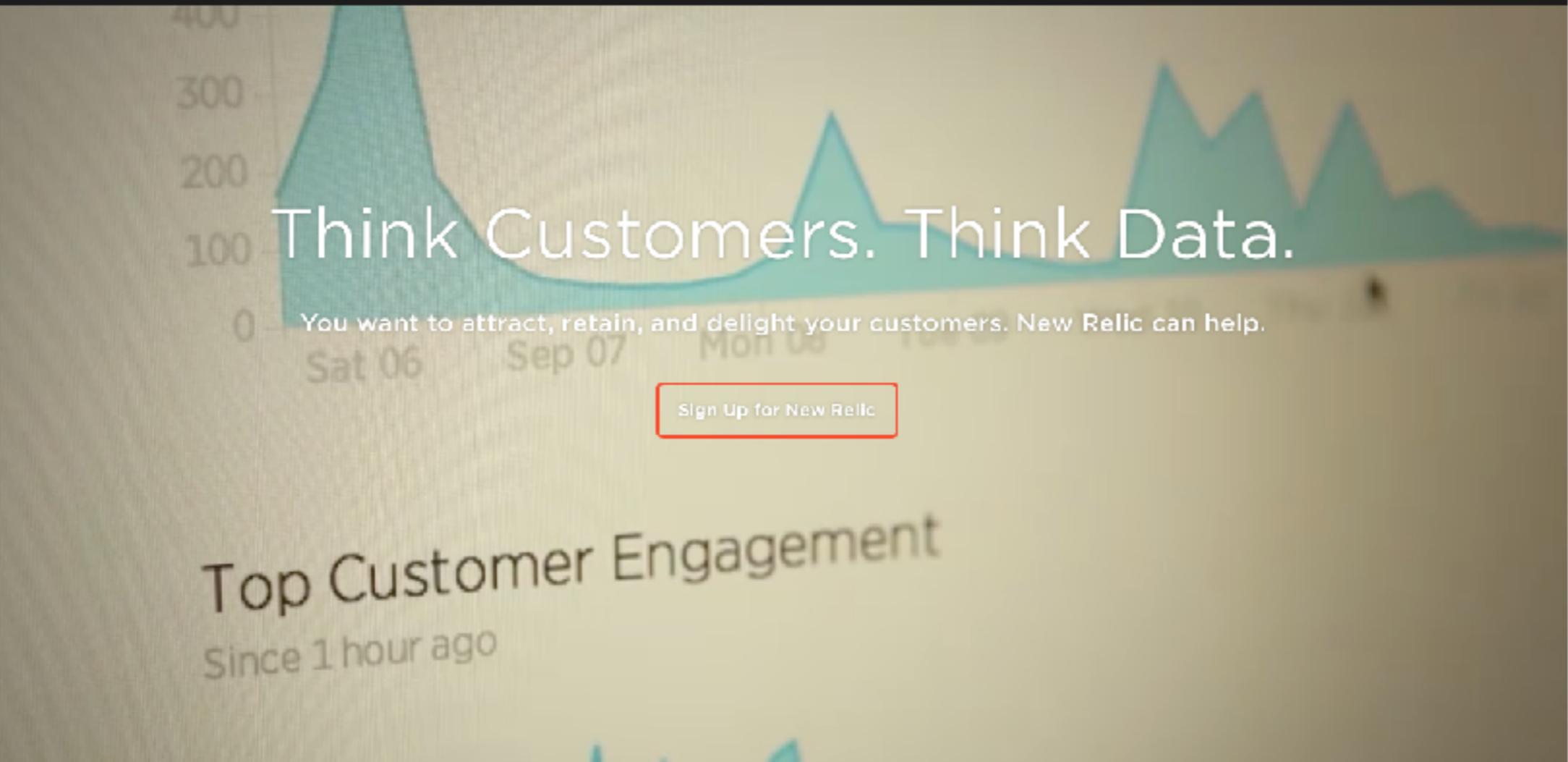


Watch the video



GET ANSWERS

Sign Up for New Relic



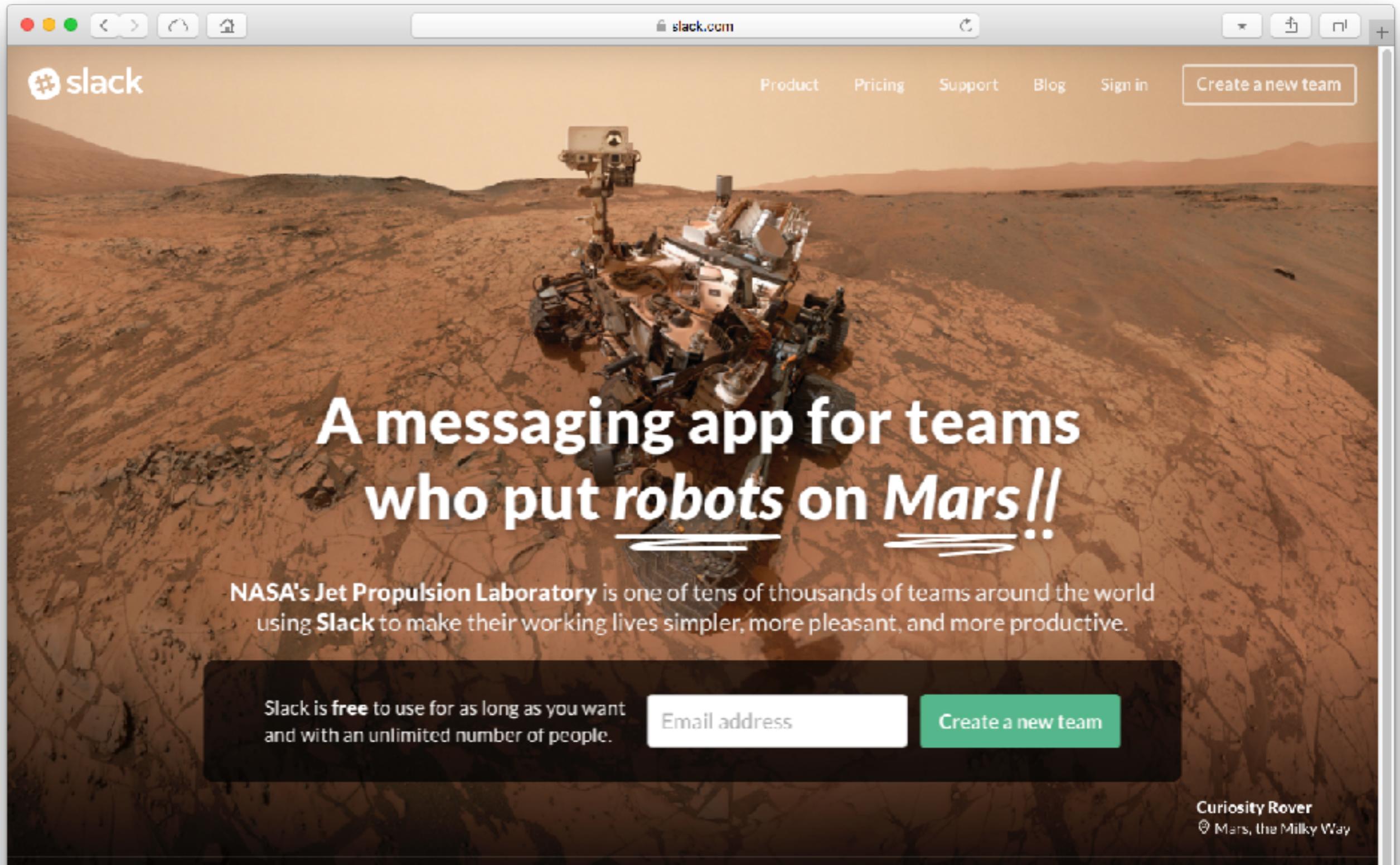
Think Software Analytics

Our platform empowers developers, IT/Ops, and business leaders to transform their business using real-time data directly from production software.



Slack 通訊工具

不要用LINE啦

The image shows a browser window displaying the Slack website. The background is a high-resolution photograph of the Curiosity rover on the surface of Mars. The rover is positioned in the center, facing the viewer. The terrain is a vast, reddish-brown desert with some low hills in the distance. The sky is a hazy, orange-brown color. The Slack logo is in the top left corner. The navigation menu includes links for Product, Pricing, Support, Blog, and Sign in, along with a 'Create a new team' button. The main headline reads 'A messaging app for teams who put robots on Mars!!' with 'robots' and 'Mars!!' underlined. Below the headline, a paragraph states that NASA's Jet Propulsion Laboratory uses Slack. At the bottom, there is a dark grey box containing the text 'Slack is free to use for as long as you want and with an unlimited number of people.' followed by an input field for 'Email address' and a green 'Create a new team' button. In the bottom right corner, there is a small text block identifying the rover as 'Curiosity Rover' and its location as 'Mars, the Milky Way'.

[Product](#) [Pricing](#) [Support](#) [Blog](#) [Sign in](#)

[Create a new team](#)

A messaging app for teams
who put robots on Mars!!

NASA's Jet Propulsion Laboratory is one of tens of thousands of teams around the world using **Slack** to make their working lives simpler, more pleasant, and more productive.

Slack is **free** to use for as long as you want and with an unlimited number of people.

[Create a new team](#)

Curiosity Rover
Mars, the Milky Way

這些訊息都可以整合進 Slack

- 有人 commit
- 有人 code review 留言
- 跑 CI 的結果
- 有人佈署
- 伺服器發生例外錯誤
- 倒站通知

ALPHA Camp

- ihower
- ★ STARRED
- ac-internal
- developers
- # engineering**
- arielchen
- bernard

CHANNELS (18)

- blog
- bootcamp-sg
- bootcamp-tw
- campus
- general
- ios-bootcamp
- ios_instructor
- seminars
- web-bootcamp

DIRECT MESSAGES (16)

- slackbot
- appeter
- benson
- brlan
- edwardchiang
- evelyn330
- kale
- kenny,lew
- tim

+ Invite People

★ #engineering

Yesterday

github BOT 6:37 PM
 [dojo:master] 5 new commits by John Chang and 1 other:

- 748d9ee: Create deadline for jobs - John Chang
- 3b2f925: Optimized coding style - John Chang
- 199ba43: Solve migration pending issue - John Chang
- 54e1617: optimized coding style - John Chang
- e7293ff: Merge branch 'feature/set_deadline_for_opportunity_v2' - Wen-Tien Chang

Deploy bot BOT 6:37 PM
 dojo deploy by ihower: starting.

Status	starting	Stage	production
Branch	master	Revision	
Hosts	dojo.alphacamp.co		

circleci BOT 6:38 PM
 Success: ihower's build (#411): push in ALPHA/Camp/dojo (master)
 - Merge branch 'feature/set_deadline_for_opportunity_v2' (e7293ff) by ihower and 4 more

Deploy bot BOT 6:38 PM
 dojo deploy by ihower: successful in 89 seconds.

Status	success	Stage	production
Branch	master	Revision	e7293ff
Hosts	dojo.alphacamp.co		

Rollbar BOT 6:38 PM
 [Dojo] ihower deployed revision e7293ffa27fd186699cf4e3c129b172db6438767 to production
<https://rollbar.com/deploy/2083091/>

github BOT 6:51 PM

- [dojo] New branch 'feature/set_jobs_options_v2' was pushed by video107
- [dojo] The branch 'feature/set_jobs_options' was deleted by ihower

[ALPHACamp/dojo] New comment on pull request #9: Feature/set deadline for opportunity v2
 Comment by ihower

+ |

About #engineering

Channel Details

Purpose
 bot messages from delivery pipeline and server monitor, including github, circleci and newrelic.

Current topic
[Set a channel topic](#)

Created by you on December 24th, 2014

Pinned Items

1/3 Members

- ihower
- arielchen
- bernard

Shared Files

Notification Preferences

Monitor 數據

- Infrastructure&Application Metrics
 - Server Status (HTTP error rate)
 - Response Time
 - Exception&Error rate
- 任何圖表的改變通常都代表最近的佈署影響，因此必須在每次佈署後密切觀察和比較。

小結

- 自動化部署程序
- 善用第三方 Monitor 和通訊工具 Slack

4. 營運成長

好的產品是營運出來的

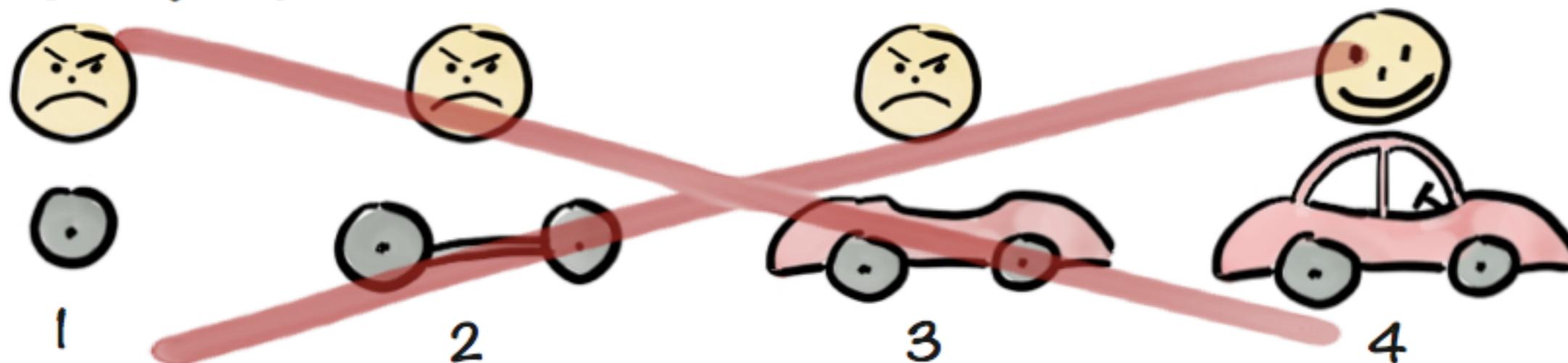
- 目標：引導用戶開始使用，並且持續使用產品
- 透過營運來驅動產品 (Measure and Learn)
 - 面對數據和真實用戶，不斷給產品回饋
- 不只是行銷部門的事情，而是圍繞用戶的一系列工作

為什麼工程師需要知道？

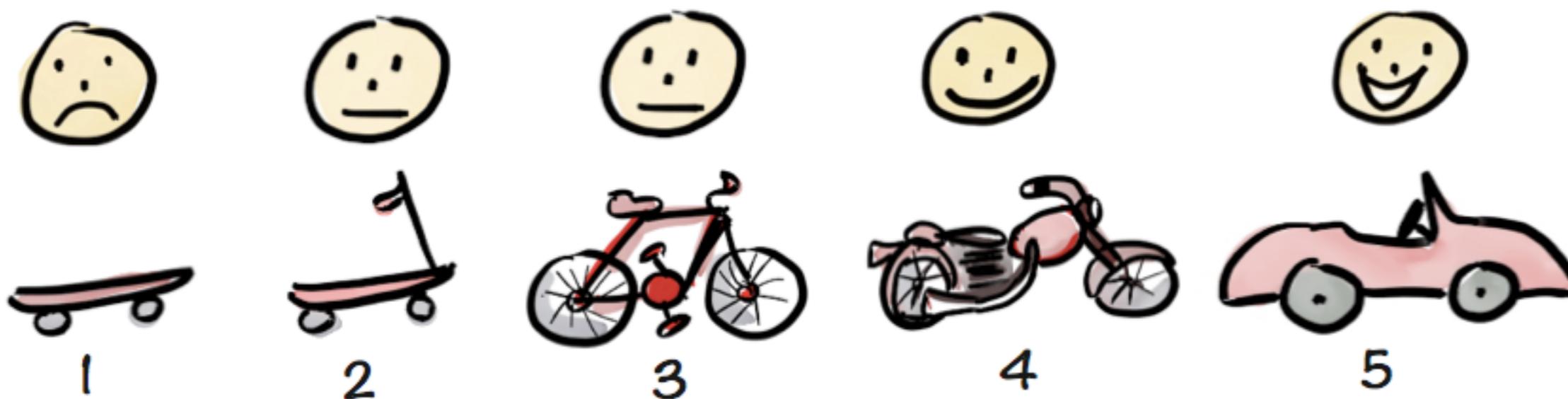
- 才能知道任務的 priority，有時這比一直開發新功能更重要
- 第三方工具需要做 integrations
- 需要工程師協助匯出數據，行銷、營運人員才能拿到
- 要埋 tracking code，需要工程師
- 一旦考慮到營運需求，一個功能就不只是一個功能了
 - 產生 Warn-Up 種子資料
 - 產生 User-facing metrics
 - 做 A/B Testing 實驗

車子做好了，而營運就像開車上路

Not like this....

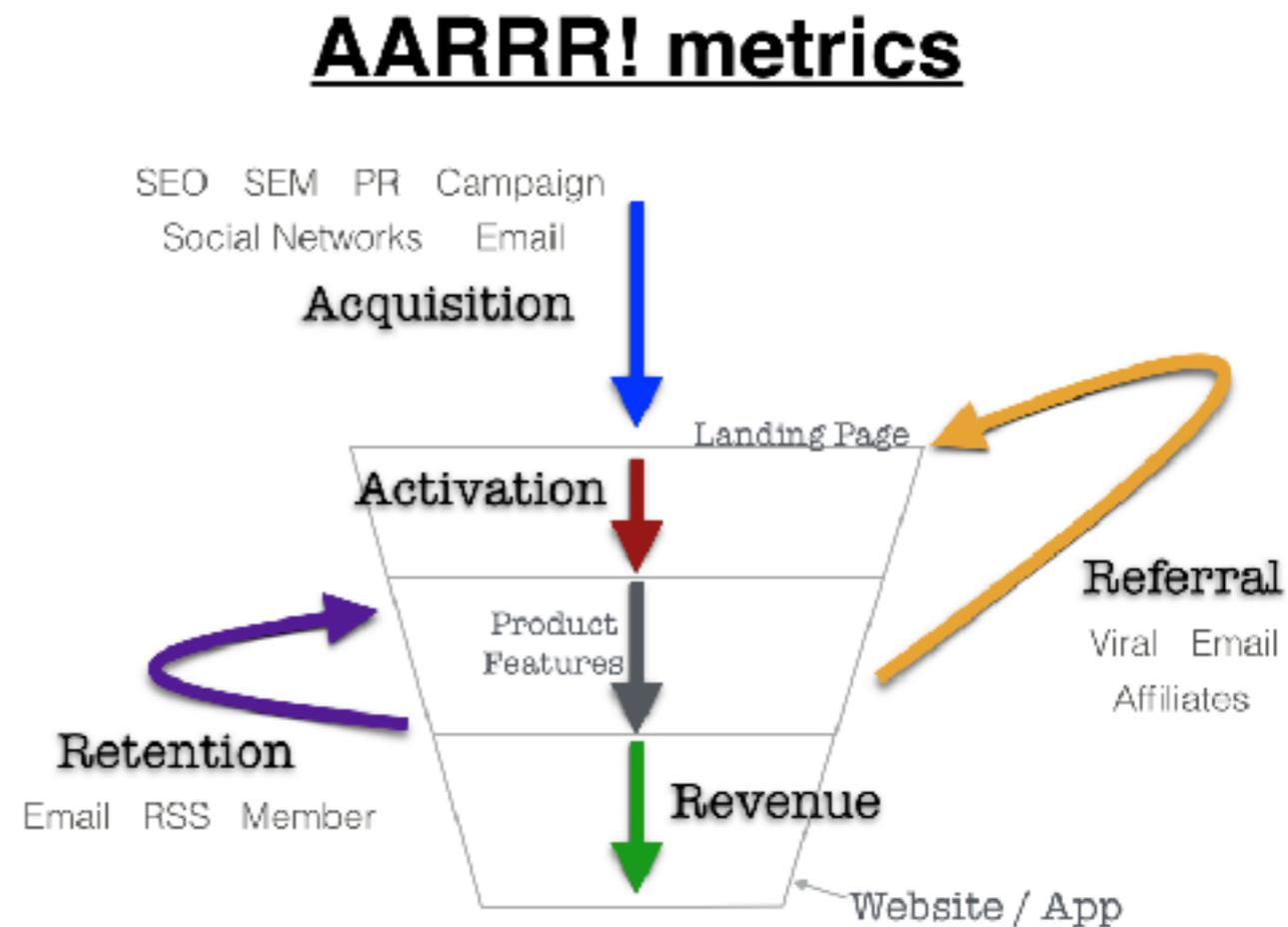


Like this!



營運的 AARRRR 模型

- 獲取用戶
- 成為有效用戶
- 提高留存率
- 用戶推薦
- 獲得收入



by George Chang, modify from Dave McClure's Startup Metrics

Acquisition

得到用戶(signup)

- Owned Media
 - Landing Page
 - Copy Writing
 - SEO
- Paid Media
 - Facebook Ads
 - Google Adwords
 - Facebook fanspage
- EDM
- Earned Media
 - Community Share
 - 搭順風車

Activation

引導客戶開始使用你的產品

- 開始使用你的最重要功能
- 可以設計 On Boarding 流程
 - 教育客戶、展現產品價值
 - 做太長，跳出率高
 - 做太少，客戶不會用就流失掉了
- 或是 sandbox 環境，例如 Basecamp 的範例 project

UserOnboard



- Home
- Teardowns
- Advice
- Training
- Merch
- Contact

Basecamp.com, 11/2013

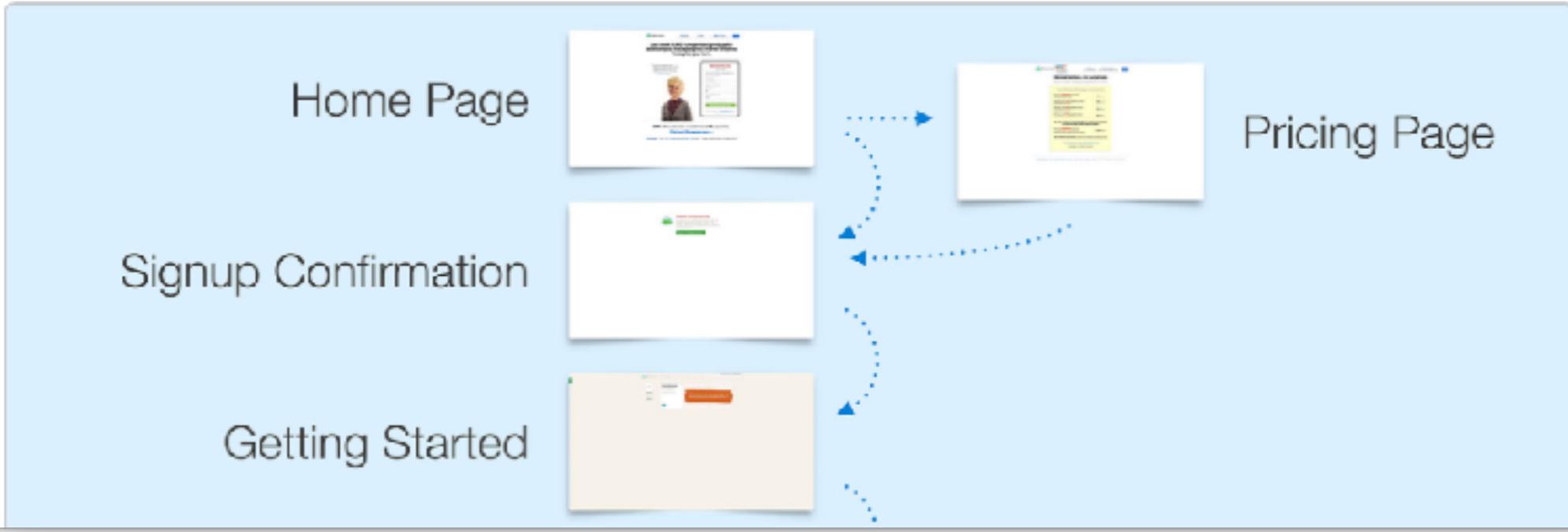
37signals is renown for producing clean, no-nonsense products, and their onboarding experience is exactly that. They also have a very cool tactic for getting new users acquainted with the interface!

Without further ado, here is...

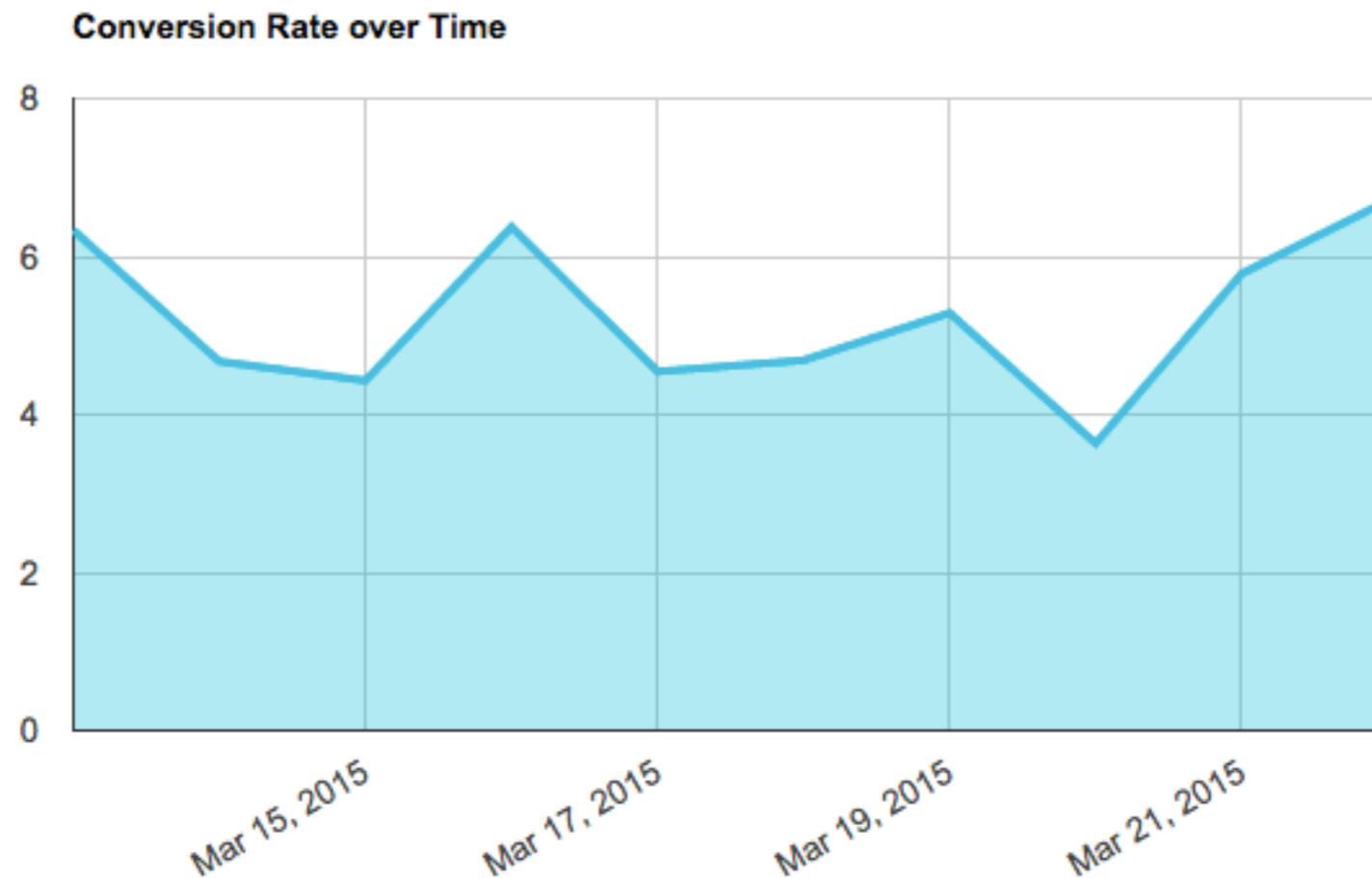
How Basecamp Onboards New Users



2

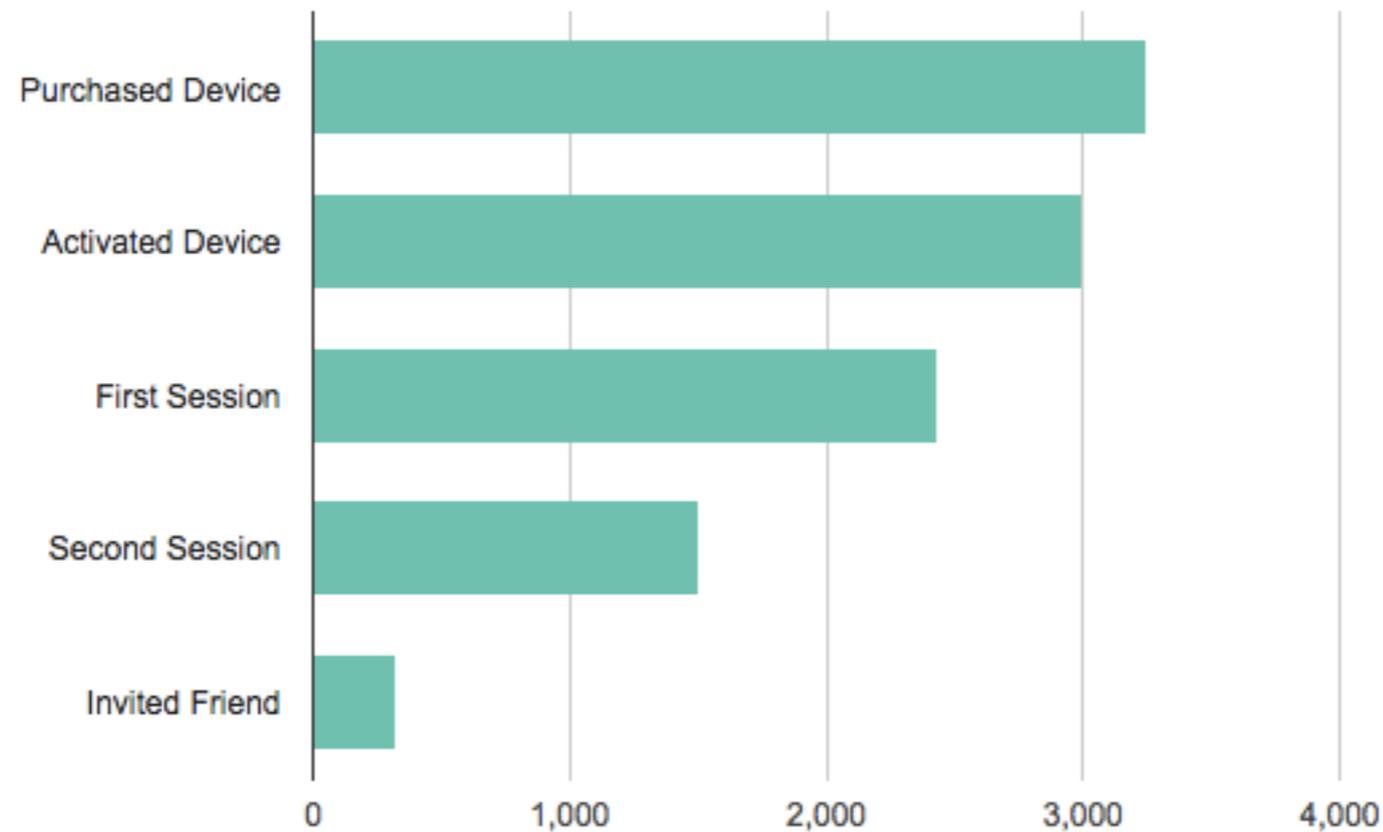


Conversion Rate



<https://keen.io/guides/conversion-analysis/>

Conversion Funnel

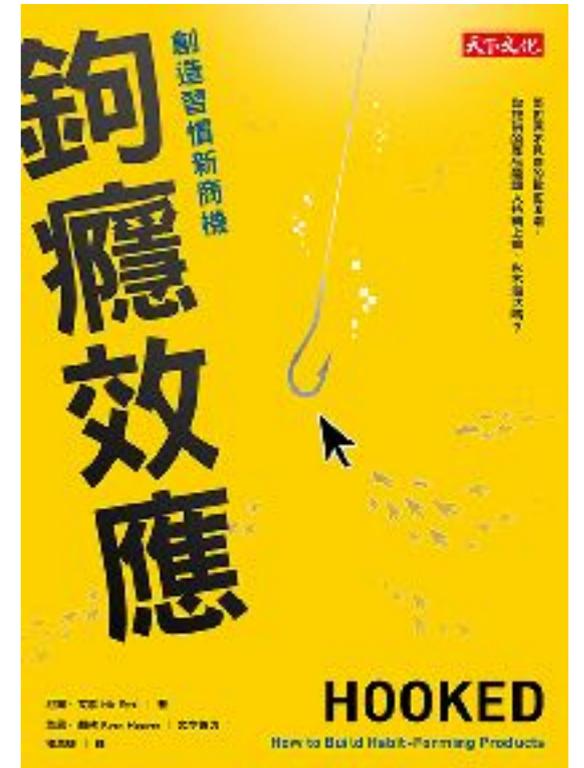


<https://keen.io/guides/intro-to-funnel-analysis/>

Retention

讓用戶繼續使用

- Active Users (DAU/WAU/MAU)
- Retention rate = 留存客戶數/新用戶數
 - 運營成功的關鍵
- 如何召回客戶: 常見用 Email
- 鉤癮效應、Gamification 和激勵體系 Reputation System
 - 上癮 (addiction) = 觸發 (trigger) + 行動 (action) + 獎賞 (reward) + 投入 (investment)



Retention rate

Cohort Week	Size	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
2012-12-24	22	5%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
2012-12-31	16	25%	25%	25%	25%	25%	25%	25%	50%	50%	50%	50%	50%	50%	25%	
2013-01-07	18	28%	60%	80%	60%	40%	40%	40%	60%	80%	60%	60%	60%	60%		
2013-01-14	33	24%	100%	100%	88%	100%	113%	125%	125%	113%	100%	100%	100%			
2013-01-21	40	28%	91%	91%	82%	91%	91%	109%	91%	82%	82%	82%				
2013-01-28	44	20%	67%	56%	44%	44%	44%	56%	44%	44%	44%					
2013-02-04	62	11%	86%	71%	86%	57%	86%	86%	71%	71%						
2013-02-11	37	16%	50%	67%	50%	50%	50%	67%	50%							
2013-02-18	41	22%	56%	56%	67%	56%	56%	56%								
2013-02-25	65	14%	78%	111%	89%	78%	78%									
2013-03-04	149	12%	72%	61%	39%	50%										
2013-03-11	89	18%	81%	81%	81%											
2013-03-18	73	21%	87%	87%												
2013-03-25	74	20%	80%													
2013-04-01	71	23%														

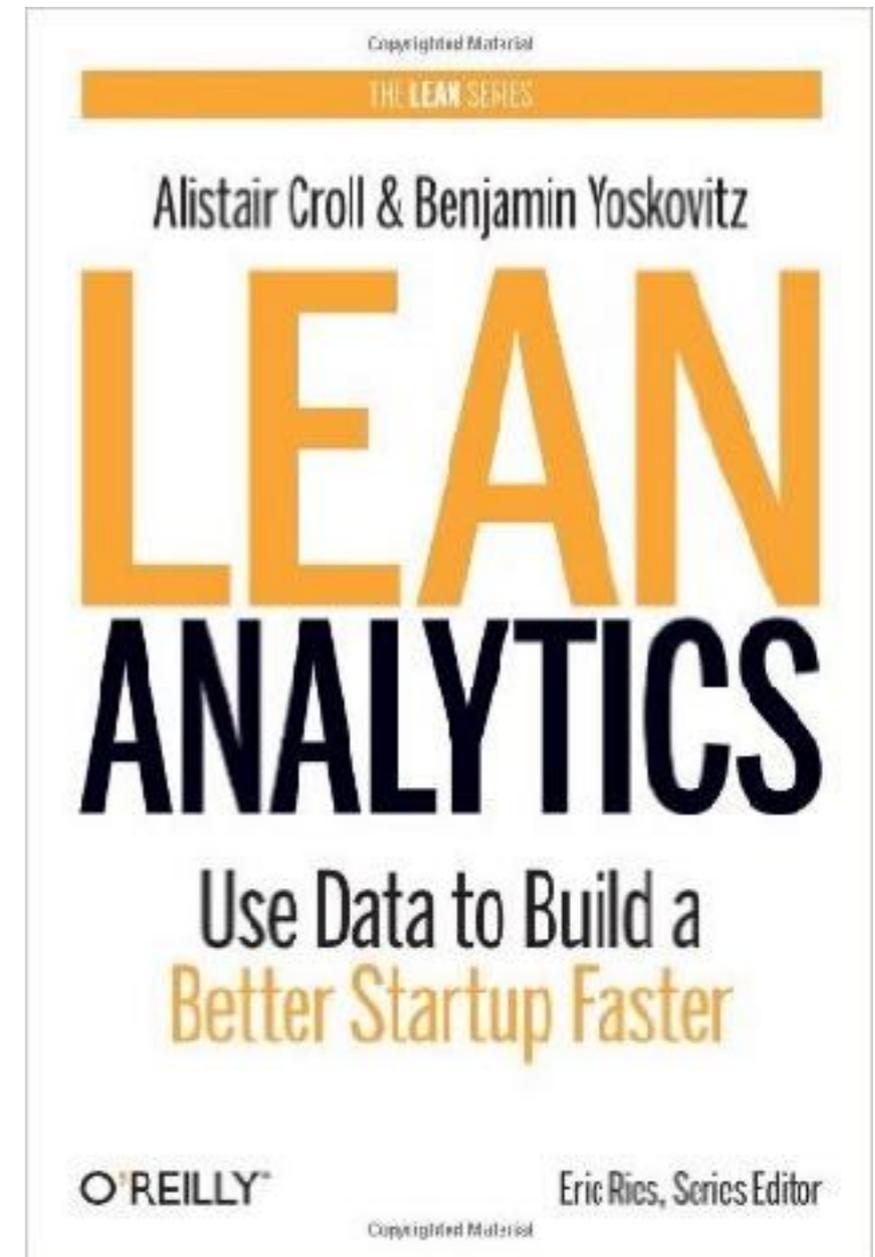
<https://keen.io/blog/47823687779/how-to-do-a-retention-analysis/>

1. 節流 2. 開源 3. 增加活躍

- 節流 (證明 MVP 達到 PMF)
 - 防止用戶流失、召回流失客戶：增加留存率(Retention rate)
- 開源 (爆發成長 Growth)
 - 增加訪問量 UV (Unique Visitor)
 - 提高轉換率 Conversion Rate
- 增加活躍 (產品品質 UX)
 - 增加用戶使用主功能的頻率(活躍度) or 增加客單價(增加 upsell)

根據不同類型的產品

- E-commerce
- SaaS
- Free Mobile App
- Media Site
- User-Generated Content
- Two-sided Marketplaces



行銷自動化工具

- Landing Page
 - 這年頭產品都還沒蓋，就先做好 Landing Page 收集 signup 了!!
- Segment your customers
 - Email marketing
- Customer Support

Build, Publish & A/B Test Landing Pages Without I.T.

The Mobile Responsive Landing Page Builder for Marketers

BUILD A HIGH-CONVERTING LANDING PAGE NOW



1. Build a Page



2. Publish It

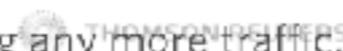


3. Test & Optimize



SOME OF THE WORLD'S MOST RESPECTED BRANDS AND AGENCIES TRUST UNBOUNCE

"We were able to test our way from a 5% conversion rate, all the way up to 20%. Without driving any more traffic, our client is getting four times the leads that he was getting before."



▶ WATCH VIDEO

Unbounce is an indispensable tool for Andrew Miller's





Features

Pricing

Support

Blog

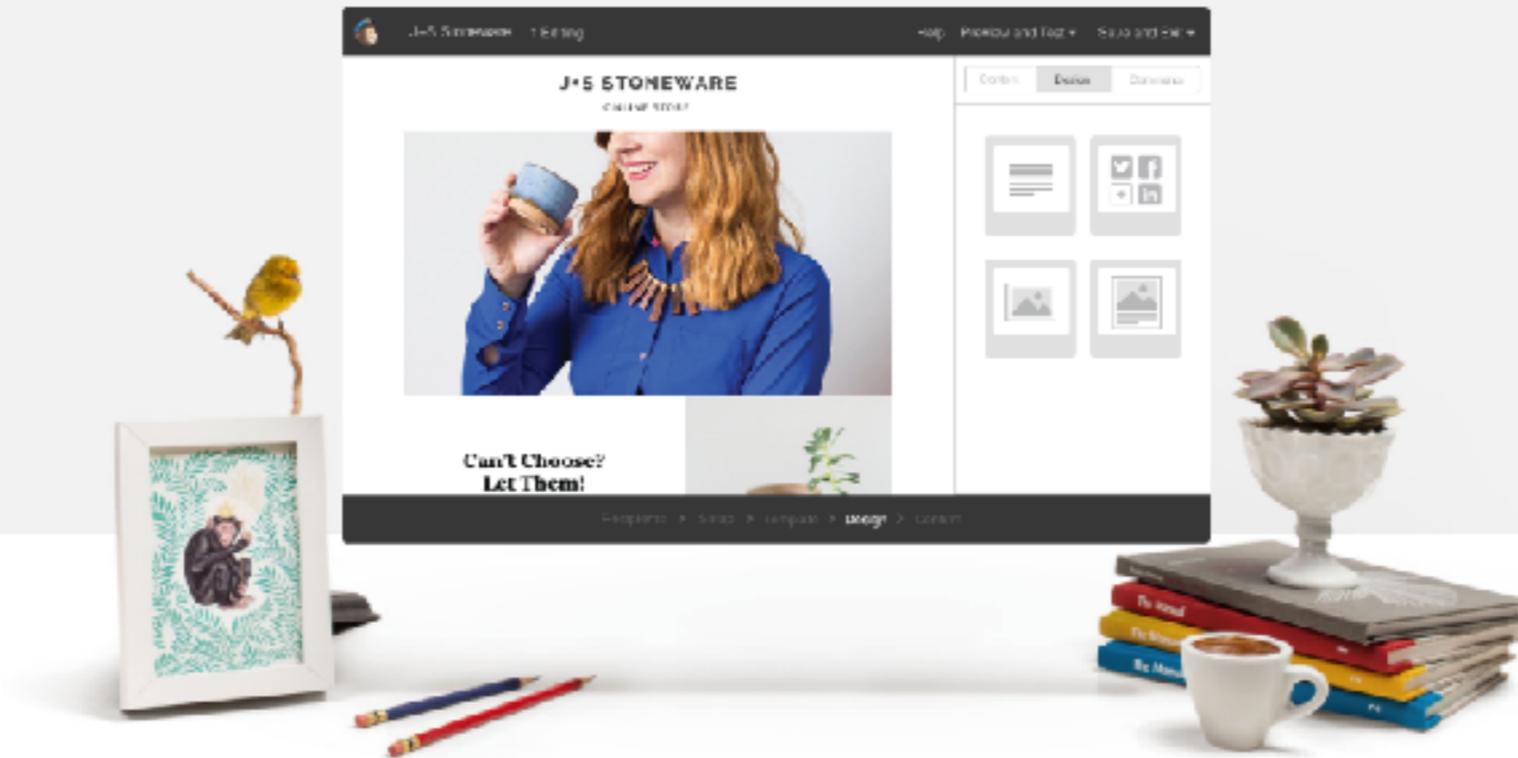
More

Sign Up Free

Log In



Send Better Email



Join more than 10 million people who use MailChimp to design and send 600 million emails every day.

Sign Up Free

MailChimp

©2001-2016 All Rights Reserved. MailChimp® is a registered trademark of The Rocket Science Group. [Privacy](#) and [Terms](#)

A platform for sending targeted messages.

All of your outbound communication in a single beautiful interface.

[Sign Up Free](#)

Start for free with our unlimited trial for up to 200 profiles and 400 emails every month.



Communicating with customers shouldn't be this hard.

THE OLD WAY



THE NEW WAY



With Intercom it's simple and personal for everyone.



[Login](#) [Support](#) [Company](#) [English](#)

[PRODUCTS](#) [PRICING](#) [WHY ZENDESK](#) [DEMO](#) [CUSTOMERS](#) [RESOURCES](#)

BUSINESSES ARE MADE OF RELATIONSHIPS

Those relationships can be complicated, but Zendesk helps make them better



RELATIONSHIPS ARE COMPLICATED

TRY ZENDESK FOR FREE

Good relationships are better

Our software helps companies provide a great overall customer experience, and build customer relationships that are more meaningful, personal, and productive.

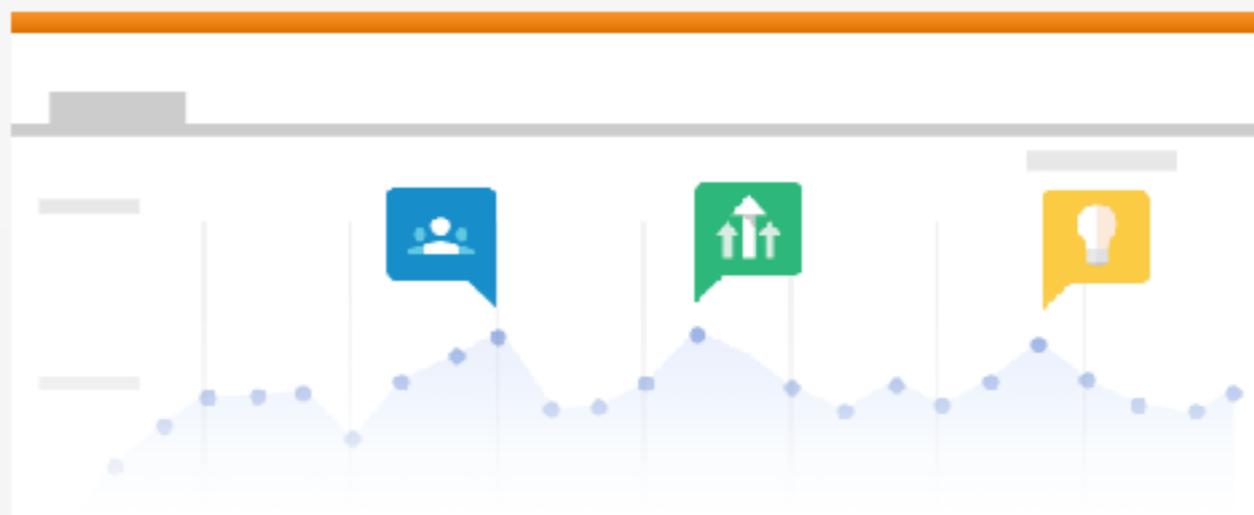


用戶分析工具

- 基本流量分析: Google Analytics
- 客戶行為定量分析: Funnel analytic
 - Mixpanel
 - Keen.IO
- 客戶行為定性分析: Heatmapping 和 Customer Survey

取得重要的分析數據

Google Analytics (分析) 不只能讓您估算銷售量和轉換率，還能提供最新的分析資料，方便您瞭解訪客如何使用您的網站、他們如何前來，以及如何吸引他們持續回訪。



分析工具

Google Analytics (分析) 以一種強大、易於使用的報表平台為基礎，讓您只要按幾下就能決定要查看的資料並自訂報表。

[瞭解詳情](#)

行動分析

Google Analytics (分析) 方便您評估行動裝置對業務的影響。此外，如果您建立了行動應用程式，Google Analytics (分析) 也提供 iOS 和 Android 專用的軟體開發套件，方便您評估他人如何使用您的應用程式。

[瞭解詳情](#)

內容分析

「內容」報表方便您瞭解網站的哪些部分成效較佳、哪些網頁最受歡迎，進而為客戶改善瀏覽體驗。

[瞭解詳情](#)

轉換分析

Google Analytics (分析) 提供一系列的分析功能，可指出您吸引到多少客戶、銷售量以及使用者與您網站互動的方式。

[瞭解詳情](#)

社交分析

網路是一種社交場合，而 Google Analytics (分析) 可為您評估社交媒體程式的成效。您可以分析訪客如何與網站上的共用功能 (例如 Google +1 按鈕) 互動，也可分析訪客如何接觸您在各種社交平台上的內容。

[瞭解詳情](#)

廣告分析

瞭解您的社交、行動、搜尋和多媒體廣告的成效，讓廣告發揮最大效益。您可以將網站活動與行銷廣告活動相連結，藉此完整掌握並改善廣告成效。

[瞭解詳情](#)

Actions speak louder than page views.

Learn how people use your app with the world's most advanced mobile & web analytics

GET STARTED



Turn data into insights for your teams & customers.

With Keen's developer-friendly APIs, it's easy to collect, explore, and visualize data anywhere. Apps and websites, customer-facing dashboards, IoT devices, you name it.

[Start Now](#)



COMMUNITY [Keen Slack Chat](#). Join the conversation on Slack! [Chat now!](#)

Get granular insights

With Keen, you get precise results — not samples — so you can control your data and build reliable reports.

Differentiate with data

Show your customers the value you provide by embedding a dashboard directly in your app or website.

Focus on product

Stay committed to your core product. Leave the analytics — and the infrastructure maintenance — to us.



TRY IT FOR FREE

SIGN IN

All-in-one Analytics and Feedback.

Hotjar is a new and easy way to truly understand your web and mobile site visitors.
Identify your hottest opportunities for improvement using:



HEATMAPS



RECORDINGS



FUNNELS



FORMS



POLLS



SURVEYS



RECRUITERS

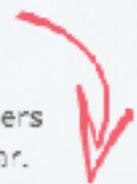
TRY IT FOR FREE

No credit card required.

From startups to million-dollar companies...
99,671 Organizations in 184 countries improve their sites with Hotjar.



See what our users say about Hotjar.



What our users are saying:

"You get all the tools you need packaged into one service"

"I am ADDICTED to your tool. EVL We installed it on"

用數據改善產品

Data-Driven Project

- 要量測什麼數據？
- 怎麼量測？
- 怎麼改進？

HEART Framework

<http://www.dtelepathy.com/ux-metrics/>

- Happiness: Net Promoter Score (NPS)
 - How likely are you to recommend us to a friend?
 - 0-6 分: Detractors
 - 7-8: Passives
 - 9-10: Promoter
- Engagement: CTR、Active Users 和功能使用數
- Adoption: 功能的新用戶數
- Retention: 重複使用數、Churn rate
- Task Success: 搜尋成功、Profile 完成數

收集 Event Data

- Event Data = Action + State + Time
- Action 例如：註冊、登入、表單送出、分享、搜尋、投票、購買、加入最愛、瀏覽
- State 例如：User、Device、Item、Record、Product、Form 等等

<https://keen.io/guides/data-modeling-guide/>

A/B Testing 實驗

- 本來版本 vs. 新改的版本
- 要同時測試不同版本，才可以排除前後時間的外部因素
- 實驗分組必須 randomness 和 independence
- 注意統計顯著性 significance testing (用統計 p-value)
 - 實驗需要人數、時間長度、準確性：三選二
- 佈署可能需要搭配 feature flag
- 小心 Big changes release: Facebook 的改版故事
- PR 信任問題 eg. Facebook 故意 crash 用戶 app

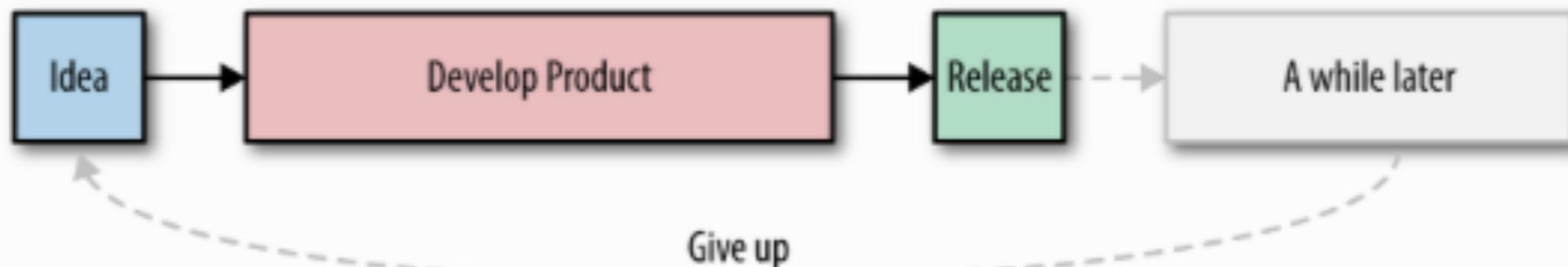


Figure 4-3. Typical product release process (diagram based on a talk by Dan McKinley [McKinley 2014a])

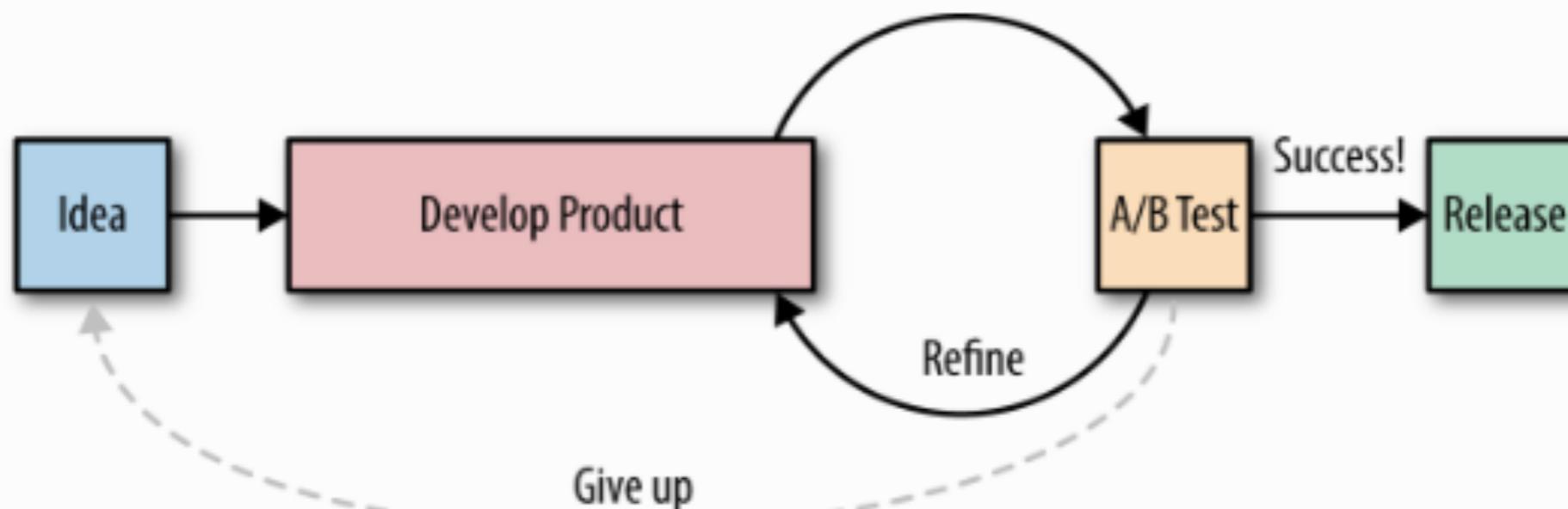
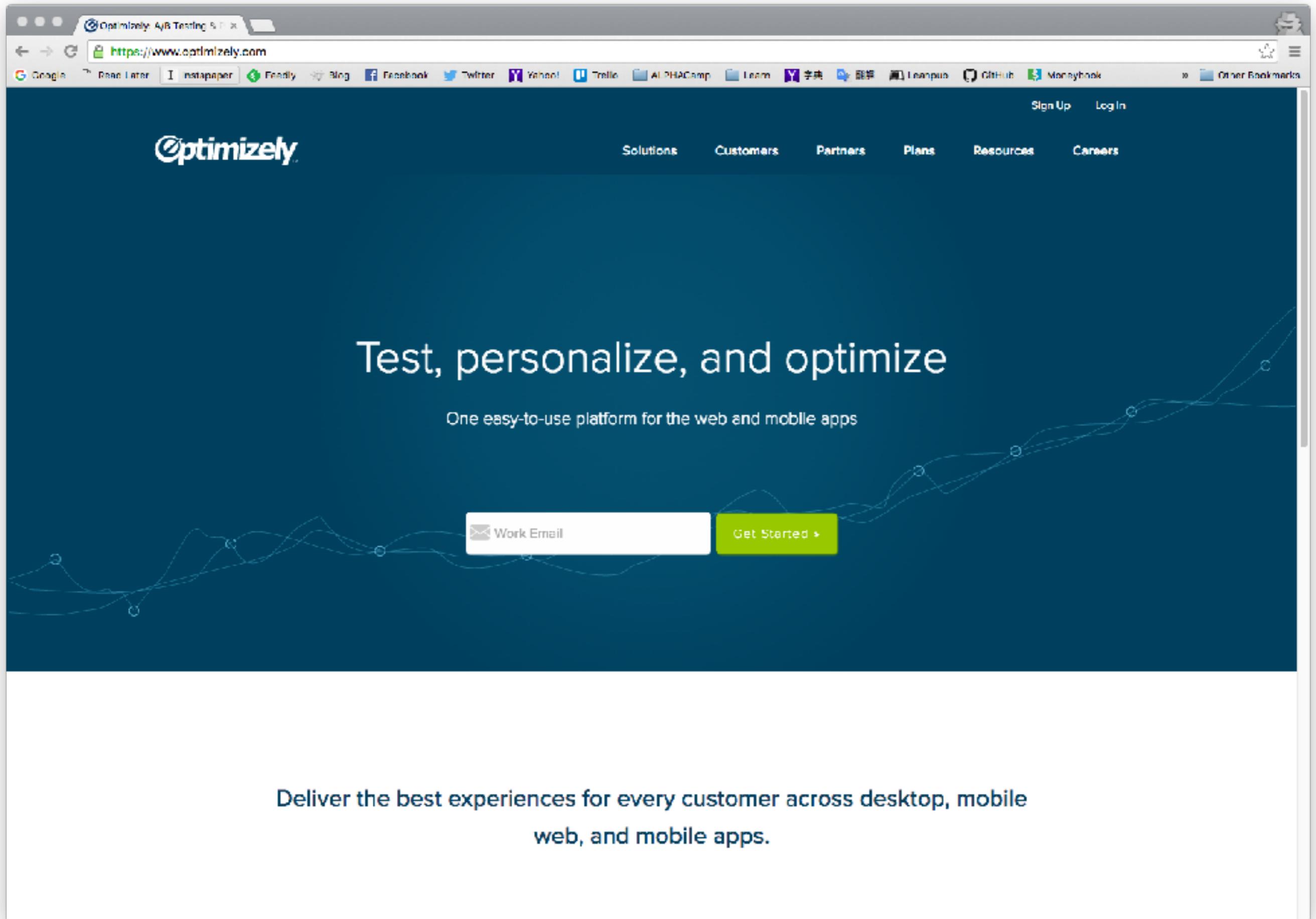


Figure 4-5. Data-driven development (diagram based on a talk by Dan McKinley [McKinley 2014a])



- [Solutions](#)
- [Customers](#)
- [Partners](#)
- [Plans](#)
- [Resources](#)
- [Careers](#)

[Sign Up](#) [Log In](#)

Test, personalize, and optimize

One easy-to-use platform for the web and mobile apps

[Get Started +](#)

Deliver the best experiences for every customer across desktop, mobile web, and mobile apps.

小結

- 了解營運 AARRR
- 整合第三方行銷自動化工具
- 整合第三方用戶分析工具
- 用數據改進產品

補充

- 數據資料分類：OLTP 和 Clickstream, user events (可用 Mixpanel, GA)
 - 可能需要合併 OLTP 和 clickstream data 進行分析
- why 不自己蓋分析工具? 1. scale 2. 提供讀取 UI 給非技術人員使用
- 少了數據就像開車沒有儀表板，車子會跑但不知道開的方向對不對。
- 這跟 Data Science 的目的不太一樣喔。資料科學的重點還包括 Data Mining。但是 startup 沒有什麼 data 可以挖出 insight。

最後一些思考

Ask the Product Manager



play 0~1:XX and 3:00 ~

<https://www.youtube.com/watch?v=cYwXkzPVhP0>

工程師文化？

- 作為一個好 PM 的不二心法當然就是：『讓工程師可以專心快樂去寫 code』？
- 讓產品經理 do the right thing，讓工程師 do the thing right？
- 小孩子把書唸好就好，讓大人決定以後要做什麼？

<https://blog.alphacamp.co/2014/12/17/good-product-manager/>

<https://blog.alphacamp.co/2016/03/21/the-importance-of-project-manager/>

我們失敗了- 完美的團隊也會失敗

「最後一年，我們幾乎沒有任何一個 sprint failed，到後期的 planning meeting 能在短短一到兩個小時完成。」

「但即使如此，我們還是失敗了。」

我也曾經有類似的經驗，跑 scrum 跑很順，每個 sprint 都非常有效率，但是公司最後倒了。

Common Agile Practice Isn't for Startups

- Common Agile Practice is for Delivery
- Most Startups are About Discovery
- Keep the spirit of Agile, and discount the dogma

7 Agile Best Practices that You Don't Need to Follow

- TDD
- Pair Programming
- Emergent Design and Metaphor
- Daily Standups
- Collective Code Ownership
- Writing All Requirements as Stories
- Relying on a Product Owner

關注全局

- 主動發現問題、瞭解問題、解決問題
- 做的東西不對，開發再有效率也是浪費
- 軟體工程和軟體開發方法論，假設 PM 和客戶會告訴你要建造什麼，然後如何(局部)最佳化工程面。至於開發前的要做什麼，上線後如何營運，不是開發團隊的事情.... orz
- 但在 startup 裡面，決定要做什麼才是最關鍵的

參與產品設計

- 在 startup 裡，PM 也不知道做什麼才是對
- 在 startup 裡，要做什麼，scope 有多大，非常彈性可以商量，可以也應該要納入工程端的想法
- 因此，參與新產品開發，不要把工程師只當做工人，而是一起積極參與產品規劃。
- 大家都是透過 Metrics 數據來學習驗證結果

參與產品設計 (cont.)

- PM 知道有什麼問題要解決，但是其實不一定不知道最好的解法是什麼
- 如果只讓 PM 寫 User Stories，就會寫出很爛的規格，然後工程師瞎做，最後生產出垃圾 App

參與產品設計 (cont.)

- 怎麼辦? 討論 User Stories 時，讓 PM 告訴工程師：
 - 我們遇到了什麼問題? 為什麼這件事會發生? 我們目前的限制是什麼?
 - PM 提出他建議的解決方案，工程師也提出建議的解決方案，激盪出最好最適切的解決方案

參與產品設計 (cont.)

- Why?
 - 重現「原始目的」與「原始情境」 讓團隊了解這個問題的資源限制
 - 一起作出最好的建議，包括 priority 優先權
 - 之後開始實作，就算規格寫不清楚的地方，工程師也可以腦補出正確的規格。

謝謝，請多指教

<https://ihower.tw>

- 需求收集
 - Lean Startup: MVP
 - 用 User Stories 描述
 - 善用線上協同工具 Quip 或 Hackpad
- 實作開發
 - 專案管理: Scrum 或 Kanban
 - 不重複發明輪子
 - 用 Wireframe 做設計
 - 寫自動化測試
 - 用版本控制系統 Git 搭配 Github flow 或 git flow
- 佈署上架
 - 自動化部署程序
 - 善用第三方 Monitor 和通訊工具 Slack
- 營運成長
 - 了解營運 AARRR
 - 整合第三方行銷自動化工具
 - 整合第三方用戶分析工具
 - 用數據改進產品
- 最後：在 Startup 就要關注全局，參與產品設計與營運



參考資料

- Udacity 的 Product Design 課程
- Data-Driven Product
 - <http://shop.oreilly.com/product/0636920040149.do>
- hello, startup
- 互聯網運營之道
- 用戶力
- 成長駭客
- Intro to Growth Hack
 - <http://courses.growthschool.com/>