開發團隊的敏捷之路(未完成)

twMVC#24 2016-10-01 Kevin Tseng



簡介

- 微軟最有價值專家（MVP）2013~2016
- twMVC 核心成員及講師 （http://mvc.tw）
- SkillTree 專任講師 （http://skilltree.my）
- 部落格「mrkt 的程式學習筆記」
  http://kevintsengtw.blogspot.tw
- Google+ 專頁
  https://plus.google.com/+KevintsengtwBlogspot
- Github
  https://github.com/kevintsengtw

主題說明

- 很多團隊存在很多現實的問題，會嘗試著使用很多方式去解決，這幾年敏捷開發已成主流，似乎這些困擾著團隊的問題就可以迎刃而解…

  許多企業和開發團隊也喊著要導入敏捷，卻對敏捷開發存有嚴重誤解；聽了許多敏捷的課程，也看了一堆書，真的就可以完全瞭解並完成嗎？

主題說明

- 這次的內容不是要說什麼是敏捷

  也不是要分享團隊如何成功建立並且完成敏捷開發

  而是分享一個團隊邁向敏捷開發卻尚未完成的過程。

Agenda

- 敏捷開發
- 部門初期做了哪些事情
- 資訊開發團隊做了哪些改變
- 開發人員因應的作為

敏捷開發

敏捷最讓人誤會的是「敏捷」兩個字

- **敏捷開發並不是讓開發變快、變得迅速**
- **敏捷開發在於強調能夠快速的反應需求變化**
- 頻繁且有週期地持續交付可用的軟體，

  快速得到回饋，讓開發可以快速修正
- 敏捷的重點在於：

  消除浪費、過程透明、交付正確可用的成品

敏捷軟體開發

- **敏捷軟體開發**（Agile software development），又稱**敏捷開發**，是一種從 1990 年代開始逐漸引起廣泛關注的一些新型軟體開發方法，是一種應對快速變化的需求的一種軟體開發能力。

敏捷軟體開發

- 它們的具體名稱、理念、過程、術語都不盡相同，相對於「非敏捷」，更強調程式設計師團隊與業務專家之間的**緊密協作**、面對面的**溝通**（認為比書面的文檔更有效）、**頻繁交付**新的軟體版本、**緊湊而自我組織型的團隊**、能夠很好地**適應需求變化**的代碼編寫和團隊組織方法，也更注重軟體開發過程中人的作用。

敏捷軟體宣言

- **個人與互動** 重於 流程與工具
- **可用的軟體** 重於 詳盡的文件
- **與客戶合作** 重於 合約協商
- **回應變化** 重於 遵循計劃

其中左邊的描述是右邊原則的重點。

(雖然右邊有其價值，但左邊的項目更為重要)

敏捷宣言遵循的原則

- 一共有 12 個項目
- 分為「交付」「溝通」「專案執行」「持續改善」四種，每一種各為 3 個項目

敏捷宣言遵循的原則 – 交付

1. **儘早並持續交付有價值的軟體來滿足客戶的需求**。
2. **歡迎需求的變化**，既使到了開發後期，**敏捷過程能夠駕馭變化，為客戶創造競爭優勢**。
3. **經常交付可用的軟體**，交付間隔可以從幾個星期到幾個月。

敏捷宣言遵循的原則 – 溝通

4. 在整個專案的開發過程中，業務人員與開發者必須要能夠每天在一起工作。
5. 依靠鬥志高昂(積極)的人來建置專案，**給予他們所需的環境與支援，並信任他們可以完成工作**。
6. **面對面的溝通**是傳遞資訊給開發團隊及團隊成員之間效率最高且效果最佳的方法。

敏捷宣言遵循的原則 – 專案執行

7. **可用的軟體**是衡量專案進度的主要指標。
8. **敏捷流程應該能夠保持可持續的發展**。出資者、開發者及使用者應當能不斷地維持穩定的步調。
9. **持續追求優越的技術與優良的設計，以強化敏捷性**。

敏捷宣言遵循的原則 – 持續改善

10. 精簡（或最大化未完成工作量之技藝）是不可或缺的（Simplicity—the art of maximizing the amount of work not done—is essential）

「**盡其所能地讓解決方案簡單化**」

11. **自我組織的團隊**才能創造優秀的架構、需求與設計。

12. **團隊定期自省如何更有效率，並據之適當地調整與修正自己的行為。**

敏捷開發的幾個重點

- 儘早並持續交付，短週期間隔經常交付可用的軟體
- 駕馭變化
- 溝通
- 持續追求優越的技術與優良的設計
- 自我組織的團隊
- 定期自省，並適當地調整與修正

Microsoft Visual Studio 2012 Ultimate

SCRUM 敏捷開發

Product Owner



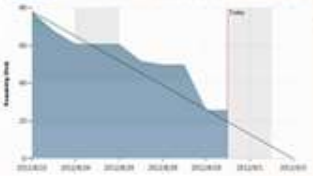
Scrum Master



Team Foundation Server 白板



燃盡圖



Daily Standup 15min

Sprint 1-4weeks



Team Foundation Server-Backlog
Managment & Capacity Planning















Product Backlog

Sprint Planning
part1
1-4h

Sprint Planning
part2
1-4h

Sprint Backlog

Product Backlog Refinement
5-10% of Sprint

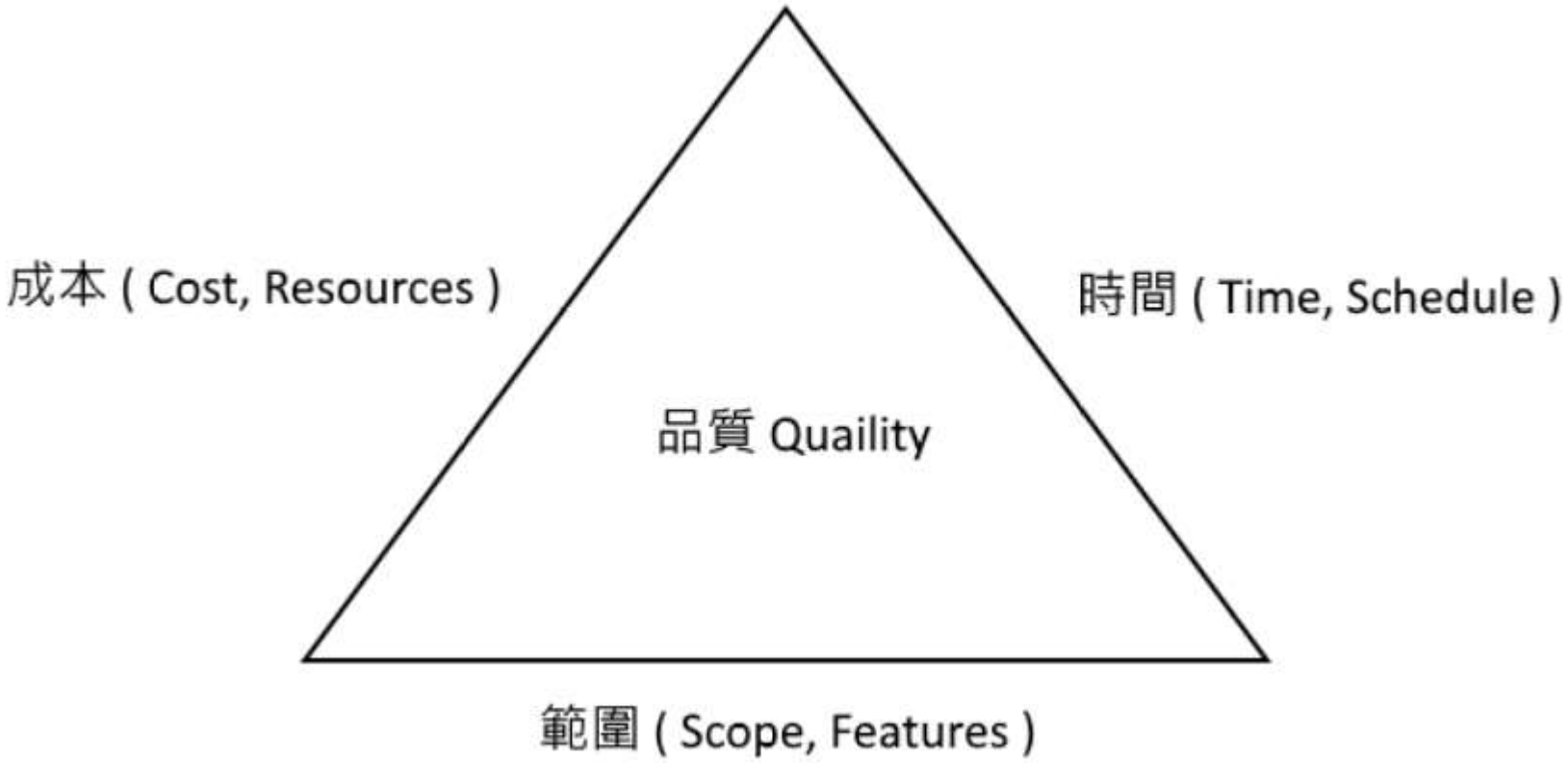
產品交付

Stakeholders

Feedback

更多SCRUM 敏捷開發，都在ALM 線上學習中心
http://aka.ms/learnalm

專案管理三角(The Project Triangle)



專案管理三角(The Project Triangle)

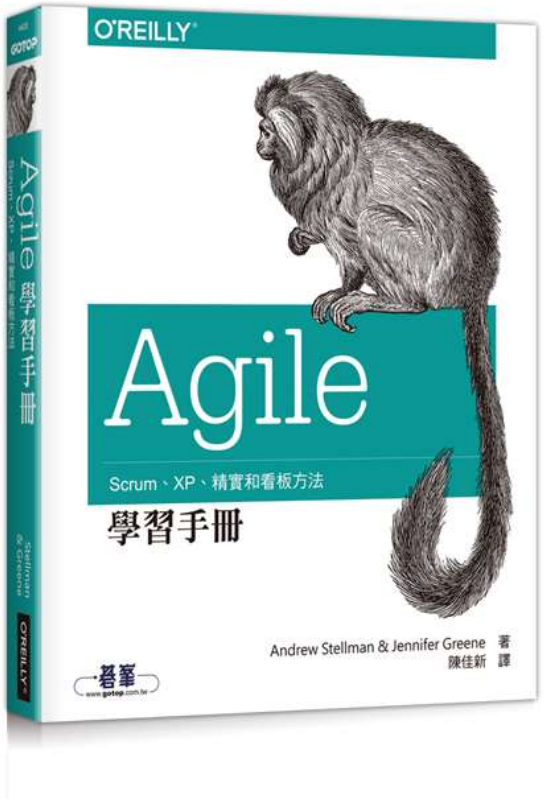
- 如果**時間（時程）**無法改變，就必須耗費更多的成本或是要減少範圍來達成
- 如果要在**成本**內完成，就必須要將時程延後或是減少範圍來完成
- 如果不能將**範圍**減少，就必須要考慮加長時程或增加人力成本來完成
- **品質**是專案三角的第四部分，更動哪一邊都會受到影響

推動敏捷開發所面臨的挑戰

- 團隊成員的接受度與技術能力
- 團隊既有的開發流程
- 開發團隊既有環境的限制
- 各級主管的認知與支持
- 公司文化

推薦參考書籍

- Agile – Scrum、XP、精實和看板方法 學習手冊

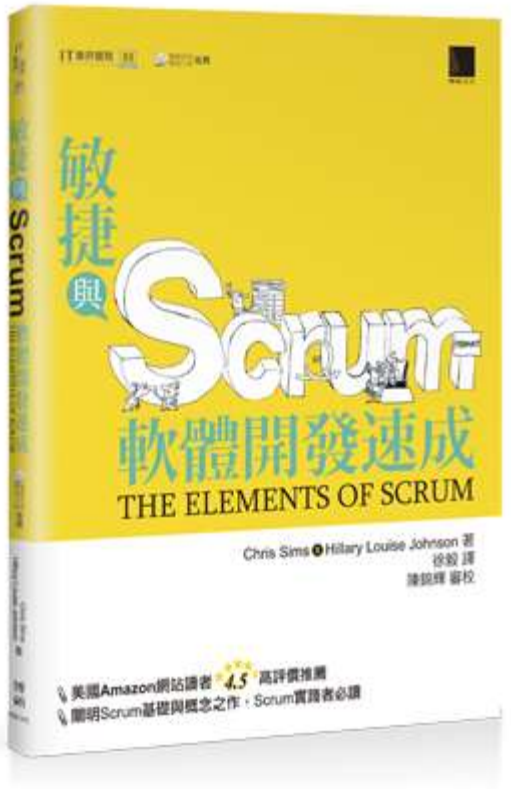


推薦參考書籍

- The Elements of Scrum

  敏捷與 Scrum 軟體開發速成

  (正體中文)

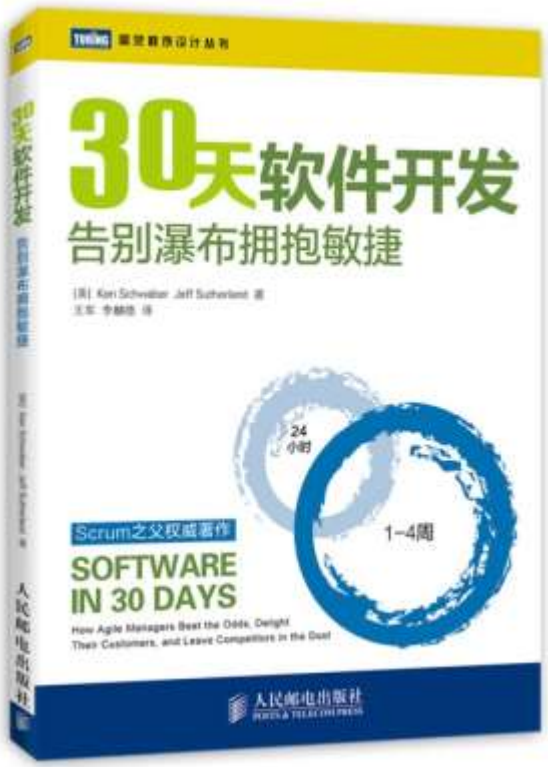


推薦參考書籍

- Sortware in 30 Days

  30 天軟體開發 – 告別瀑布擁抱敏捷

  (簡體中文)



推薦參考書籍

- 精實開發與看板方法

  李智樺 Ruddy Lee

  (正體中文)



參考資源（一定要看）

- 敏捷專案開發精神 | Ruddy Lee 分享空間





部門初期做了哪些事情

2015 資訊處某部門的年度 KPI 項目

- CI（持續整合, Continuous integration）
- RM（發行管理, Release Management）

RM（發行管理，Release Management）

當時的做法：

將產品的發行管理權限拿回來，由部門進行建置管理，

使用既有的 TFS 所提供 Release Management，

之後的目標是要把專案上版的動作由原本開發人員改

為 Tech Leader 去執行。

部門 CI 所要做的事項

CI 將會包含以下五種內容，分別為：

- **程式版本控管** – 使用 Git 及 採用 Git Flow
- **自動化測試** – 於專案開發導入單元測試
- **程式碼品質** – 程式碼規範、落實 Code Review
- **缺陷管理** - Issue Ticket
- **訊息通知** - Notification

程式版本控管

- 資訊團隊既有使用的版本控管為 TFVC

  集中式版本控管

  Team Foundation Version Control

- 實用習慣卻是沿襲以往所使用的 VSS (Visual Source Safe)

- 單一鎖定、版本混亂、開發者版控觀念不清

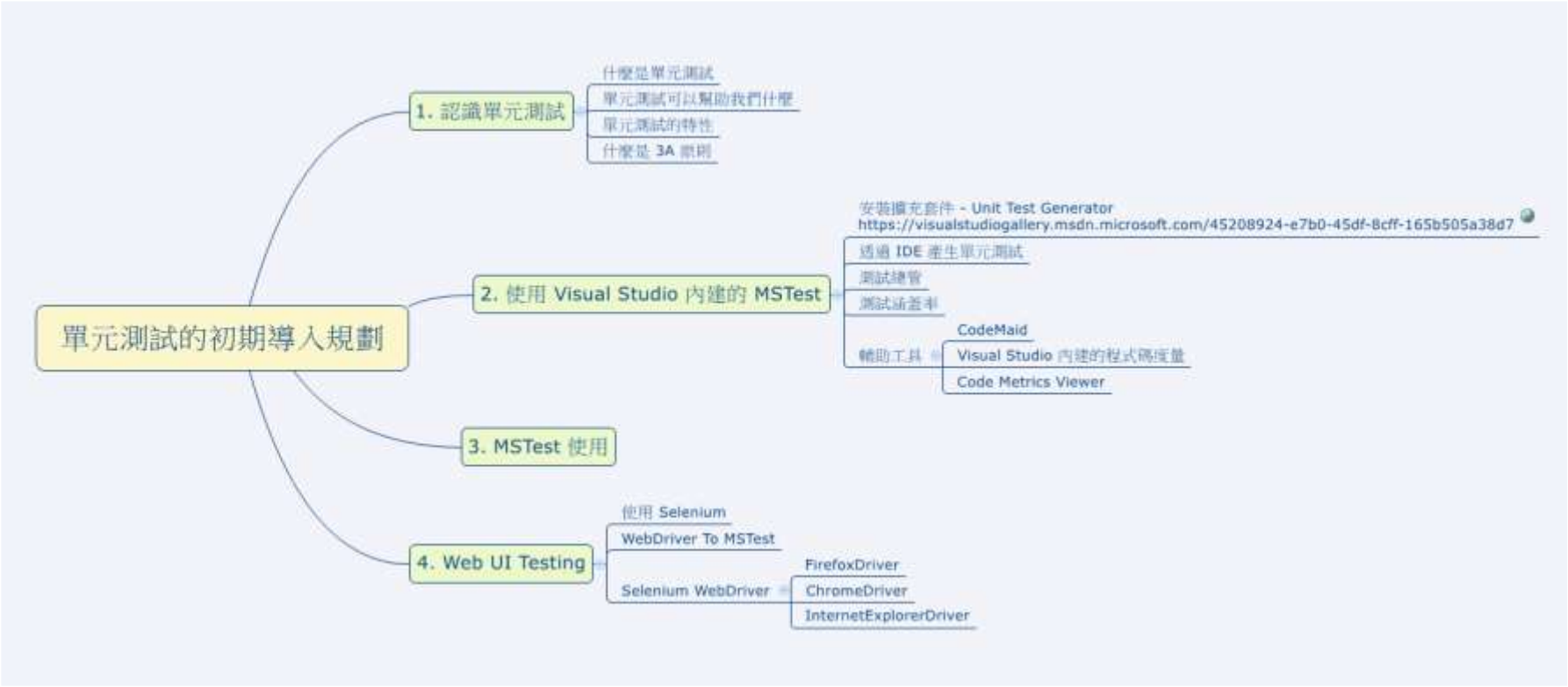
程式版本控管

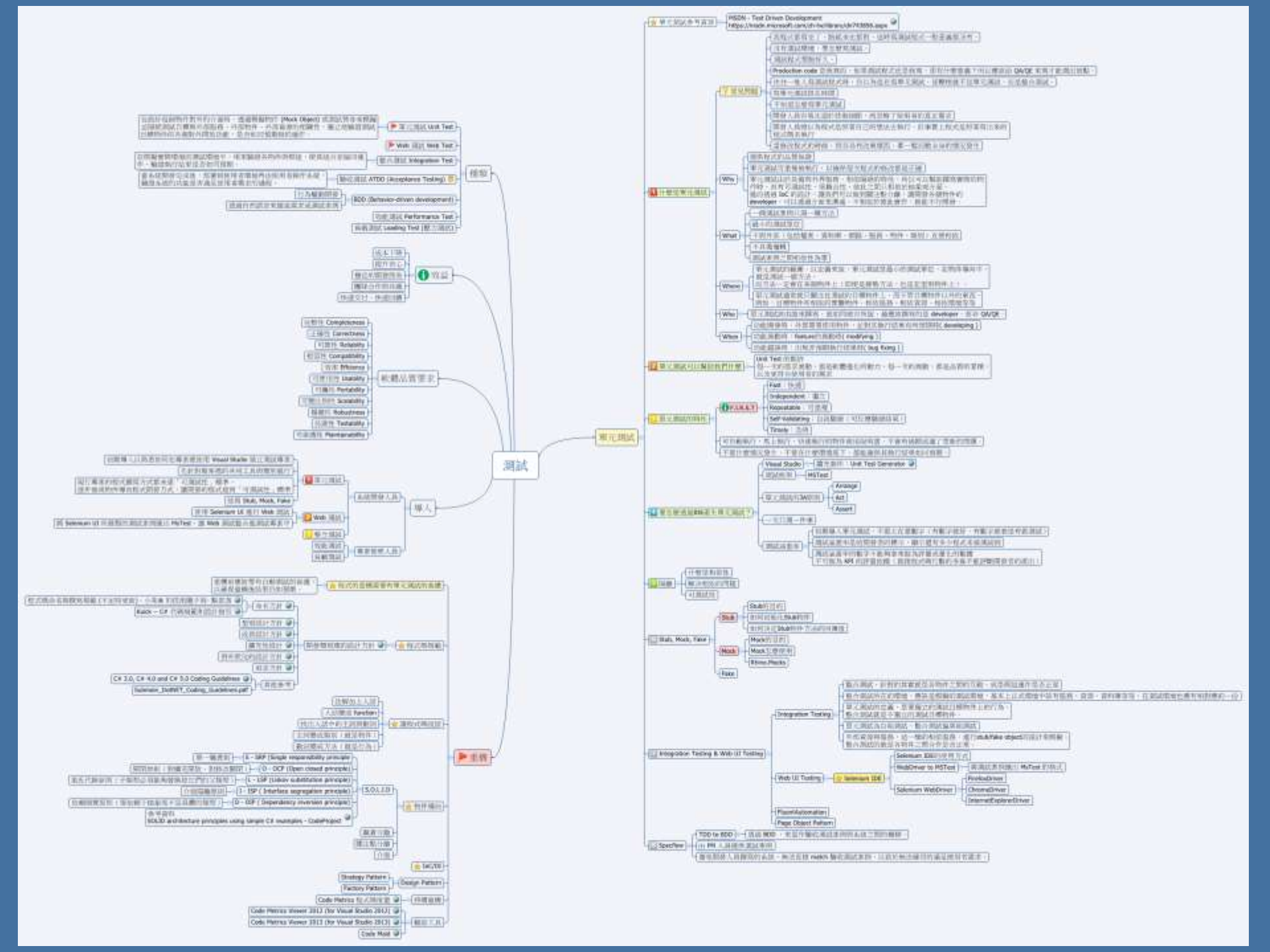
- 當時已經陸續有新專案開發使用 Git，
  但版控管控上還是離不開 TFVC, VSS 的習慣
- 採用 Git Flow 來做開發分支的管控
- 後續部分專案嘗試使用 Github Flow
- **目前狀況：目前採用 Git Flow + Github Flow 的混合體**

自動化測試

- 專案將逐步導入測試，舉凡單元測試、整合測試等等各種測試，以 Visual Studio 與 .NET Framework 所提供的測試框架(MsTest)為基礎，讓以後專案都要導入測試.
- **目前狀況：目前新的專案都已導入單元測試與整合測試，並且已在專案品質看出顯著成效**

內部訓練-單元測試第一階段導入規劃




測試
概型
單元測試 Unit Test
Web 測試 Web Test
整合測試 Integration Test
驗收測試 ATDD (Acceptance Testing)
BDD (Behavior-driven development)
功能測試 Performance Test
負載測試 Loading Test (壓力測試)
可讀
可維護
可擴展
軟體品質要求
Completeness
Correctness
Reliability
Compatibility
Efficiency
Usability
Portability
Scalability
Robustness
Testability
Maintainability
導入
單元測試
Web 測試
整合測試
系統開發人員
專案管理人員
Visual Studio
Stub, Mock, Fake
Selenium IDE
Selenium WebDriver
MsTest
重構
程式的基礎需要有單元測試的基礎
程式碼設計
Code Metrics
Design Pattern
視覺工具
Code Metrix Viewer 2013 (for Visual Studio 2012)
Code Metrix Viewer 2013 (for Visual Studio 2013)
Code Maid
單元測試
參考資源
MSDN - Test Driven Development
計劃單元測試
常見問題
Why
What
Where
Who
When
單元測試可以幫助我們什麼
Unit Test 的設計
單元測試的特性
F.I.R.S.T.
Fast
Independent
Repeatable
Self-Validating
Timely
單元測試的寫法
Visual Studio
測試框架
MSTest
單元測試的3A原則
Arrange
Act
Assert
一次只測一件事
測試命名
Stub, Mock, Fake
Stub
Mock
Fake
Integration Testing & Web UI Testing
Integration Testing
Web UI Testing
Selenium IDE
Selenium WebDriver
PageAutomation
Page Object Pattern
SpecFlow
TDD to BDD
Gherkin
Step Definitions
Unit Test
Integration Test
Selenium IDE
WebDriver to MSTest
Selenium WebDriver
FirefoxDriver
ChromeDriver
InternetExplorerDriver

訊息通知

- 程式簽入、Bug、異動或任何重大變動時進行通知
- 避免通知訊息散落各處而難以追蹤
- 訊息通知不只限於一般的 Email 傳遞，將使用 Slack 做為替代或輔助。

計畫趕不上變化…

- **計畫趕不上變化，而變化也比不上老闆的一句話**
- 因為下半年的公司重大專案，使得原計畫的部門 CI/RM 計畫的所有推動項目因此停擺
- 但是「**Git 版本控管**」「**單元測試**」已經有落實在部門的專案開發過程中

推薦參考書籍

- The Art of Unit Testing – with Example in C# Second Edition

  單元測試的藝術(第二版)

  Roy Osherove

  (簡體中文)



推薦參考書籍

- Effective Unit Testing – A guide for Java Developers

  有效的單元測試

  Lasse Koskela

  (簡體中文)



課程 – SkillTree.My

- 91 – 自動測試與 TDD 實務開發(使用C#)



https://skilltree.my/

課程 - SkillTree.My

- 陳小風 - JavaScript 實務測試新手班

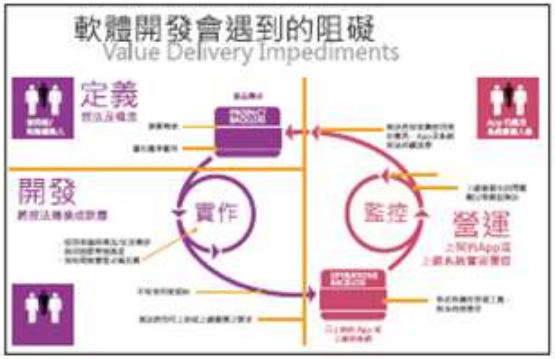




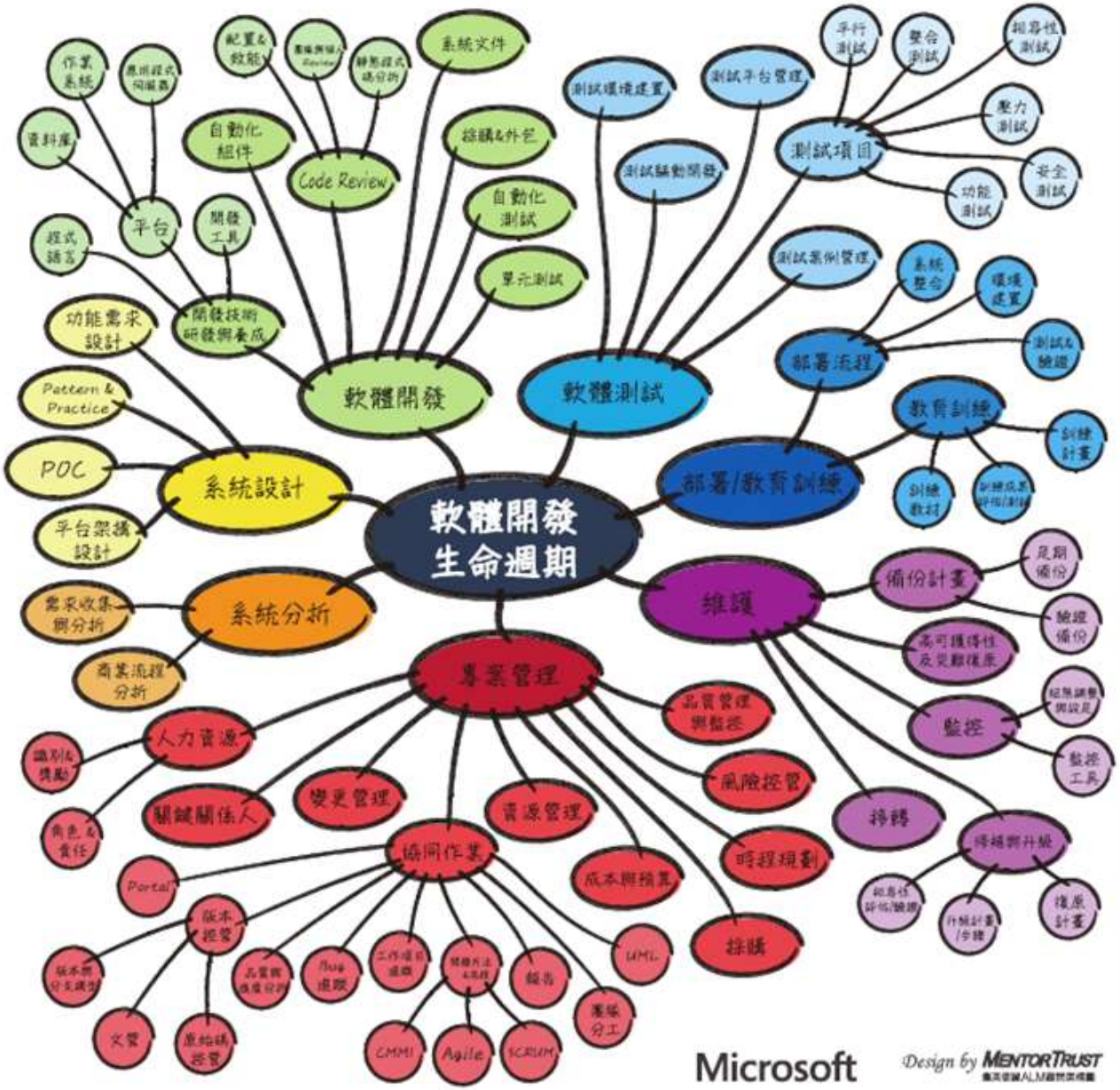
資訊開發團隊做了哪些改變

Visual Studio 2012
Ultimate









Team Project 入口 - Team Foundation Server
專案 Dashboard - 管理更加透明



Backlog Management - Team Foundation Server
待處理項目 和 進度分類



白板 (Task Board) - Team Foundation Server
每日 Review 及 工作調整



Tip：在白板上直接指定時間、成員



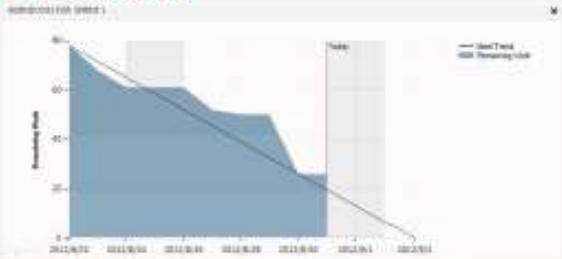
Tip：工作單內容及備註的方式



資源規劃 - Capacity Planning
思考工作的合理性 和 資源的分配



燃盡圖 - Burn down Chart
了解團隊的進度



軟體生命週期 ALM - Microsoft Solution



雲端團隊開發平台
Team Foundation Service
tfs.visualstudio.com



- 零安裝, 零維護, 立即開通
- .NET/Java / Android/iOS 支援
- MSDN 客戶免費

了解軟體生命週期管理、軟體測試、敏捷開發
請上ALM 線上學習中心 http://aka.ms/learnalm

ALM-Application Lifecycle Management

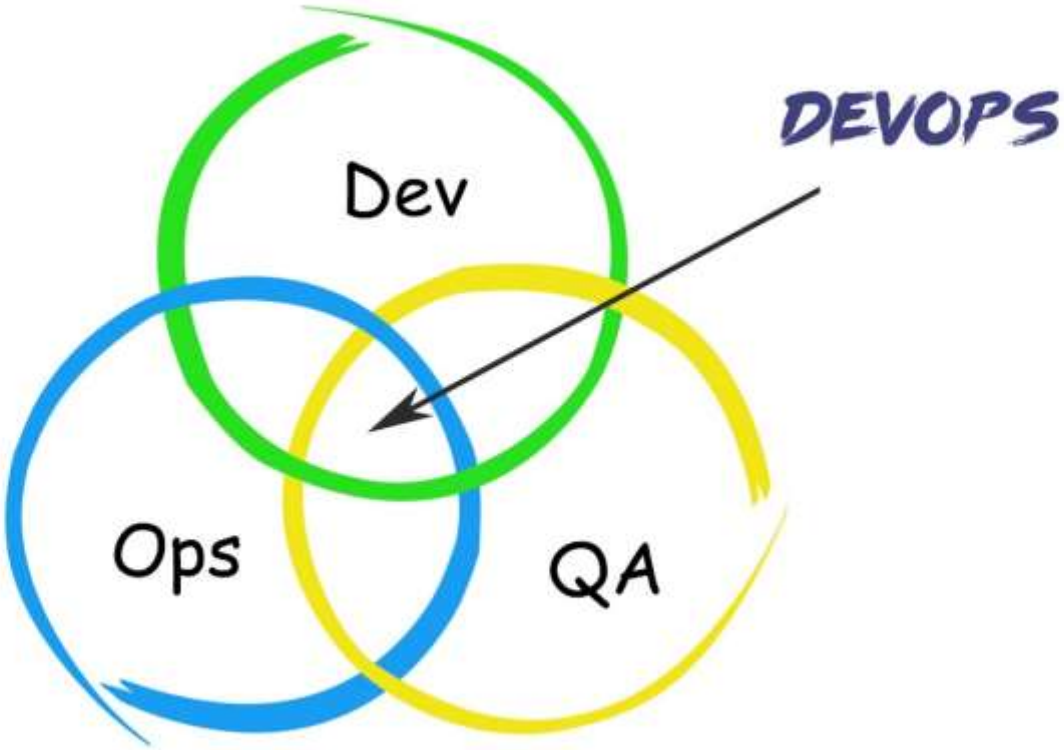
- 軟體生命週期管理 (ALM)、軟體測試、敏捷開發之應用



什麼是 DevOps

- Development + Operations
- 重視軟件開發人員和運維人員的溝通合作，通過自動化流程來使得軟件構建、測試、發布更加快捷、頻繁和可靠。
- **敏捷、透明度、自動化**

DevOps – 開發、測試、維運



DevOps - 敏捷開發

- 消除浪費
- 過程透明
- 持續交付正確可用的成品

消除浪費

七種軟體開發的浪費

Seven Wastes of Software Development

1. 部分完成的工作 (Partially Done Work)
2. 多餘的流程 (Extra Processes)
3. 多餘的功能 (Extra Features)

消除浪費

七種軟體開發的浪費

Seven Wastes of Software Developement

4. 任務切換 (Task Switching)

5. 等待 (Waiting)

6. 動作(移動) (Motion)

7. 缺陷 (Defects)

DevOps – 透明度

- 使用者需求、反饋（Bug, Feedback）
- 專案管理、專案風險、專案現況、完成率
- 系統維運狀況：

  Server, Application Monitoring
- 程式碼管控、版控記錄、工作項目狀態

DevOps – 自動化

- 自動組建
- 自動測試
- 自動發佈
- 自動部署
- 藉由一致性標準化的設定進行自動化的管理，確保工作的可重複性，大幅減少人工操作的錯誤

DevOps 基礎建設

- **程式碼管理**
- 敏捷專案規劃
- 測試用例管理
- **自動化構建**
- **持續發佈**
- **部署管理**
- 整合、效能、負載測試
- 回饋管理
- 團隊協作
- **應用系統監測**
- 測試環境管理

推動 DevOps 轉型的成功 2015 DevOps Day 團隊開發日

活動 | DevOps TW | 2015 DevOps Day 團隊開發日

推動 DevOps 轉型的成功要素

日期: 十二月 7, 2015 from 1:30下午 to 2:30下午 | b01

(1) | 1 則留言

普通：5

reddit 推文 Like 16



+ 佇列 ‹› 內嵌 格式

https://channel9.msdn.com/Events/DevOps-TW/2015-DevOps-Day/b01

推薦參考書籍

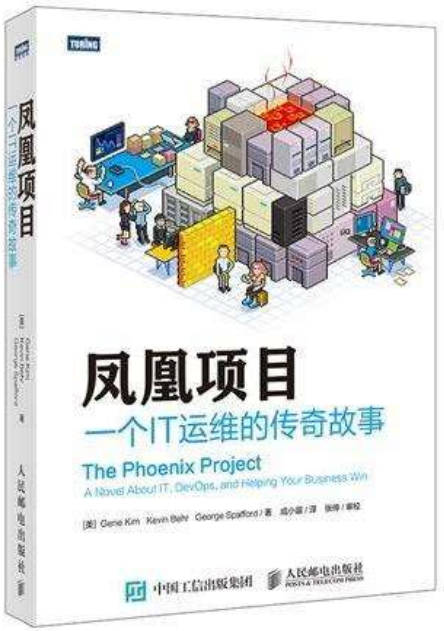
- 鳳凰計畫：一個IT計畫的傳奇故事

  (The Phoenix Project)

  (簡體中文)

- **強烈推薦**

- 作者訪談及書評



四種工作類型

- **業務項目**
- **IT維運工作**
- **變更** (前面兩種工作而引起的，並對服務產生影響)
- **計畫外的工作** (救火、收拾爛攤子、前面三種引起的)

三步工作法 The Three Ways

- 工作流
- 回饋
- 實驗與學習

- The Phoenix project 導讀 | Ruddy Lee 分享空間
- 從限制理論看 DevOps - William Yeh

團隊所面臨到的問題

- 版本控管混亂，容易造成錯誤
- 專案文件及檔案的管理
- 產品錯誤訊息不及時、Log 記錄查詢不易
- 元件及套件使用、發佈與管理的問題
- 產品上線方式紊亂

版本控管轉移到 Git

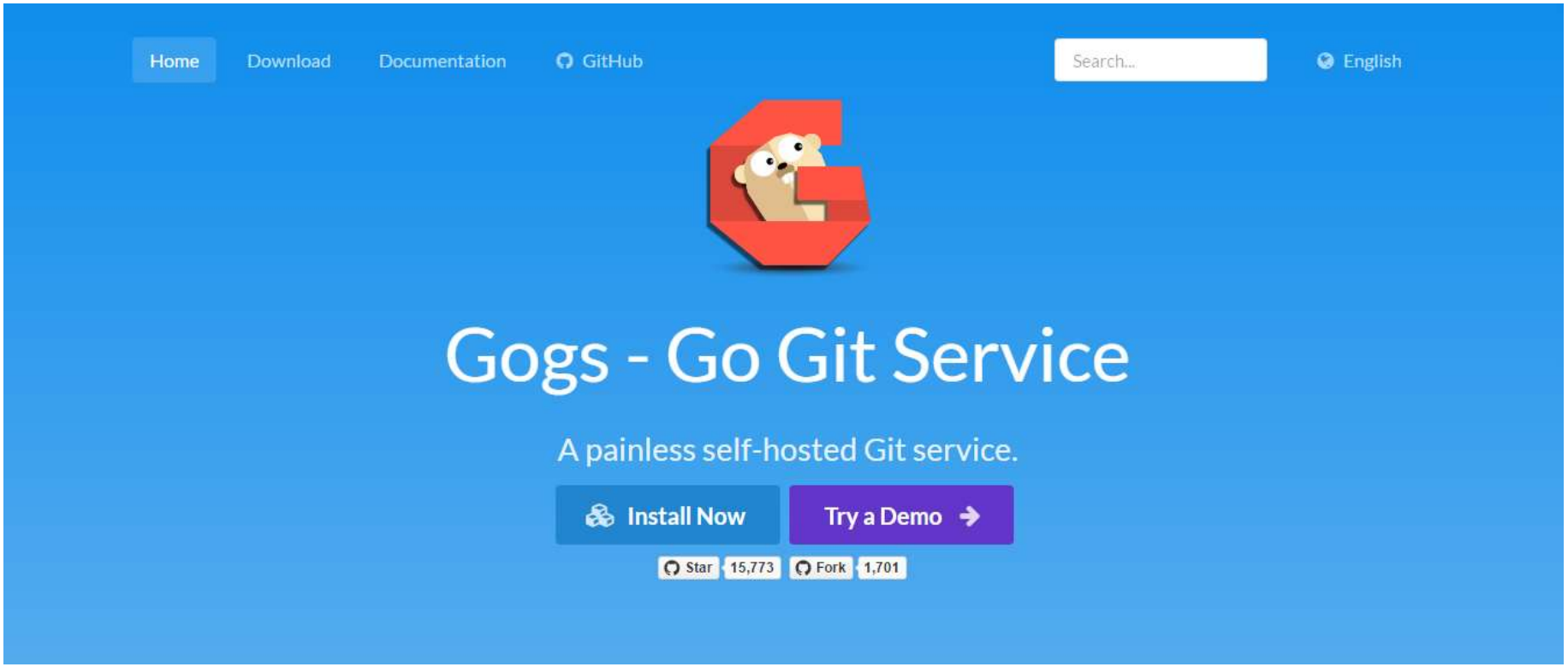
- 將舊版 TFS 內的專案遷移到新的 TFS 2015
- 逐步將專案的版控改為使用 Git
- 上線使用 master 主幹，開發用 develop 分支

  新功能的開發從 develop 分出 feature 分支

專案文件及檔案的管理

- 專案文件以及檔案的管理，以往都是使用網路磁碟機（共用資料夾）的方式，容易造成檔案版本不一致，找不到專案資料夾的問題
- 開發團隊的程式碼分享與溝通
- Lab 專案的 Source Code 也需要版控
- IT 團隊的組態設定檔案也需要版控

Gogs – Go Git Service



https://gogs.io/

為什麼需要另一套 Git Server



方便分享程式碼



參與程式碼討論

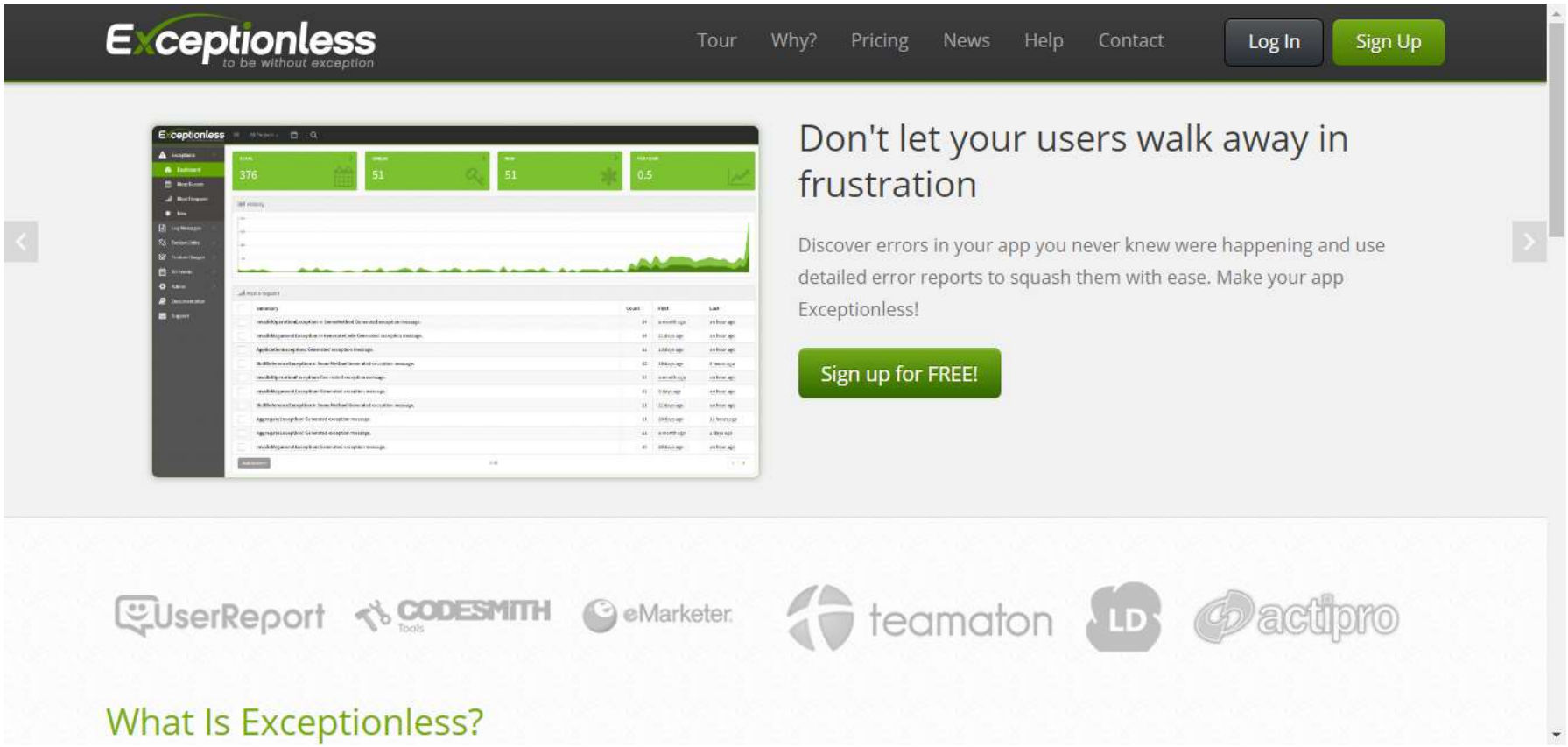
幫助檔案的版控



產品錯誤訊息不及時、Log 記錄查詢不易

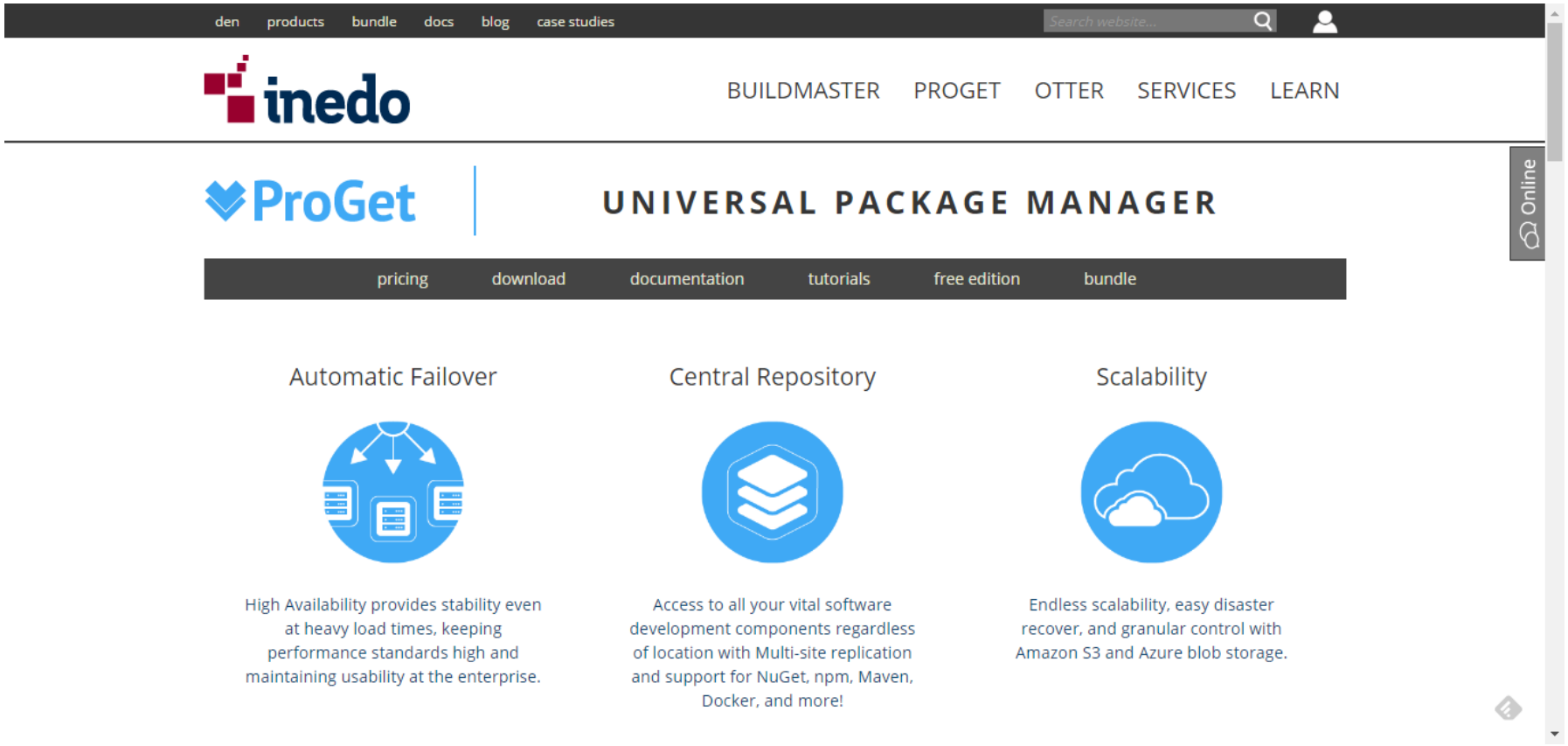
- 已經有一套運行已久的產品 exception 記錄機制，但隨著產品服務的增加，exception 產生的速度也驚人的成長，將 exception 記錄存到 DB 就時常卡住而失去及時性
- 各產品的 Log 記錄不一致，有的存檔案、有的存 DB，甚至有的根本就不管 Log

Exceptionless



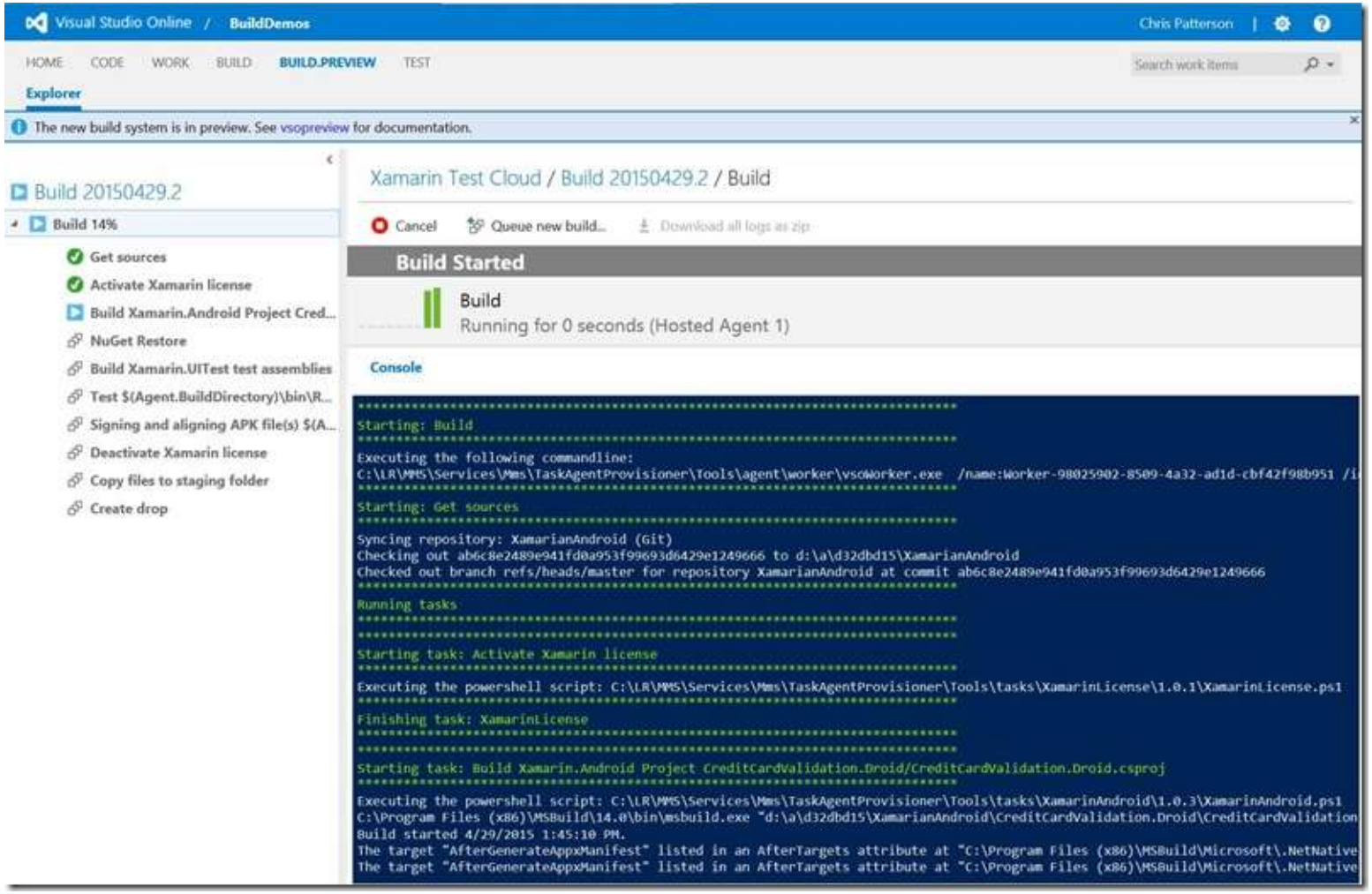
https://exceptionless.com/

元件及套件使用、發佈與管理的問題

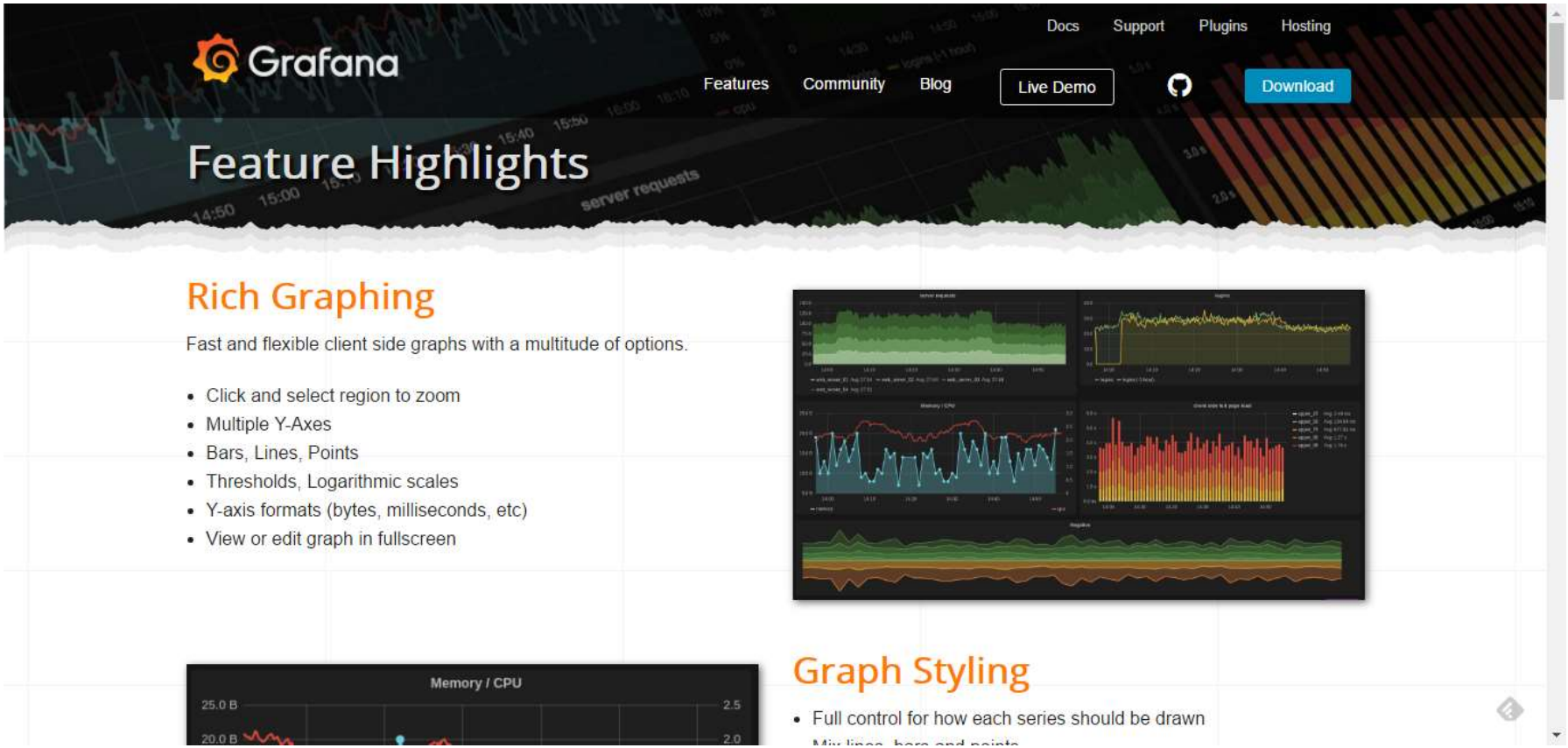


http://inedo.com/proget

使用 TFS 進行 CI/CD

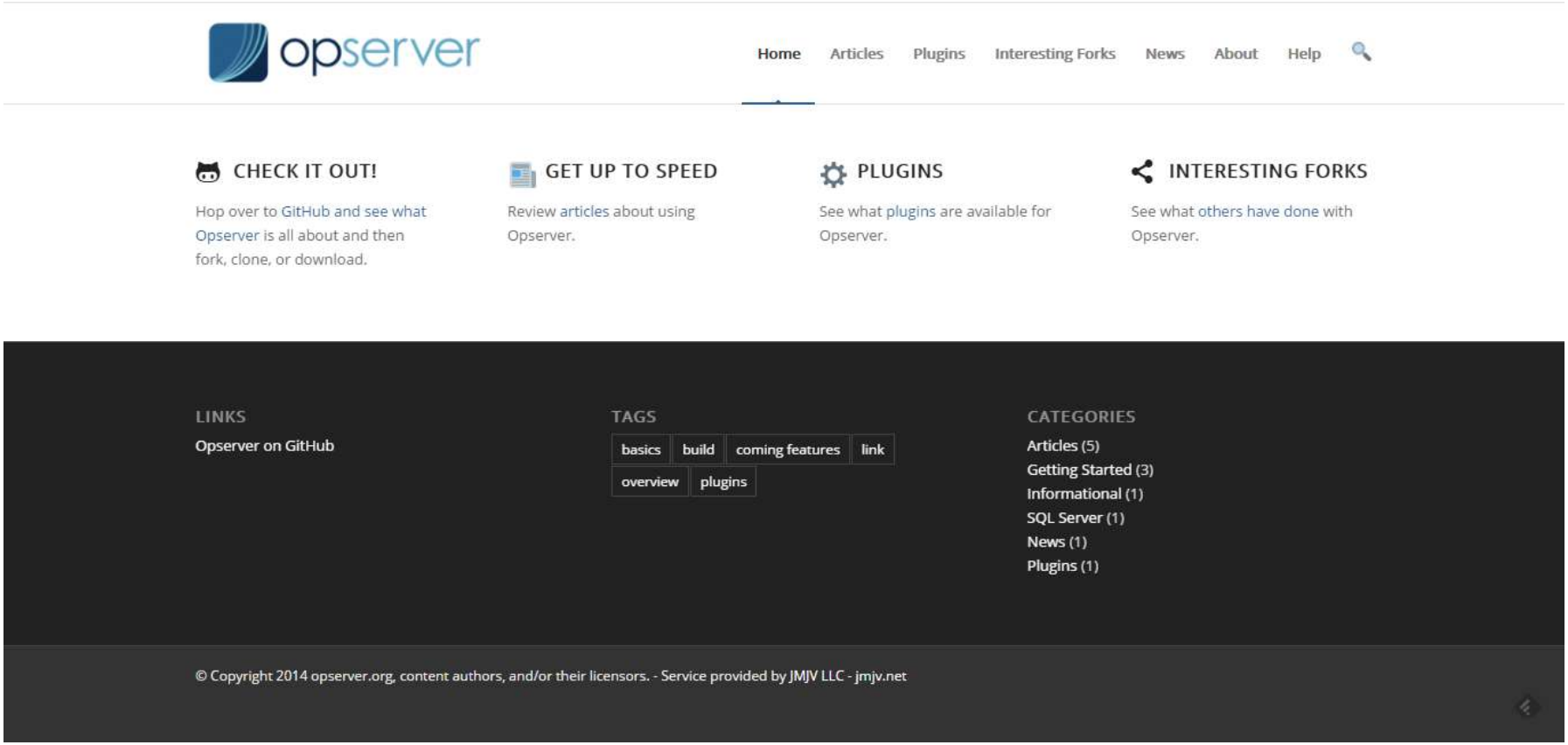


系統監控數據 - Grafana



http://grafana.org/

系統監控 - Opserver



http://opserver.org/

https://github.com/opserver/Opserver

twMVC#19 -Opserver 監控服務的解決方案



https://docs.com/is-twMVC/7231/twmvc-19-opserver-chengju

The Joel Test – 約耳測試

- 團隊有使用原始碼控制系統嗎？
- 團隊成員能用一個步驟建出所有結果嗎？
- 團隊有沒有每天都重新組建(daily builds)嗎？
- 團隊有沒有問題追蹤系統(bug tracker)？
- 團隊成員會先把問題都修好之後才寫新的程式嗎？
- 團隊有一份最新的時程表嗎？

The Joel Test – 約耳測試

- 團隊有規格嗎（團隊成員都清楚知道規格嗎）？
- 程式人員有沒有安靜的工作環境？
- 團隊成員有沒有用市面上最好的工具？
- 團隊有沒有測試人員（有沒有寫測試）？
- 有沒有在面試時要求面試對象寫程式？
- 有沒有做走廊使用性(hallway usability)測試？

The Joel Test – 約耳測試

- **邁向高品質程式碼的 12 個步驟**
- 項目有做到就得一分，全部 12 項，滿分為 12 分
- 12 分，完美
- 11 分，可以接受
- 10 分以下，需要做檢討
- 不到 5 分，~~塊陶呀~~

推薦參考書籍

- 敏捷開發專案管理與架構設計實務（電子書）
- 董大偉著
- https://goo.gl/SfBVDB



推薦參考書籍

- Visual Studio Team Foundation Server 2012: Adopting Agile Software Practices

  敏捷開發實戰 使用 Visual Studio Team Foundation Server 2012

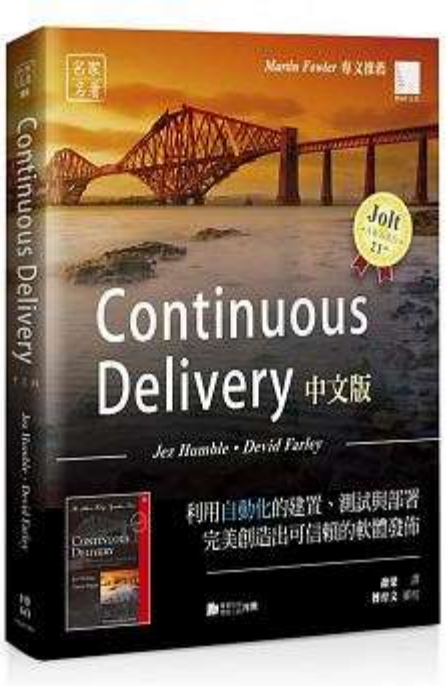
  翻譯：胡百敬、陸雲中、陳仕傑(91)



推薦參考書籍

- Continuous Delivery 中文版 - 利用自動化的建置、測試與部署，完美創造出可信賴的軟體發佈

  Jez Humble, David Farley



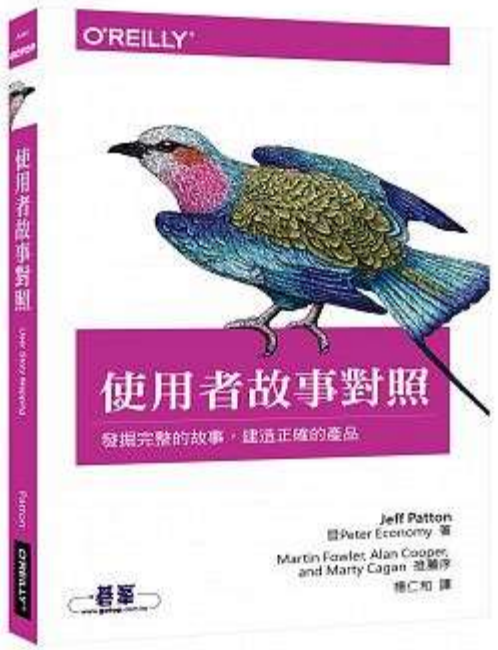
推薦參考書籍

- **使用者故事對照：User Story Mapping**

  發掘完整的故事，建造正確的產品

  Jeff Patton

  (正體中文)



推薦參考書籍

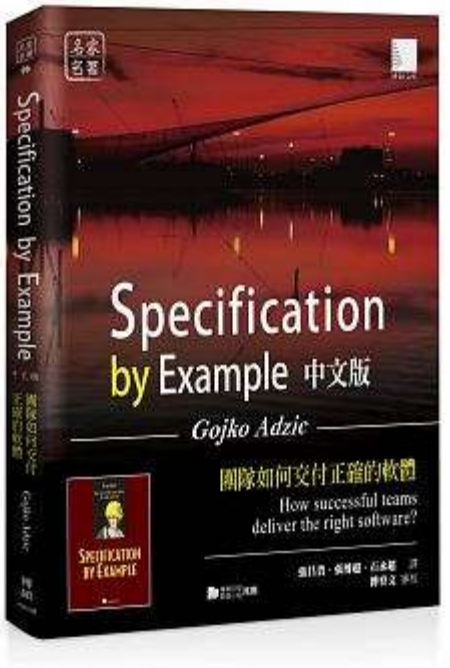
- Specification by Example 中文版

  團隊如何交付正確的軟體

  Gojko Adzic

- 「需求規格實例化」

- InfoQ - 《实例化需求》采访与书评



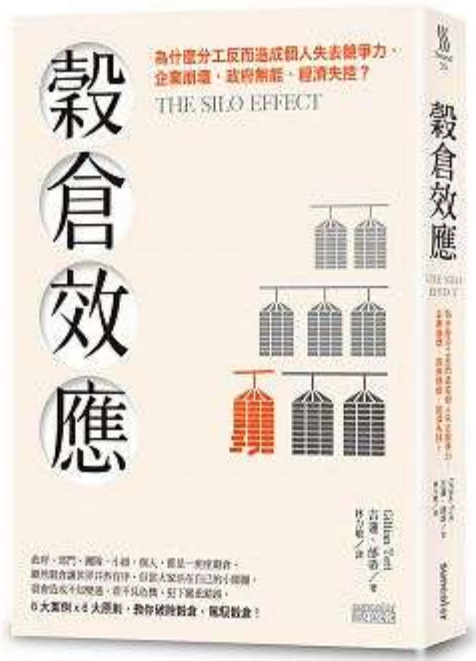
推薦參考書籍

- The Silo Effect

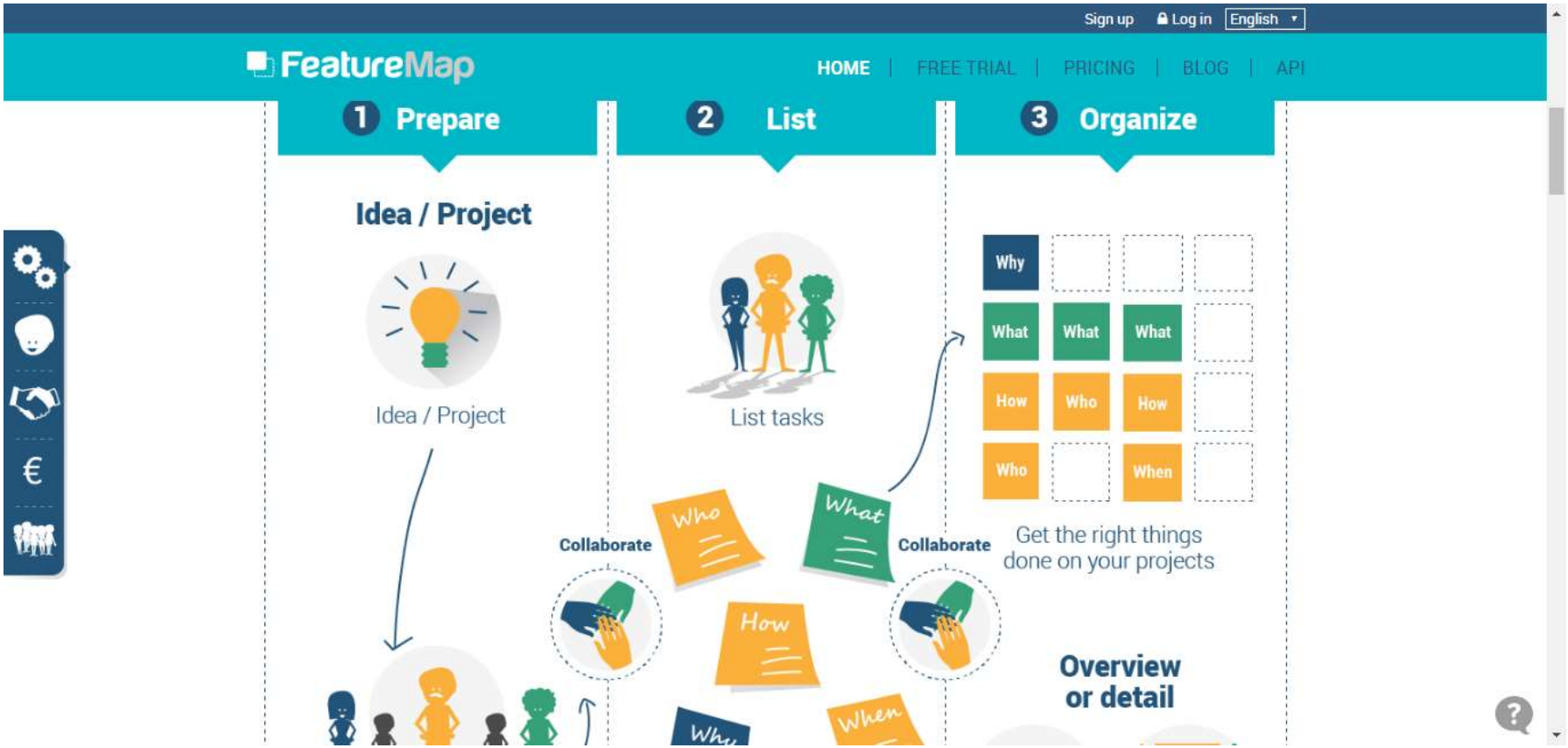
  穀倉效應：為什麼分工反而造成個人失去競爭力、企業崩壞、政府無能、經濟失控？

  Gillian Tett

  (正體中文)



開發團隊需求與工作項目管理工具



https://www.featuremap.co



開發人員因應的作為

技術導入所遭遇的質疑者類型

- 缺乏認知型
- 跟風從眾型
- 故作聰明型
- 不願再試型
- 忙於應付型
- 高高在上型
- 不講道理型
- 漠不關心型
- 頻繁抱怨型
- 一知半解型
- 閉門造車型
- 惶恐不安型

敏捷對於開發人員的影響

- 敏捷的導入，為開發團隊**創造盈餘時間**
- 開發人員自我**專業意識**的提升
- **專業技術的掌握和應用**
- 悶著頭努力做事並不能夠為團隊帶來價值
  而是你所做的事情**是否能為團隊提升價值**

知識與專案技術的分享

- 各個開發團隊完成專案開發後的分享，把遇到什麼樣的問題、使用什麼方法或技術給分享出來，讓其他不同部門的開發團隊能夠知道
- 讓各個團隊可以了解彼此在做什麼、用什麼技術
- **營造分享與相互競爭的氣氛**


I WANT YOU
FOR TECHNOLOGY SHARING
每週三技術分享日
★★★ 別再當殭屍・快來當講師 ★★★
即日起，資訊處將每週三訂定為「技術分享日」，只要
你是資訊處或設計中心的一員，皆可以自由報名成為「
明日講師」，與同仁一起分享與成長，並且資訊處提供

2016 年已經進行的教育訓練與分享

- 2016-01-21 資訊處單元測試 Lesson-01
- 2016-01-28 資訊處單元測試 Lesson-02
- 2016-04-20 資訊處單元測試 Lesson-03
- 2016-06-08 資訊處單元測試 Lesson-04

2016 年已經進行的教育訓練與分享

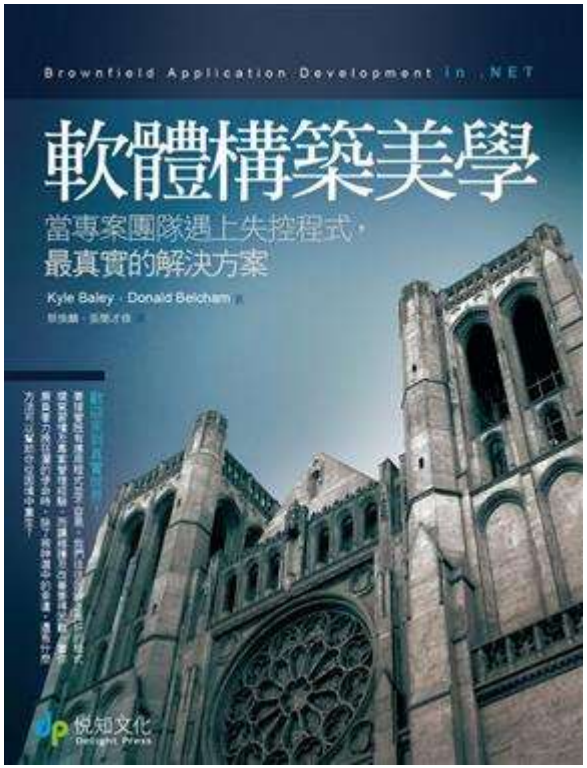
- 2016-04-28 資訊處服務架構
- 2016-05-18 Hello Git (1)
- 2016-05-25 Hello Git (2)
- 2016-06-01 Nuget Package 教學
- 2016-08-17 物件導向概念篇 (林楊森)
- 2016-08-19 物件導向SOLID原則篇 (林楊森)

2016 年已經進行的教育訓練與分享

- 2016-04-13 單元測試(09) Selenium Part.1
- 2016-05-11 單元測試(10) Selenium Part.2
- 2016-06-15 單元測試(11) Repository Part.1
- 2016-06-29 單元測試(12) Repository Part.2
- 2016-08-24 單元測試(13) Repository Part.3

推薦參考書籍

- 軟體構築美學：當專案團隊遇上失控程式，最真實的解決方案
- 已絕版（請找二手書）



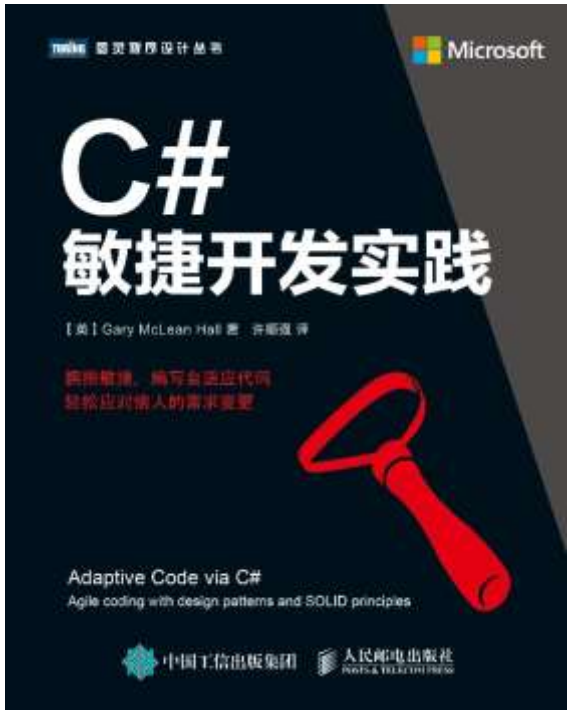
推薦參考書籍

- 敏捷軟體開發：原則、樣式及實務 (Agile Software Development: Principles, Patterns, and Practices)



推薦參考書籍

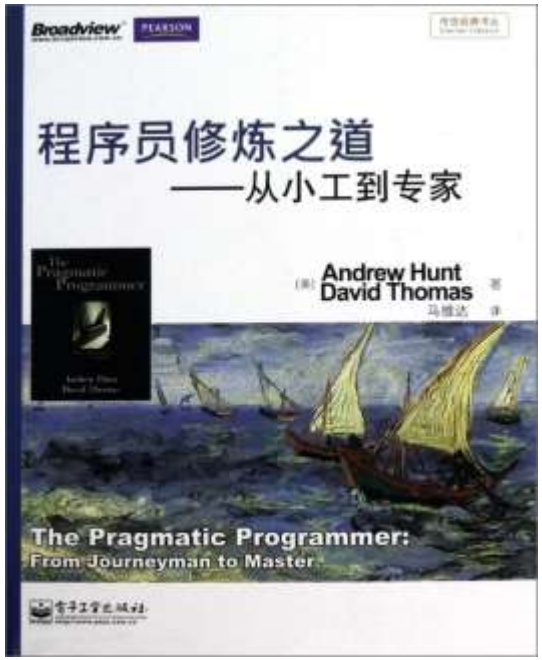
- C# 敏捷開發實踐 - Adaptive Code via C# Agile coding with design patterns and SOLID principles (簡體中文)



推薦參考書籍

- 程序員修練之道 – 從小工到專家

  The Pragmatic Programmer: From Journeyman to Master

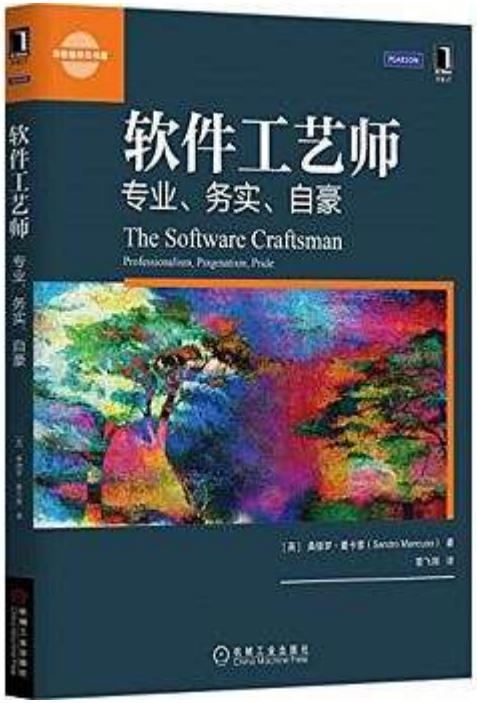


推薦參考書籍

- 軟件工藝師 - 專業，實務，自豪

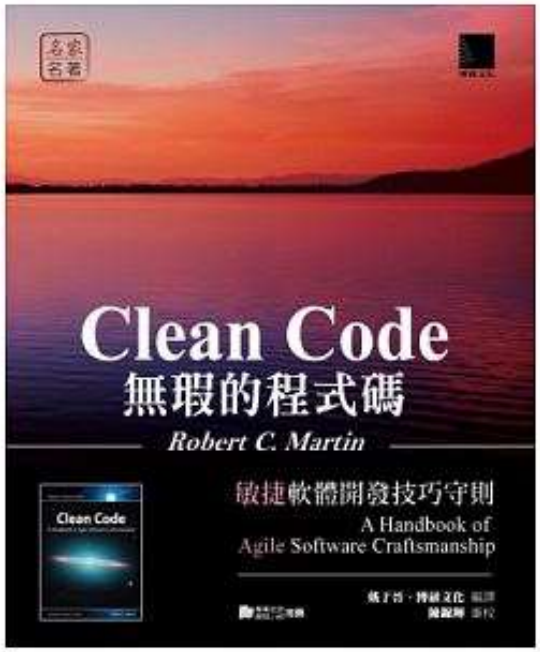
  The Software Craftsman: Professionalism, Pragmatism, Pride

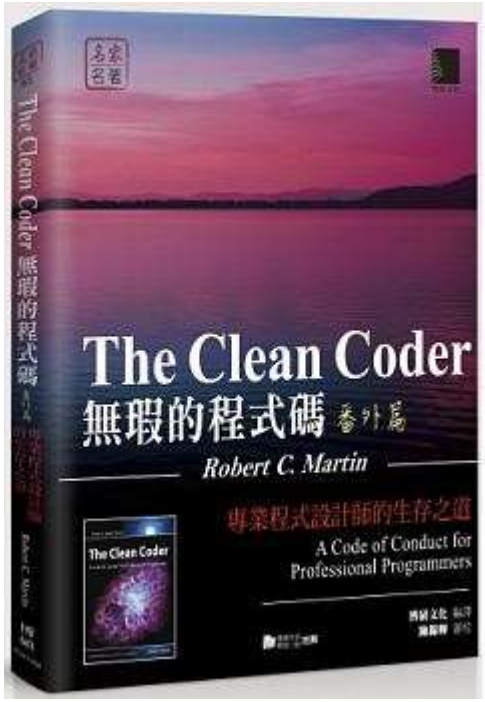
  (簡體中文)



推薦參考書籍

- 無瑕的程式碼：敏捷軟體開發技巧守則
- 無瑕的程式碼 番外篇：專業程式設計師的生存之道





敏捷開發

- **不是快速開發的方法，是用來解決複雜問題的方法**
- 務實，重要的需求先做
- 快速反應需求變化
- 漸進式的開發，持續交付
- 團隊自我管理

敏捷開發之路(未完成)

- **敏捷之路沒有所謂的完成，持續檢討、改進與修正**
- 主動發掘問題、重視回饋的訊息，持續改善
- 以引導的方式進行（必要時還是要棍子）
- 資訊團隊仍在摸索中，但已經有共同的願景與目標
- 公司文化的改變仍然持續努力中

One more thing…

我們在徵人!

永慶房產集團-資訊處:

- 系統架構師
- 資深軟體開發工程師
- 軟體開發工程師

.NET Framework C# ASP.NET MVC / Web API
Linux Docker Elasticsearch(ELK) …

史上最強的注音輸入法



千呼萬喚

MAC 平台上，最聰明的注音輸入法
現在立即前往購買 ！

tw MVC
樂在Coding 愛上MVC

完全支援

請放心在 Windows 8 上使用 新自然輸入法 10 系統升級免煩惱，平板模式也沒問題！

Blog 是記錄知識的最佳平台



感謝 Jetbrains 贊助贈品



ReSharper

https://www.jetbrains.com/resharper/

感謝 OzCode 贊助贈品



http://www.oz-code.com/

感謝 ALIVE 贊助贈品



https://comealive.io/

業界師資、實戰教學



http://skilltree.my

謝謝各位

http://mvc.tw

- 本投影片所包含的商標與文字皆屬原著作者所有。
- 本投影片使用的圖片皆從網路搜尋。
- 本著作係採用姓名標示-非商業性-相同方式分享 3.0 台灣授權。閱讀本授權條款，請到 http://creativecommons.org/licenses/by-nc-sa/3.0/tw/，或寫信至Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.