

Towards scrum of scrums

Observation & Retrospective

遊戲橘子 行動雲端創新處

軟體架構師 杜秉穎

主辦單位：AgileCommunity.tw

協辦單位：

贊助單位： 新加坡商 鈦坦科技
www.titansoft.com.sg

 TREND
MICRO

 國際厚生數位科技股份有限公司
OmniHealth Group, Inc.

關於我

- 臺北科技大學資訊工程所博士
 - 法定專長：視覺化程式語言
 - 軟體工程
- 國小玩了Doom後就想寫遊戲
- 經歷
 - 叻揚資訊 資深軟體分析師
 - 遊戲橘子 資深軟體工程師 -> 軟體架構師

Outline

- 背景與觀察到的問題
- 如何分組
- 分組後進行的情況
- Summary

團隊類型

新創團隊

接各式各樣的外包



幫公司
開發產品

以上皆是
或皆不是

團隊規模

$m < 5$

$5 < m < 10$

?

$10 < m < 50$

$m > 50$

開發方法或流程

Waterfall

Doing
Agile



Being
Agile

自以為
Agile

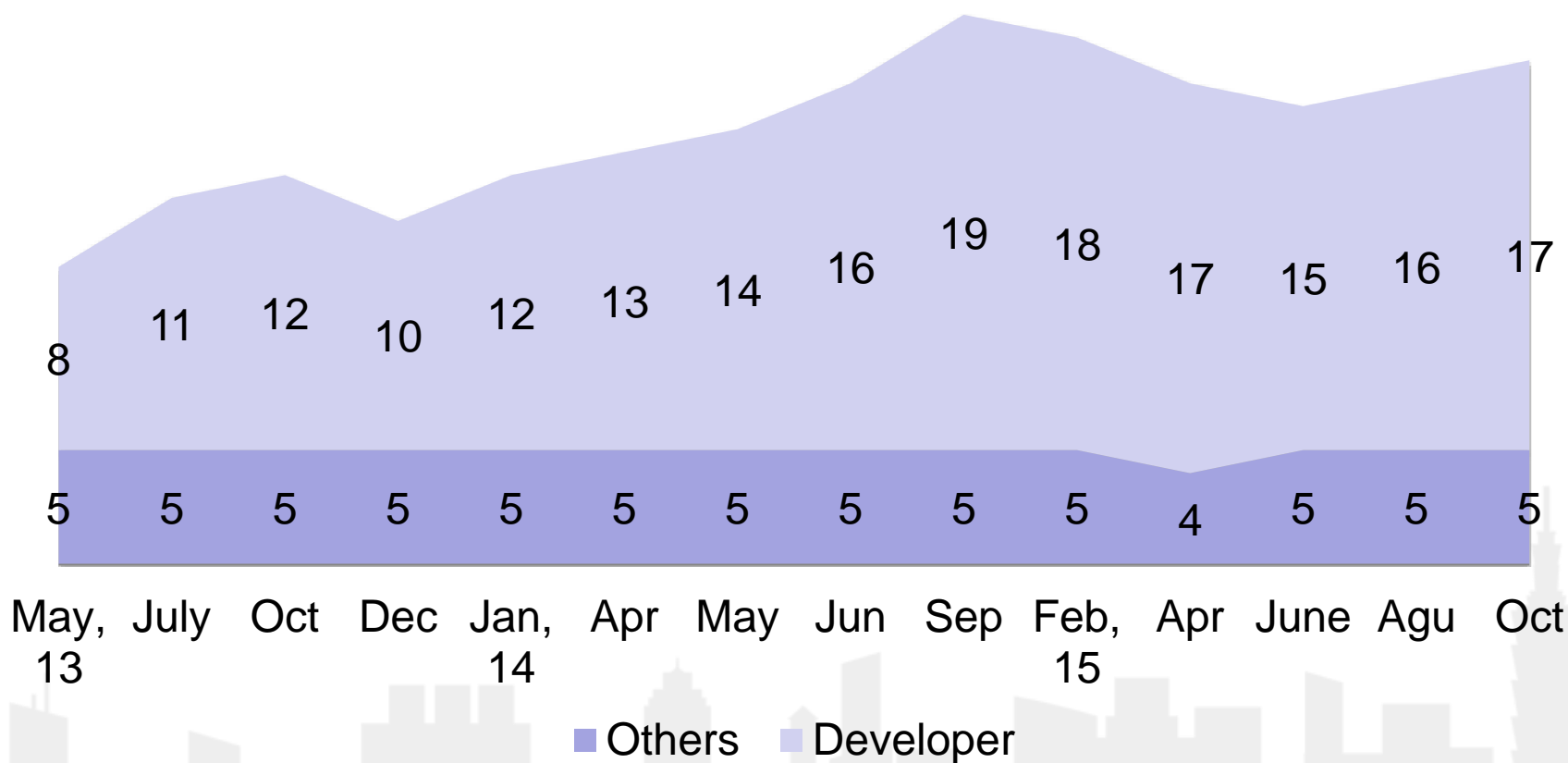
關於Project GATE團隊

- 行動裝置上的應用程式專案
 - 跨平台、使用雲服務、提供社群平台服務、導入應用程式遊戲化(gamification)
 - 一個預計自集團內部切分出的新創團隊(?)
 - 專案執行面及技術選擇上有相當的自主權(?)



2015 11/28 – 11/29
華山1914文化創意產業園區

持續成長的團隊成員



團隊好像很Agile啊~

Project GATE

Unit testing with high coverage	✓
Automatic acceptance testing	✓
Continuous integration	✓
Automatic deployment	✓
Scrum	✓
Continuous improvement	✓
Eating your own dog food	✓

還是隱隱約約感到什麼不對勁

- 會議很長、沒效率、難達成共識
 - 激烈地討論PO提出的設計
 - 這設計我本來就不喜歡...
 - 挑好做的設計
- 對sprint failed好像沒感覺又好像很害怕

如何分組

PO的定位

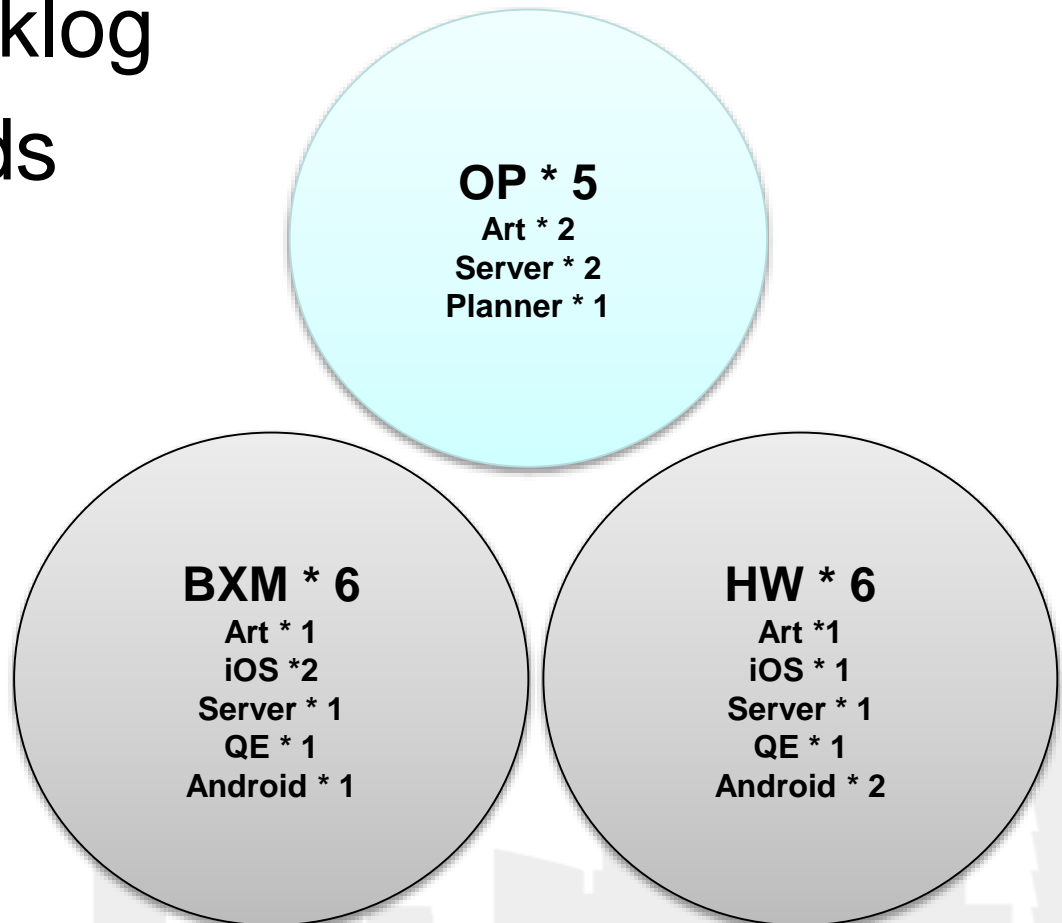
- 為了讓成員對產品更有認同感與責任感
 - PO不再提設計
 - PO -> Problem owner
 - 只定義問題
 - 由團隊提供設計

分組

- 在Sprint 4.7的自省會議結束後，由PO提出分組的進行
 - 讓團隊討論是以component teams或**feature teams**的方式分組
 - 先隨機分組
 - 讓團隊自行決定是否互換
 - 討論怎麼進行refinement meeting
 - Dice game

三個scrum teams

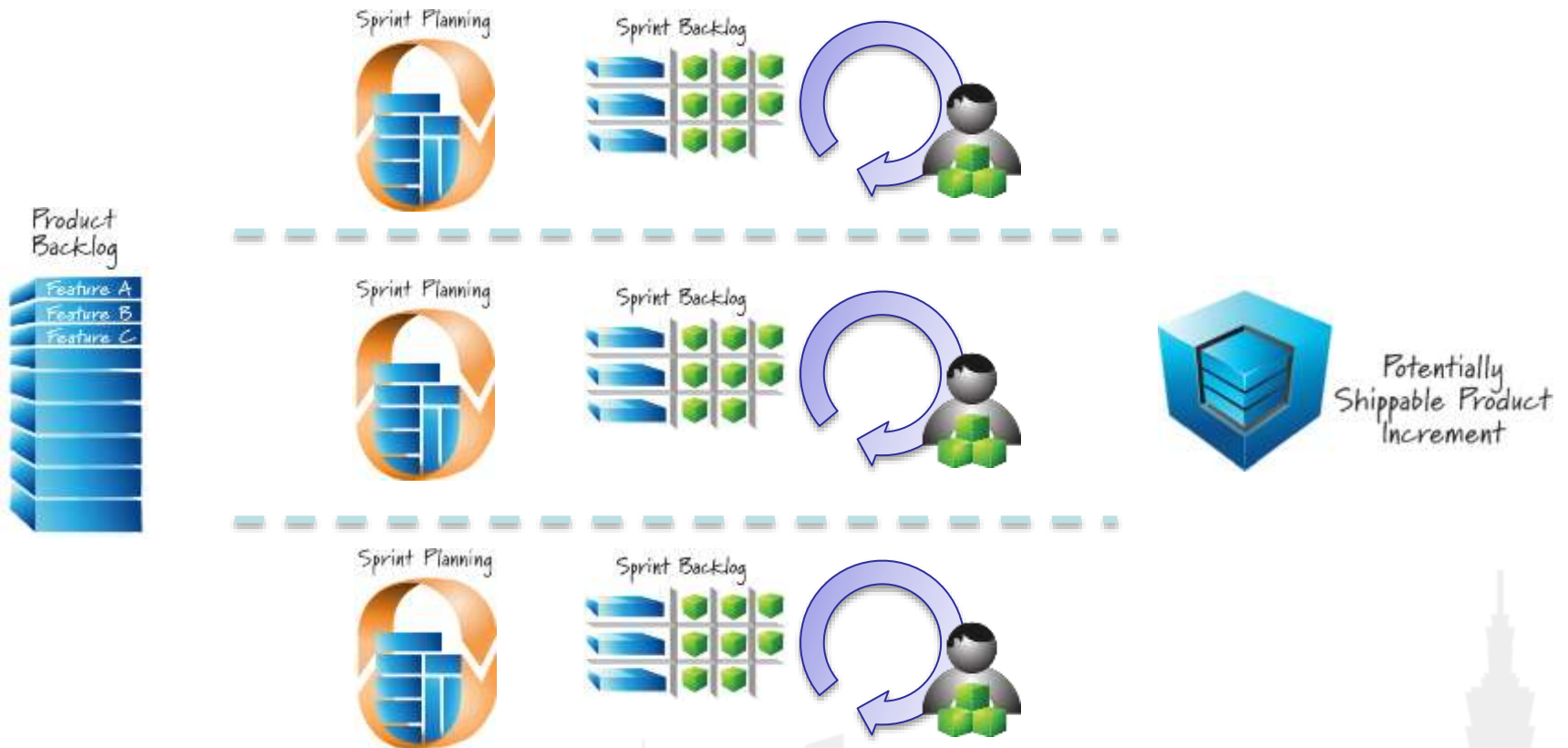
- 1 Product backlog
- 3 scrum boards



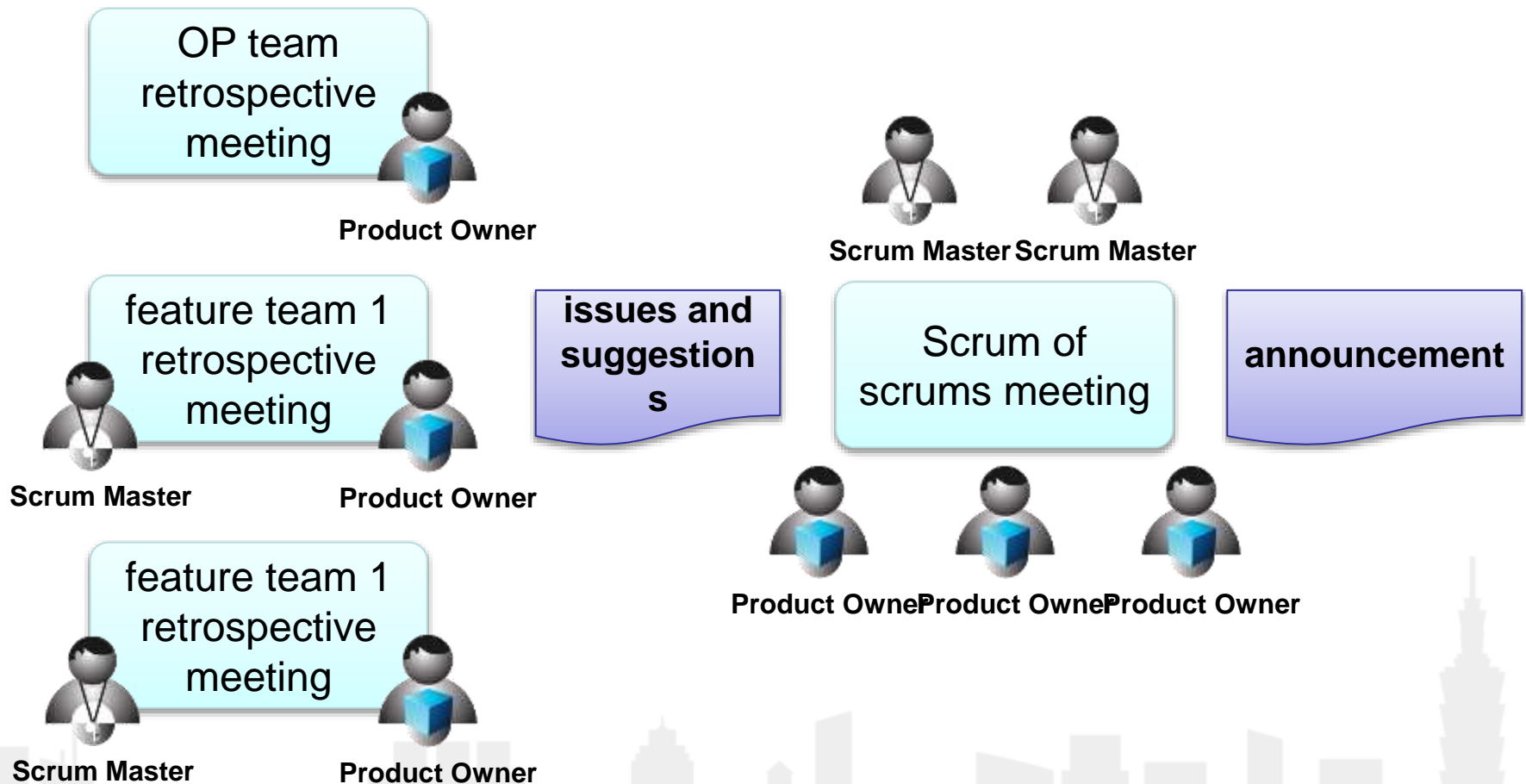
Sprint週期的改變

	Mon	Tue	Wed	Thu	Fri
Week 1	Planning				Refinement
	Planning				Refinement
Week 2				Tuning	Review
			Polish	Tuning	Retrospective
	Mon	Tue	Wed	Thu	Fri
Week 1			Planning	Refinement	
			Planning	Refinement	
Week 2					
					Polish
Week 3	Tuning	Review			
	Tuning	Retrospective			

Sprint的進行



自省會議的方式



Refinement meeting Version 1

- PO的期望
 - 1 feature team refine 1 story (feature)
- Team的決定
 - 2 feature teams refine 2 stories **together**

分組後進行的情況

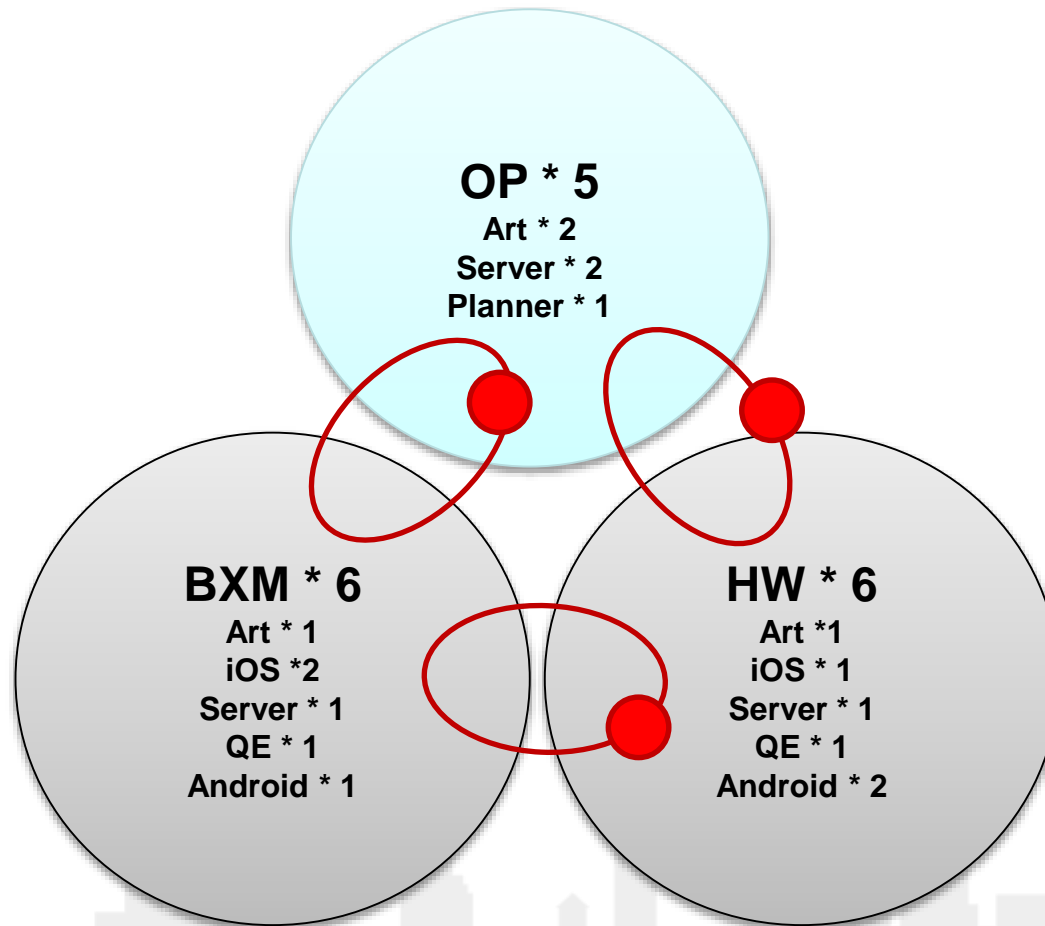
分組後第一次refinement差點看日出



Sprint 4.8 自省與觀察

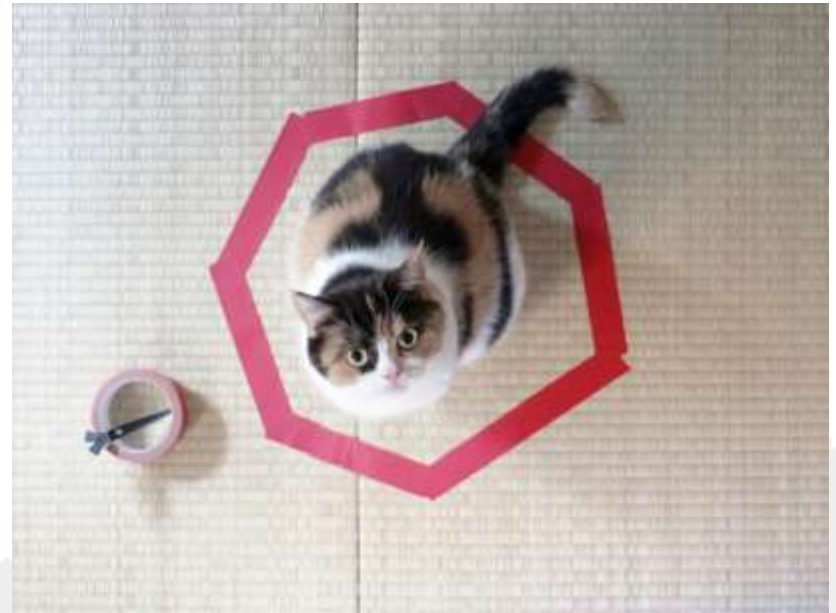
- 分組太過草率，時機不對
 - 沒有自主權，設計無法單一feature team做決定
 - 資源不均
 - Cross functional team?
- Refinement meeting時間根本沒縮短
 - 需求不清楚？
 - 一次開完？
 - 效率不佳？

跨組支援

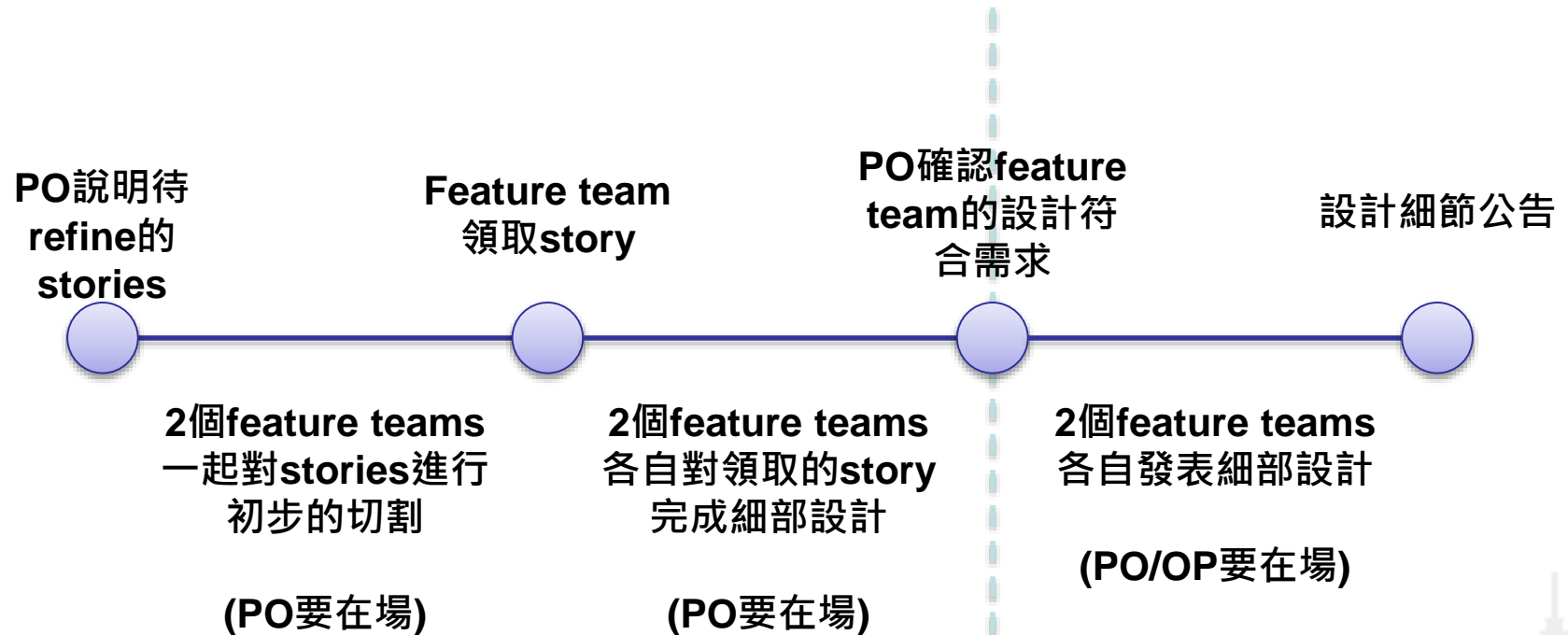


Sprint 4.9 自省與觀察

- Retrospective meeting原本大家可以一起討論出一個結果，但現在討論完還要進scrum of scrums討論，而且還不一定有結果
- 跨team的溝通不良
- 跨組支援
 - 該怎麼plan？



Refinement meeting Version 2



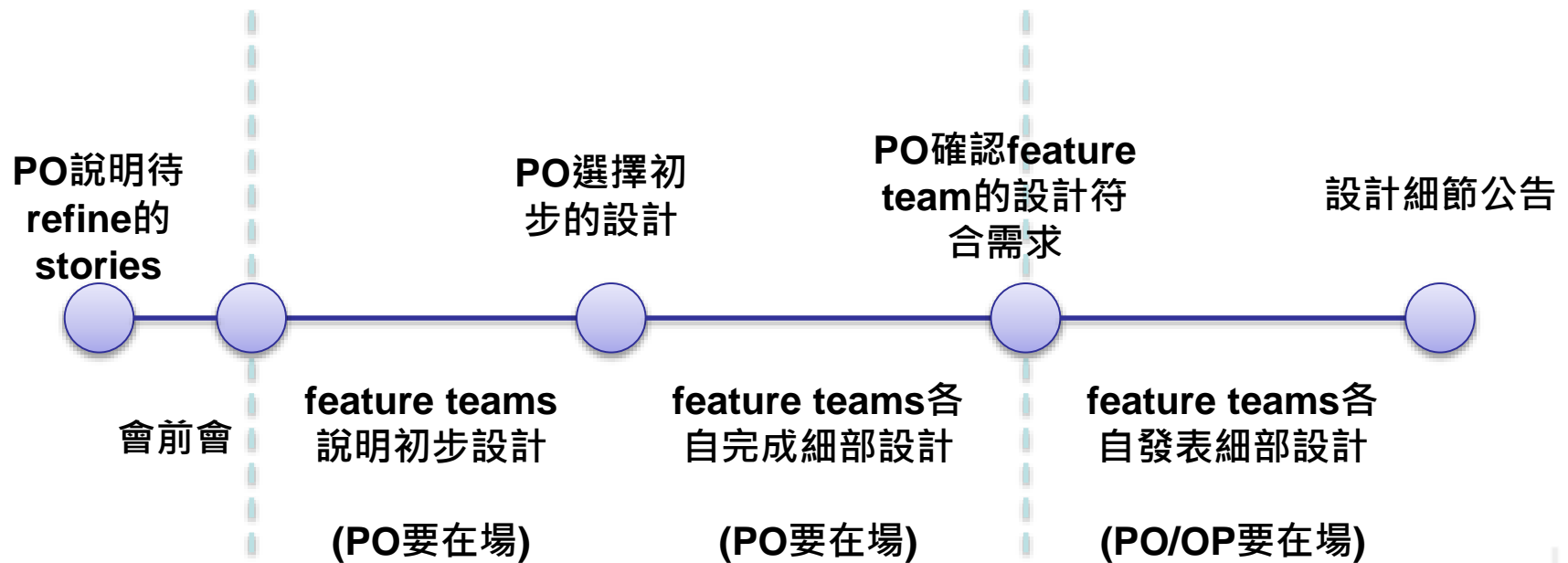
Sprint 4.10 自省與觀察 (1/2)

- 兩個feature teams對於refinement meeting 流程的認知完全不同
 - 兩個feature teams都開了會前會
 - 但提出的設計和story切割的細膩度完全不同

Sprint 4.10 自省與觀察 (2/2)

- Story (verification) failed 怎麼辦？
 - PO的說法
 - Story描述上有但沒實作
 - 部分細節(動畫/UI細節)在iOS和Android不一致
 - Story描述上沒有，但有(很明顯的)Bug
 - Team的說法
 - Planning時要plan得更細
 - 估時可能要保守一點
 - 盡早和PO驗證

Refinement meeting Version 3



Sprint 5.1 自省的回饋

- 開發中的設計調整(UI/UX)該如何同步？
 - feature teams <-> PO <-> OP
- 人力資源嚴重不均
 - 原本承諾要支援B，但A組自己都做不完了，結果AB兩組都沒做完，引起誤會
 - 說好的Android dojo呢？

Sprint 5.2 自省與觀察

- Android dojo好棒棒
 - 一個user story在dojo中完成
- Refinement meeting還是很長很累Orz
 - PO的角色到底是什麼？對設計打槍？
- 中斷
 - 在sprint中加新stories
 - 新stories的時數要plan嗎？

Sprint 5.3 自省與觀察

- 漫長的建置時間
 - 大概需要30分鐘建置出可布署的程式
 - 不含程式碼，要建置超過16k個資源檔
 - iOS/Android/Server
 - 分類建置布署
- Polish的重點到底是體驗還是驗證stories
 - QE希望大家幫忙驗證stories

Sprint 5.4 自省與觀察

- 為了產品好，還是要polish
 - 手動測試的時間點？
 - iOS開發比較快，但不代表iOS一定是對的，需求還是來自story本身的敘述
- 建議改採用branch-based的方式開發
 - 企畫用的資源原始檔(Excel)怎麼辦？
 - Build pipeline的調整
- 不屬於feature team的stories不應放在feature team的scrum board中

Sprint 5.5 自省與觀察

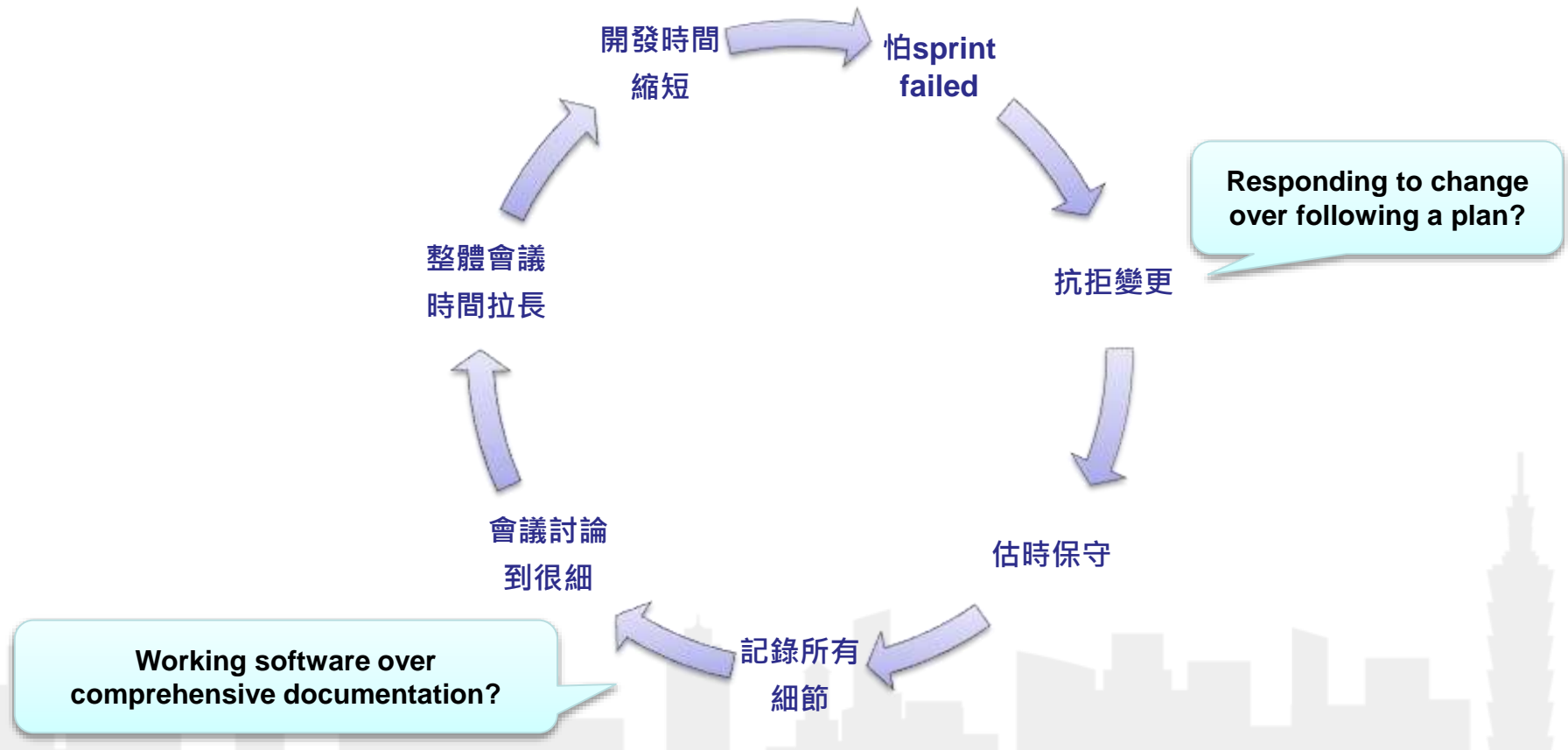
- Planning meeting的時間是否不能再縮短？
 - Task一定會切，但若時間不夠，就不估時間
 - Scrum board只秀story burn down
- 這2個sprints太趕，未能充足的Refinement，Story有許多未釐清的細節，如何認定Story完成(ready to plan)需要一個規則

Sprint 5.6 自省與觀察

- 自發地pair programming學習iOS
- Android很容易crash
 - 該refactor就做
 - 增加容錯機制
- Story應再切小一點
 - 單一Story，story point原則上不超過5點，每個sprint大約能完成20點
- 跨組支援到底該怎麼plan？

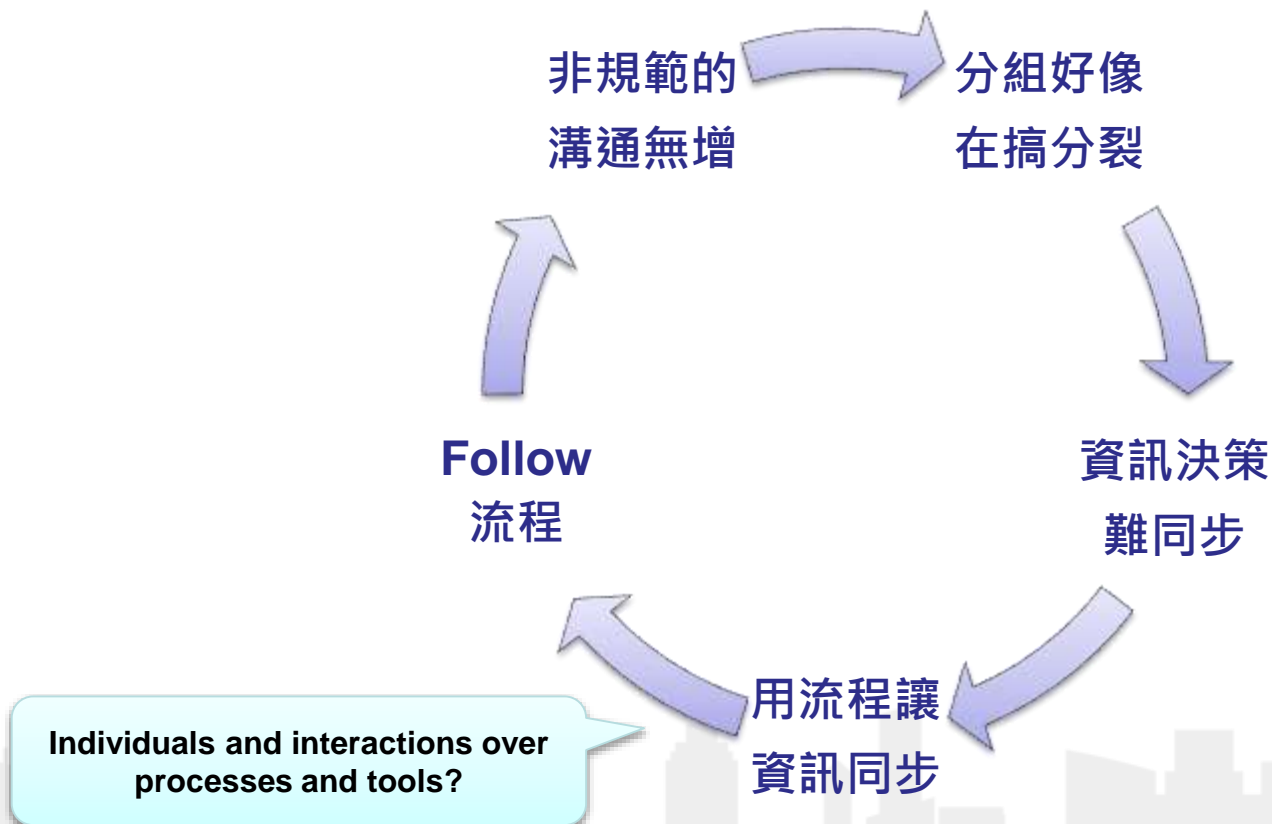
核心的問題 (1/3)

- Doing agile ≠ Being agile



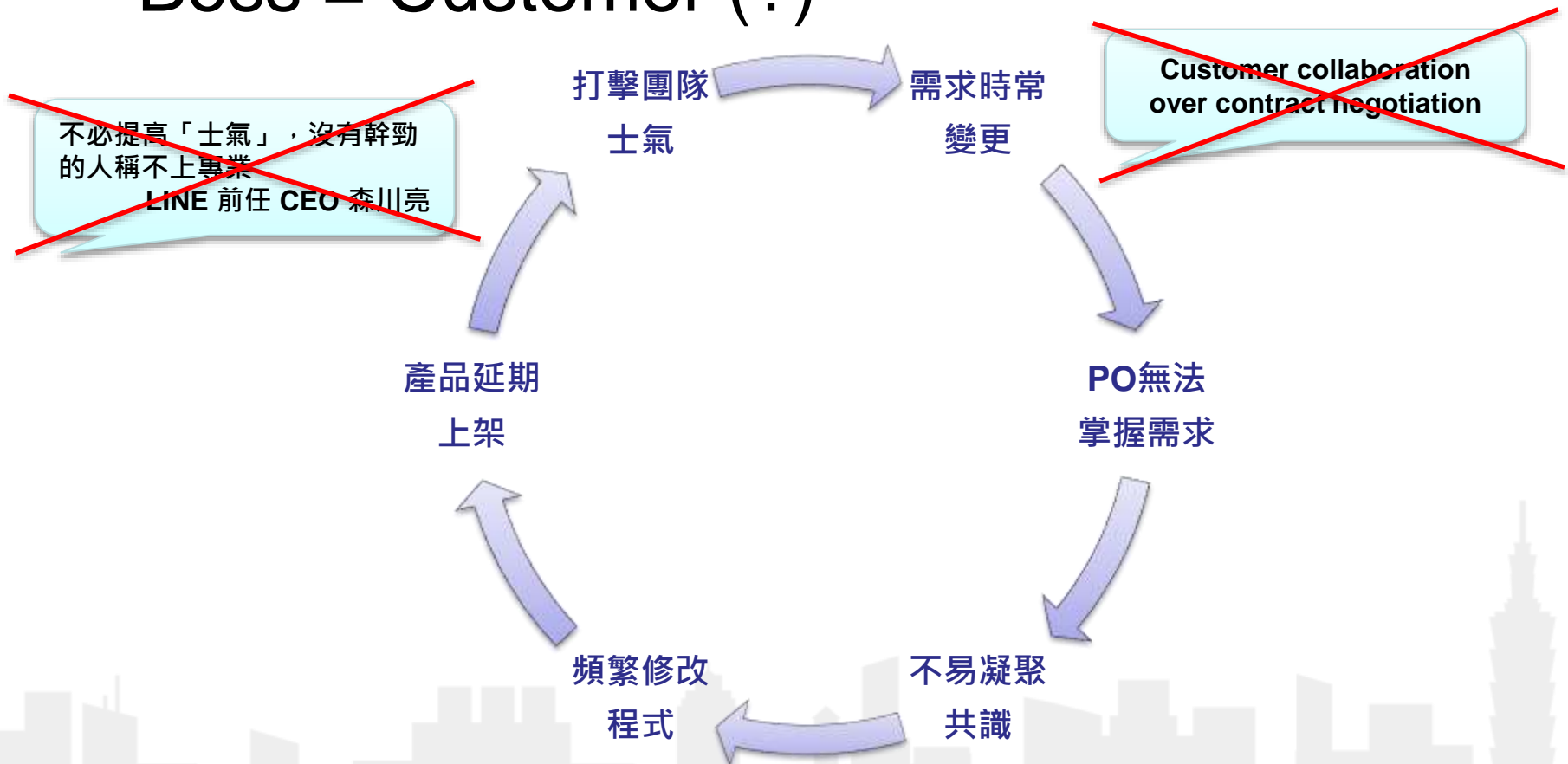
核心的問題 (2/3)

- Synchronization \neq Communication



核心的問題 (3/3)

- Boss = Customer (?)



SUMMARY

分組能做什麼

- 減少大鍋飯的心態
 - 暴露出問題
 - 自省的議題較過去辛辣 (敢說、願意說)
- 讓團隊意識到cross functional的重要性
- 有競爭意識
 - 在意自己團隊的設計是否被採用
 - 在意自己團隊的burndown chart

分組不能做什麼

- 建立對需求的共識
- 加速會議的進行
- 自發地讓自己變cross functional
- 讓團隊更Agile (?)

未來

- 雖然分組沒能真正解決當初的問題，但也不用悲觀，在scrum的框架中，團隊還是能以自己的步調調整跟改進
- 整個團隊都須對Agile有更進一步的了解
 - PO 頻繁地協調客戶(Boss)與團隊有一致的目標
 - Scrum team 用高品質的產品贏得信任
 - Scrum master 讓團隊能更agile