

DevOpsDays

Taipei 2021

過來人經驗 - 在企業中推行 DevOps 前該具備的認知與工具箱

Tim Wang
2021/11/24

Tim Wang

網通產業



Agile community / DevOps Taiwan 社群講者

開發者→團隊管理→架構設計→產品管理

都在 B2B 產品公司做軟硬整合方案

2017 開始接觸 DevOps 觀念並參與社群活動

Agenda

#1 近兩年的實踐方式

#2 遭遇的困難及應對

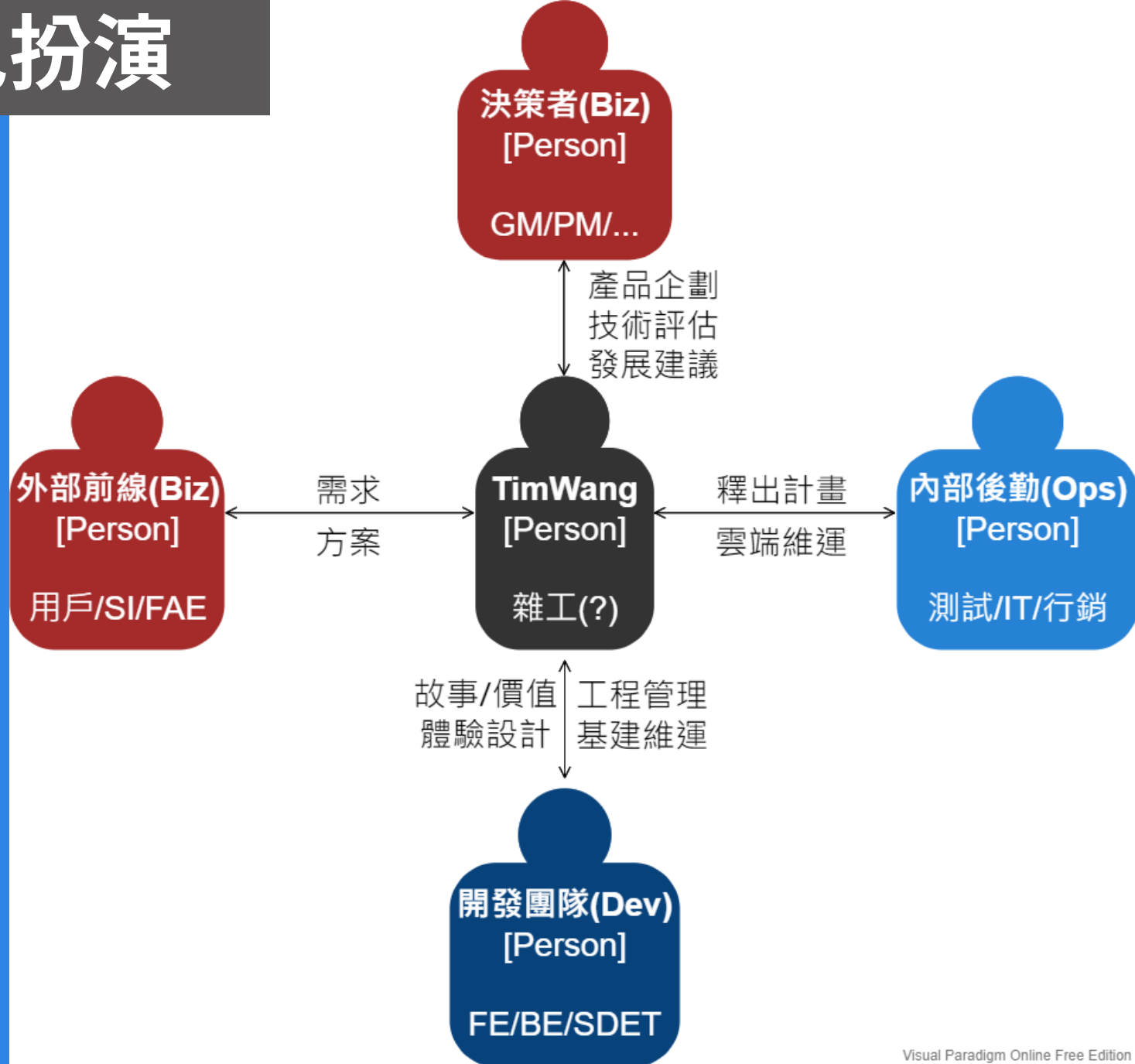
#3 如何幫助 DevOps 順利落地

#1

一個 DevOps 各自表述

分享我與團隊過去兩年的歷程

個人角色扮演



Visual Paradigm Online Free Edition

團隊背景狀況



想要我的知識嗎? 可以全部都送給你，
我都放在 **Email** 跟 **註解** 裡了，去找吧！

- 工業級 **On-premise** server app.
- 產品這兩年正在拓展新市場與商模
- 團隊擴編但知識管理與傳承在**有與沒有之間**
- 工程管理靠純手工工藝
- 歷史需求及設計文件難以追溯
- 曾跑過玄妙版的 Scrum (有形式迭代沒價值增量)

團隊目標設定

Dev:

建立工程管理加速遞交價值

Ops:

導入新工具提升流程效率

Biz:

獲取回饋拓展銷售機會

	Aspect of Software Delivery Performance*	Elite	High	Medium	Low
交付價值的頻率	Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months 一年
交付價值的前置期	Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month 一周到一個月	Between one month and six months
修正問題要多久	Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day ^a	Less than one day 一天到一個月	Between one week and one month
改版出包的比率	Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15% ^{b,c}	0-15% ^{b,d}	0-15% ^{c,d}	46-60% 暫不參考

參考 DORA State of DevOps Report (2019) 為指標

開發自動測試項

- 1. 影響力地圖
- 2. 事件風暴
- 3. 知識管理 (KM)

- 1. 產物管理 (AM)
- 2. 自動化打包/傳遞

- 1. 導入虛擬化平台
- 2. 自動化部署
- 3. 自動測試服務化



自動化建置

- 1. 自動化測試
- 2. SCA/SAST

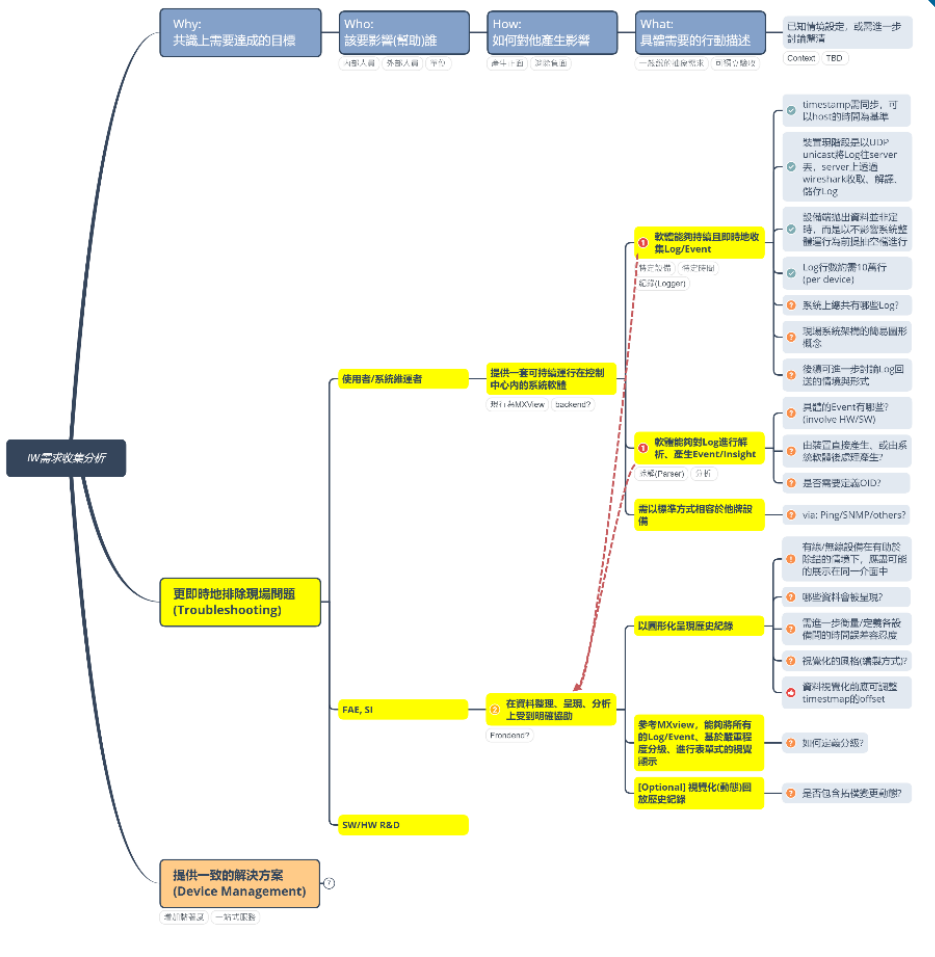
- 1. 營運數據視覺化
- 2. 平台數據長期趨勢

- 1. 展示服務上雲
- 2. 追蹤量化線上經銷商大會成效

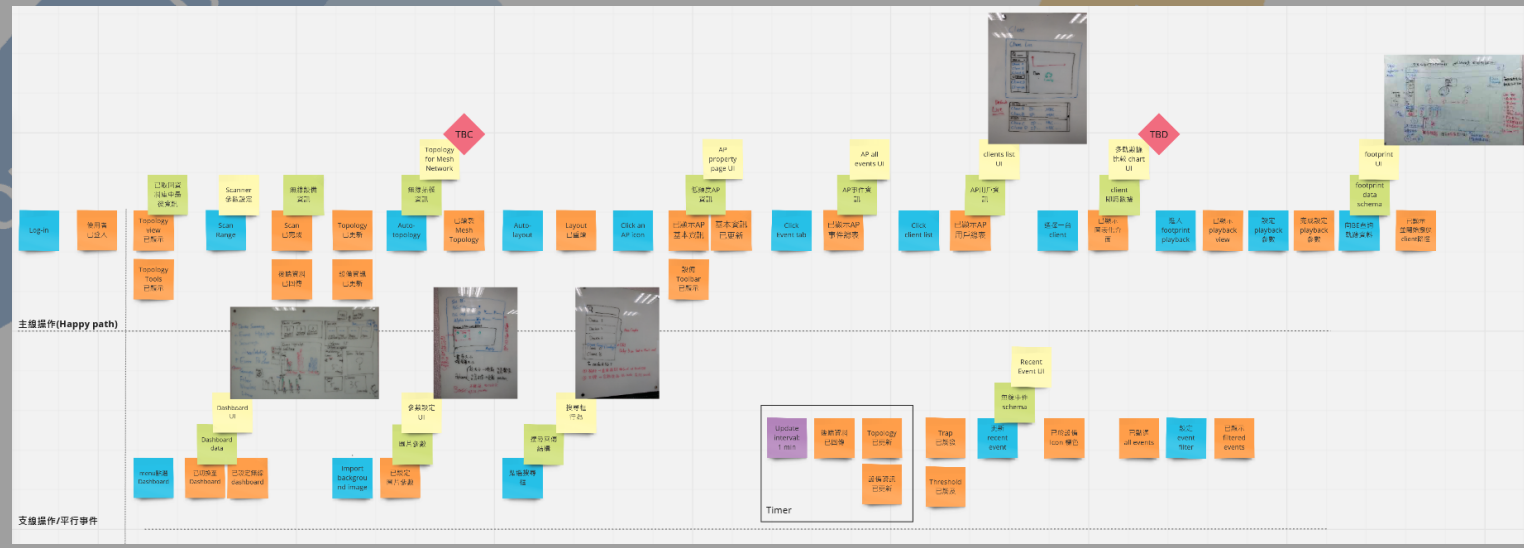
1. 影響力地圖

2. 事件風暴

3. 知識管理 (KM)



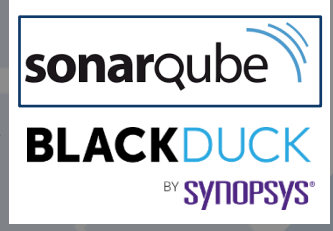
PLAN
BUS TECH



1. 產物管理 (AM)
2. 自動化打包/傳遞

開發自動測試項

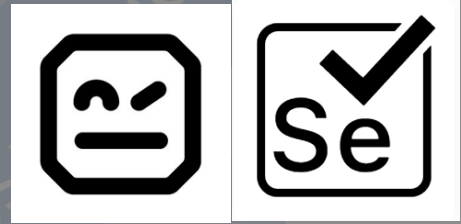
靜態掃描/套件分析



原始碼管理
分支管理



每日建置
隨時建置



測試自動化



打包/遞交



產出物管理

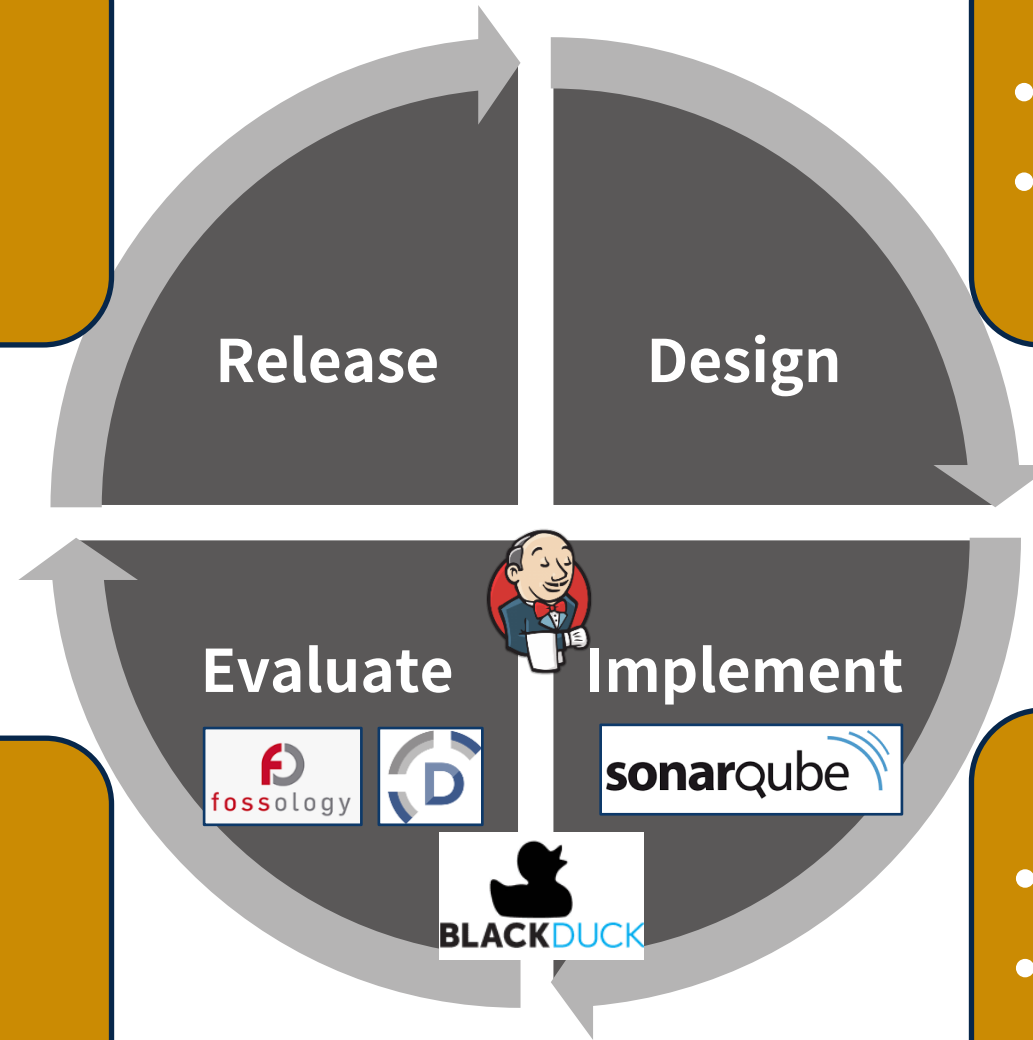
Auto-Build

1. 自動化測試
2. SCA/SCA

SSDLC (Secure SDLC)

- 數位簽章
- Hash

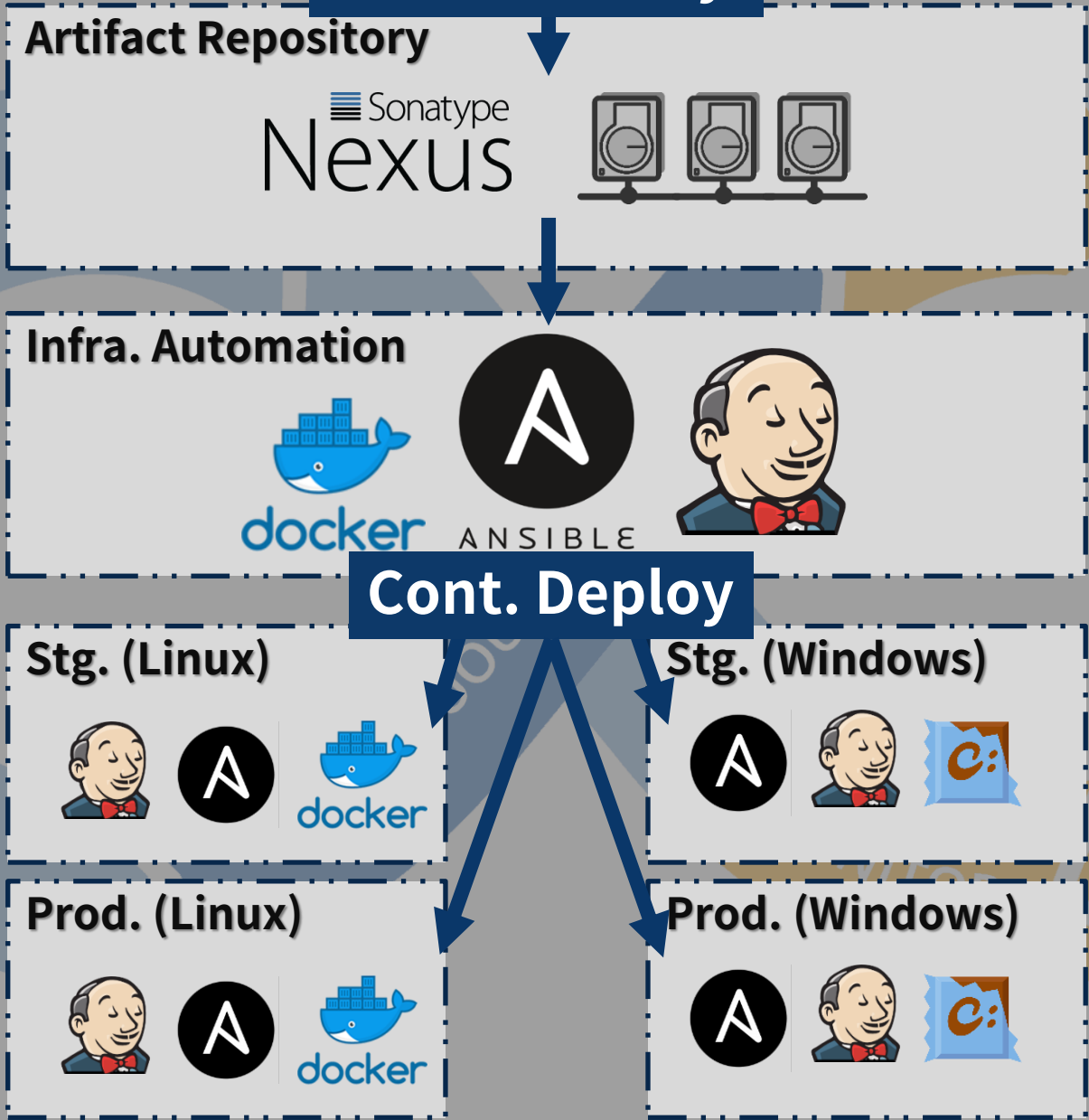
- 盤點架構設計
- 評估工具鏈合宜性



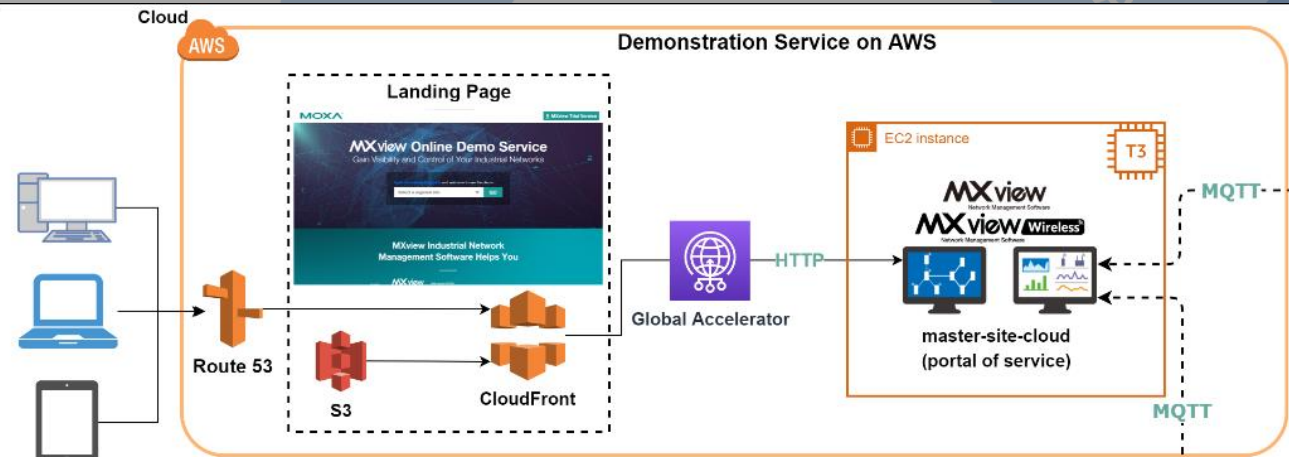
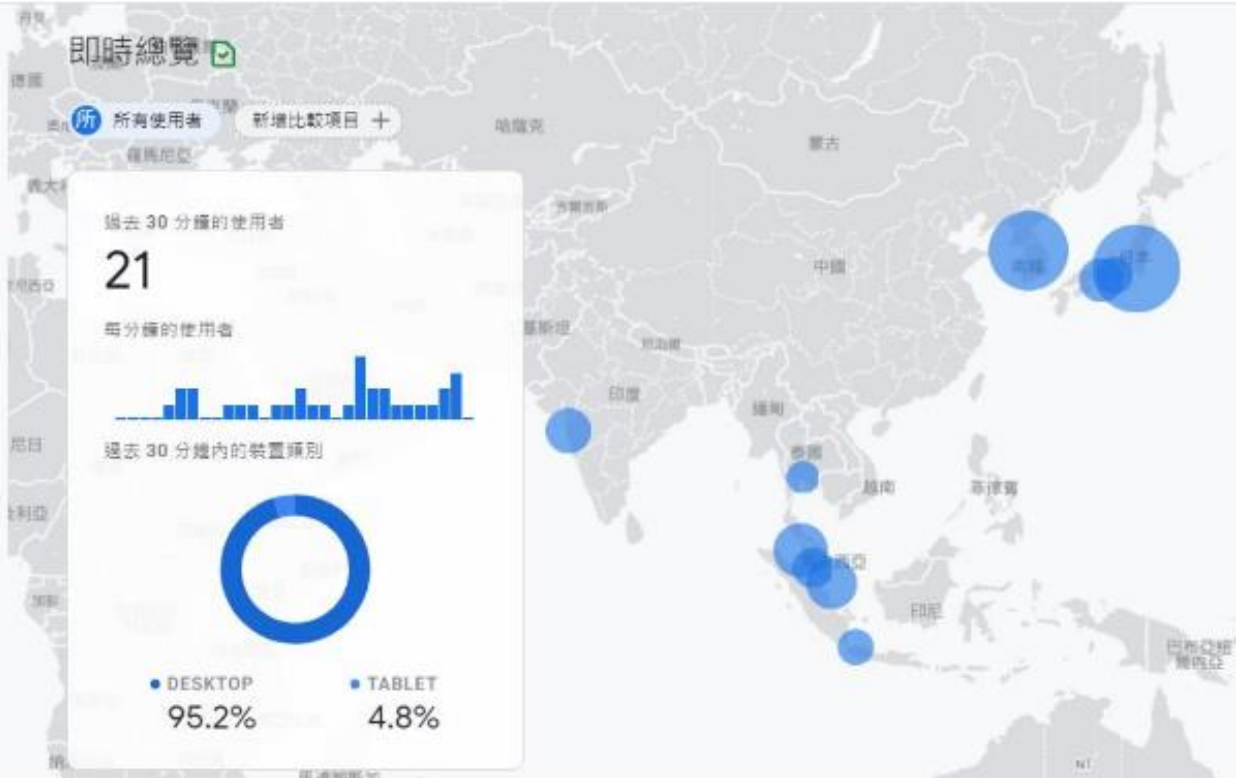
- 定期進行黑箱弱掃
- 開源合規 (FOSS)
- Software BOM

- 週期靜態掃描 (SCA)
- 判讀報告、設定通知

Cont. Delivery



- 1. 導入虛擬化平台
- 2. 自動化部署
- 3. 仿真環境壓測



MXview
Network Management Software
MXview Wireless
Network Management Software

Staging Environments



Production Environments



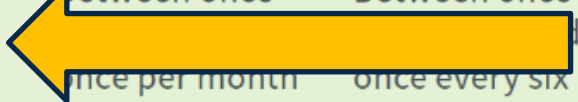
Cloud Demo Service



1. 營運數據視覺化
2. 平台數據長期趨勢

	Aspect of Software Delivery Performance*	Elite	High	Medium	Low
交付價值的頻率	Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once once per week	Between once once per month	Between once once every six months
交付價值的前置期	Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one one week	Between one week and one month	Between one month and six months
修正問題要多久	Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Less than one week	Between one week and one month
改版出包的比率	Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15% ^{b,c}	0-15% ^{b,d}	0-15% ^{c,d}	46-60%

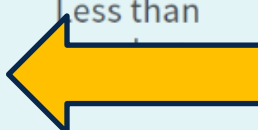
按需/每天部署



一天內



少於一天

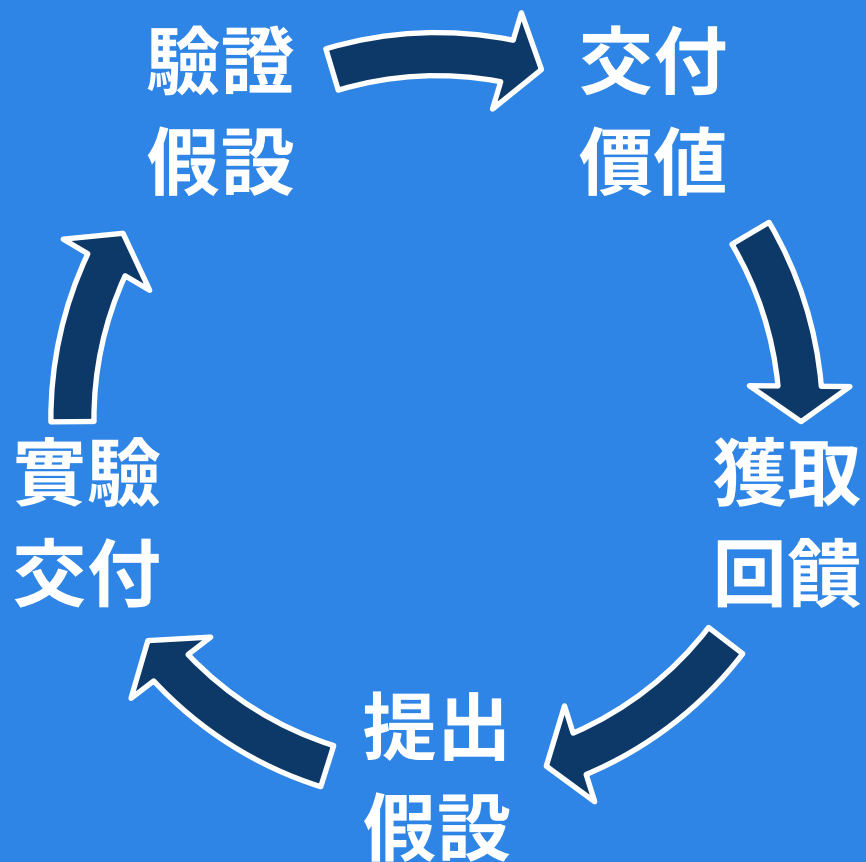


暫不參考

年交付 → 日(季)交付 / 透過展示平台接觸 1000+ 新使用者

實施 DevOps 對於產品的意義

- 確保品質同時**加速價值交付**
- 增加與市場**互動獲得回饋**
- **加速假設驗證循環**
- 呼應**三步工作法**

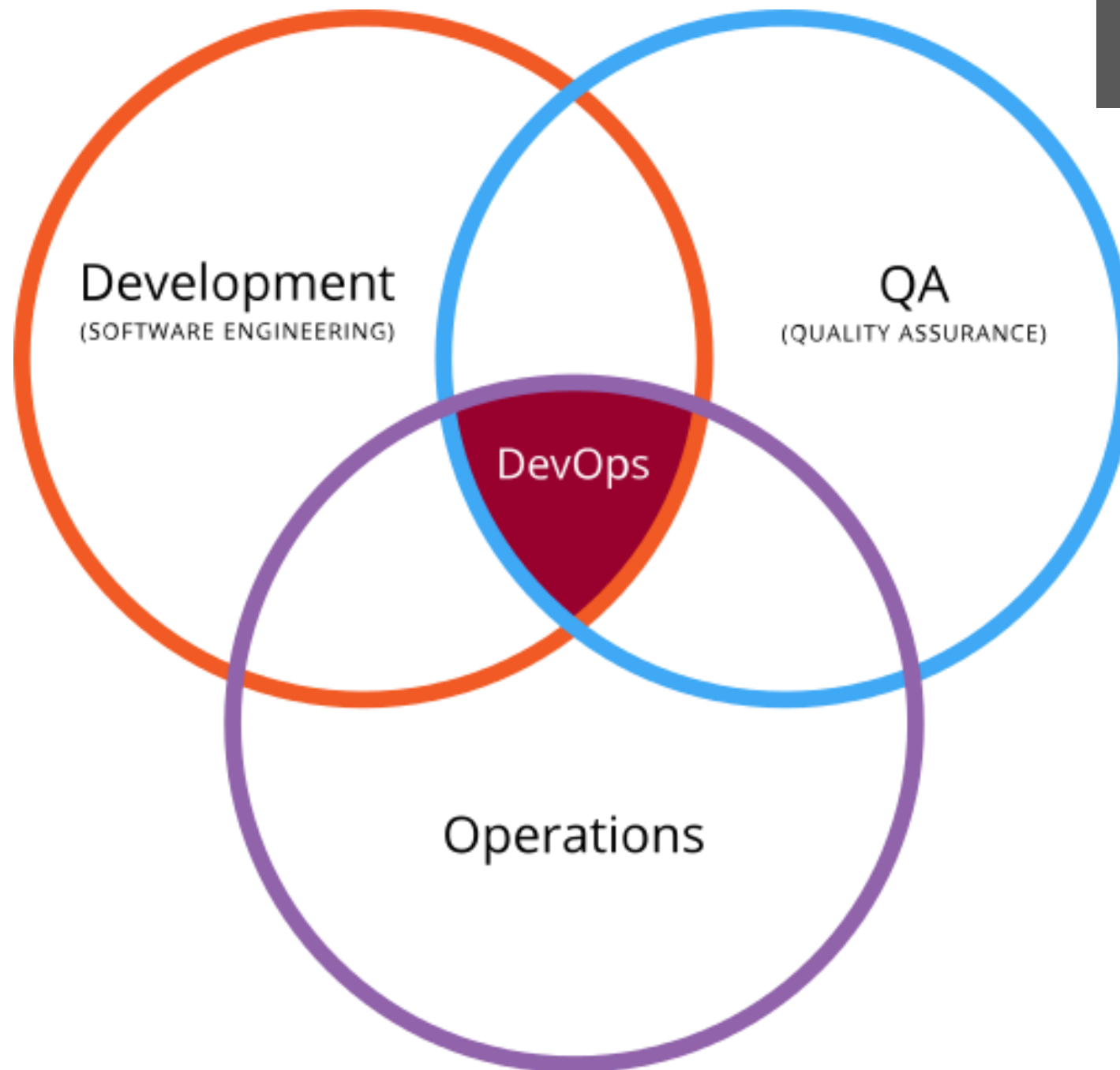


#1 小結

- 不僅做 CI/CD 區間解法，更藉由 DevOps 打穿整個價值交付
- 妥善管理知識(KM)與工程流程(DX)有助團隊能高效長久運行
- 品質先行，自動測試佔整體投資 50% 以上

#2

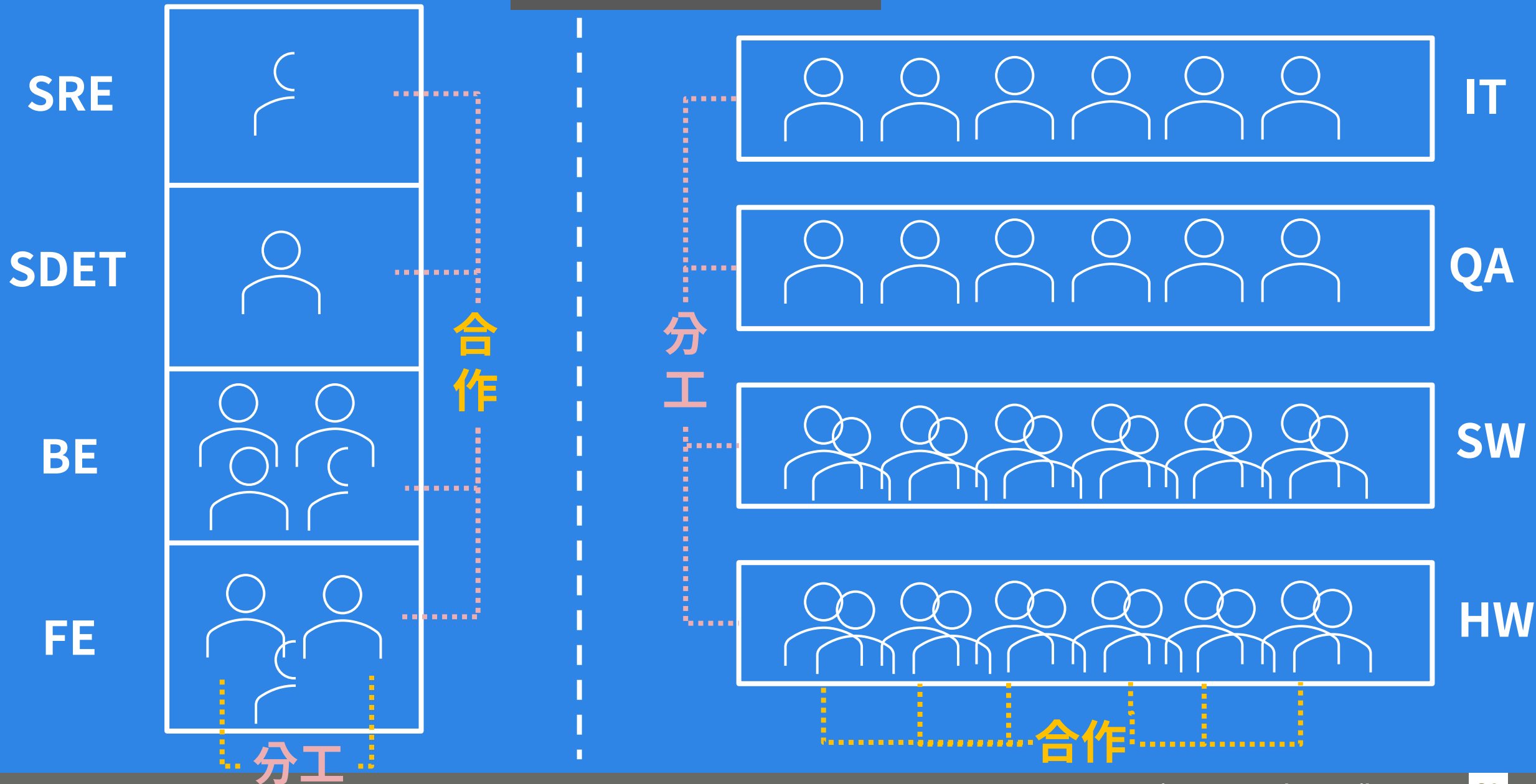
每個公司/產業都有難為之處
有掉坑才像是人生 T_T



Product Team

共同目標

Function Team (Silo?)



Class SRE Implements DevOps, but...

當 IT 團隊的目標不含**軟體產品服務**時

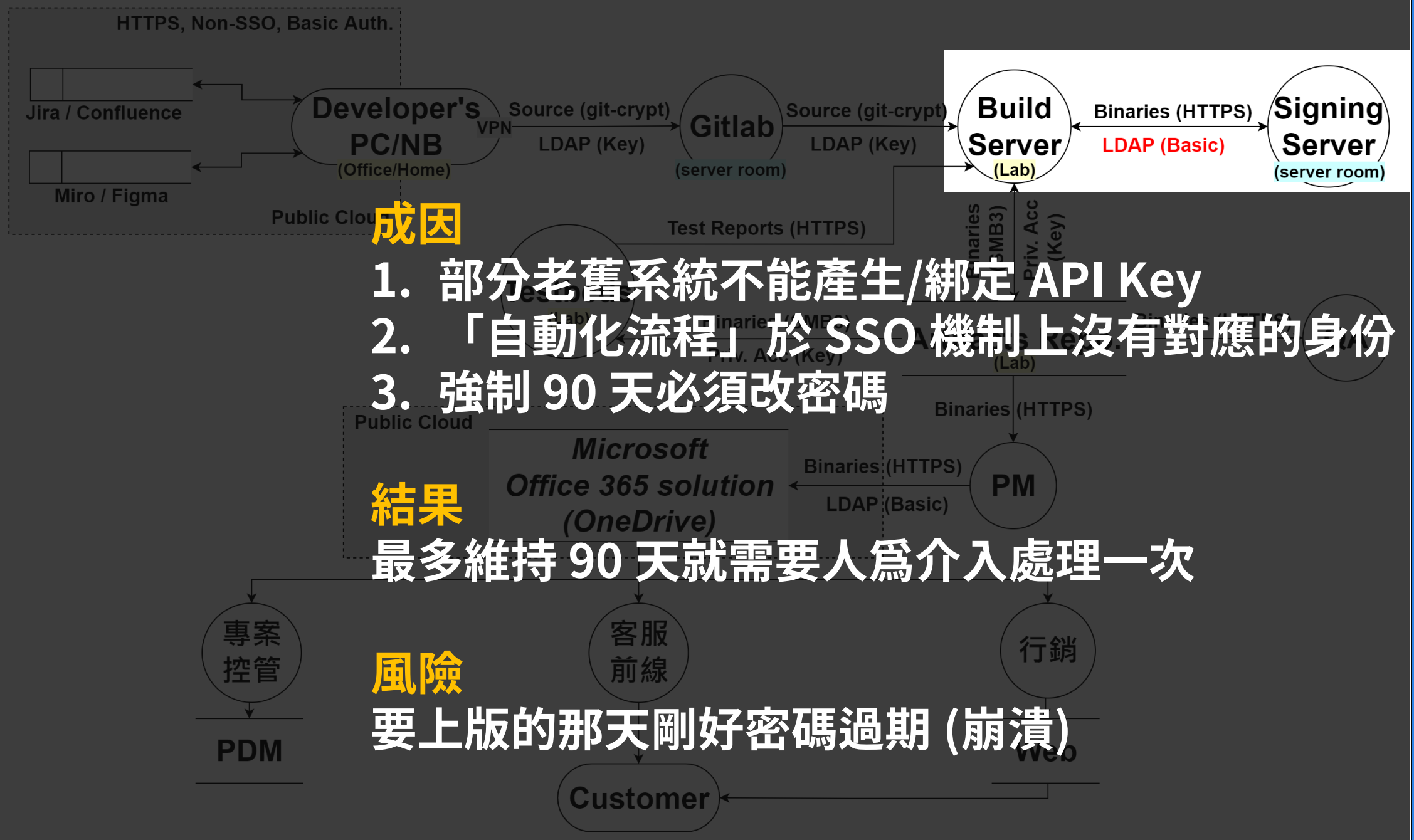
- IT = 企業運維 (成本中心)

- 管網路、顧 VPN、買/修電腦、資安認證、軟體授權、開發內部流程系統、工廠生管系統...

Dev 端發起的 DevOps 改變，
通常得兼當基建 SRE (一條龍)

常見的 IT 政策

1. 強制 X 天必須改密碼、最短 Y 天、回溯 Z 代
2. 部分老舊系統不能產生/綁定 API Key
3. 強制 MFA，然而部分系統不支援 MFA
4. 「自動化系統」於 SSO 機制上沒有對應的身份
5. 禁止任何人自架系統
6. 禁用一切開源方案 (社群免費版)
7. 未登入網域禁止任何傳輸
8. 禁連外網



成因

1. 部分老舊系統不能產生/綁定 API Key
2. 「自動化流程」於 SSO 機制上沒有對應的身份
3. 強制 90 天必須改密碼

結果

最多維持 90 天就需要人為介入處理一次

風險

要上版的那天剛好密碼過期 (崩潰)



駭客行爲已從興趣發展爲事業
企業 IT 只會愈來愈累

選基建方案的考量



降低阻力

1. 挑選合規的外部供應商
(預算/省事)
2. 盡可能遵循資安最佳實踐
(MFA/SSO/RBAC/DMZ/...)



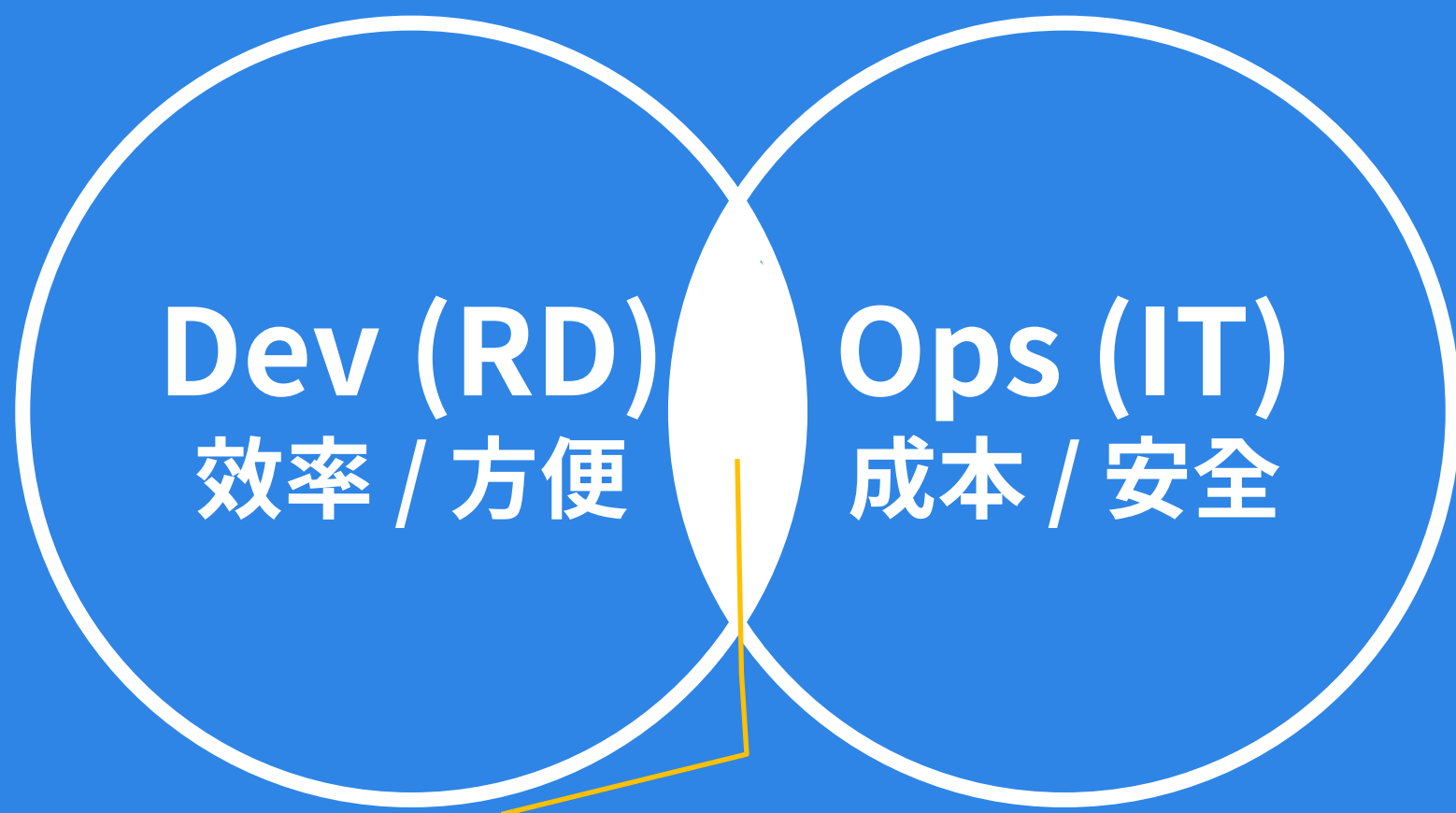
尋找產業規範支持
例: **IEC62443-4-1**
對於 **SSDLC** 的需求



尋找助力

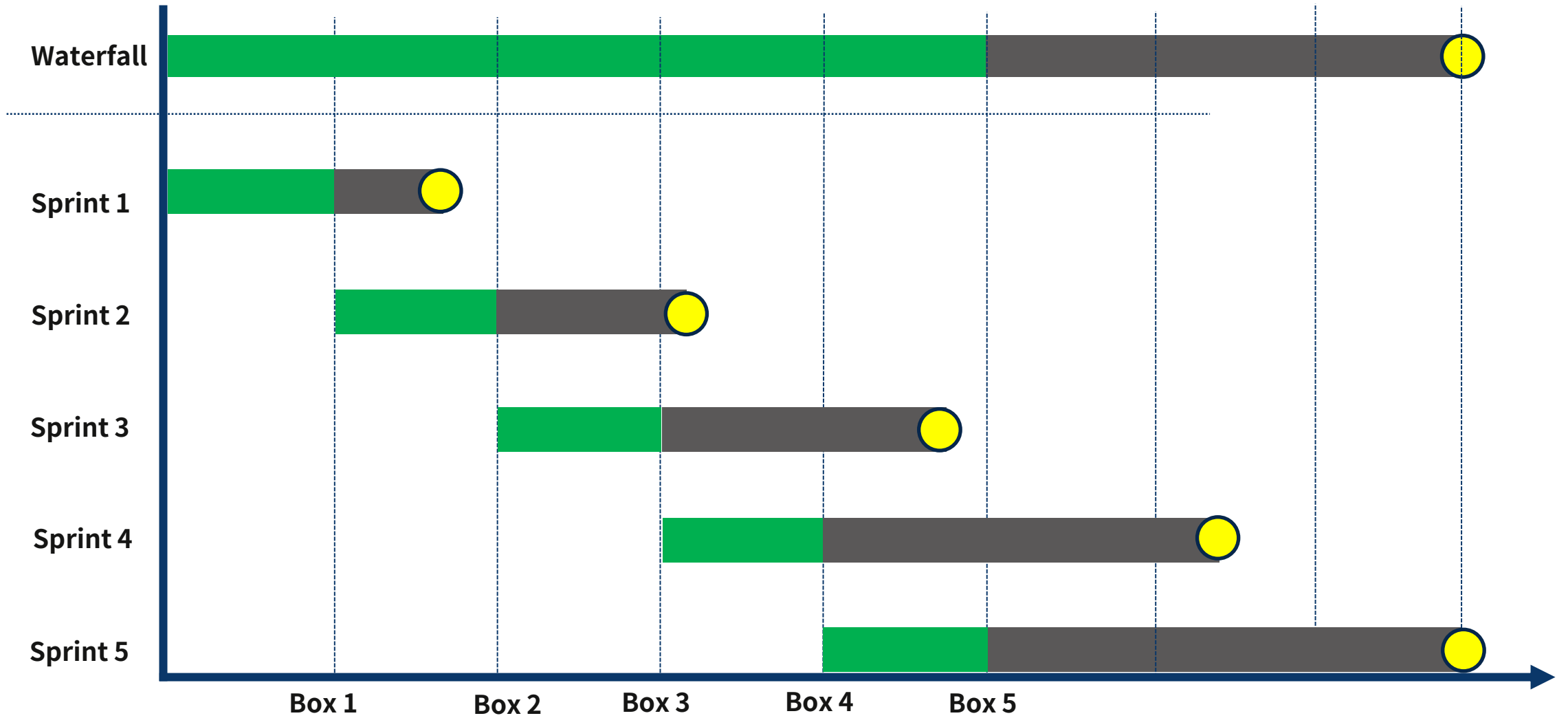
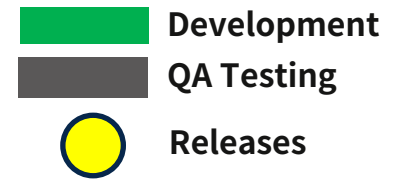
選基建方案的考量

	採購成本	運作效率	資安成本	維運成本	SLA
買現成	<p>營業費用</p> <p>OPEX</p> <p>資本支出</p> <p>CAPEX</p>	<p>延遲/頻寬</p>			
自己建	<p>資本支出</p> <p>CAPEX</p> <p>營業費用</p> <p>OPEX</p>	<p>整合較穩定</p>	<p>考量企業 合規程度</p>	<p>SRE (IT/RD)</p>	<p>SRE (IT/RD)</p>



**DevOps 要在企業內平安落地
取決於
雙方溝通的文化與平台**

QA / SQA / SDET

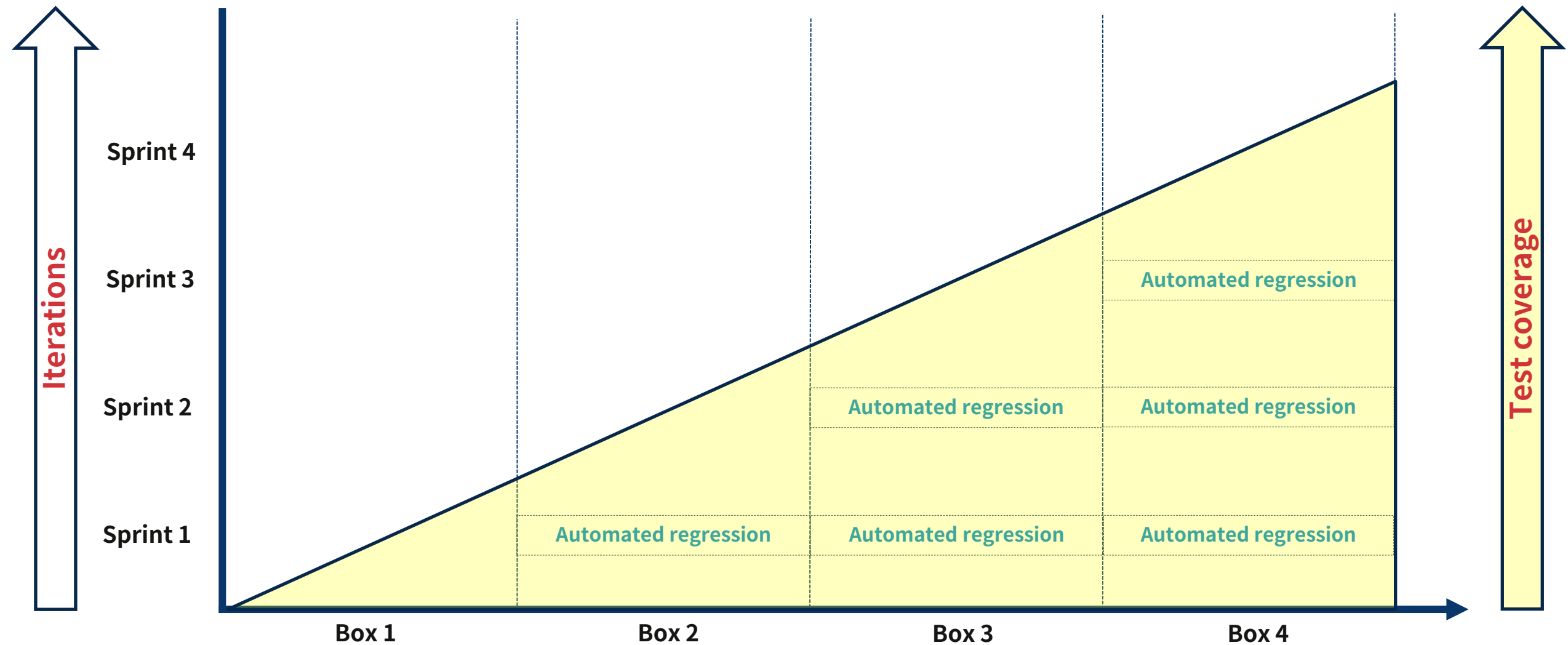


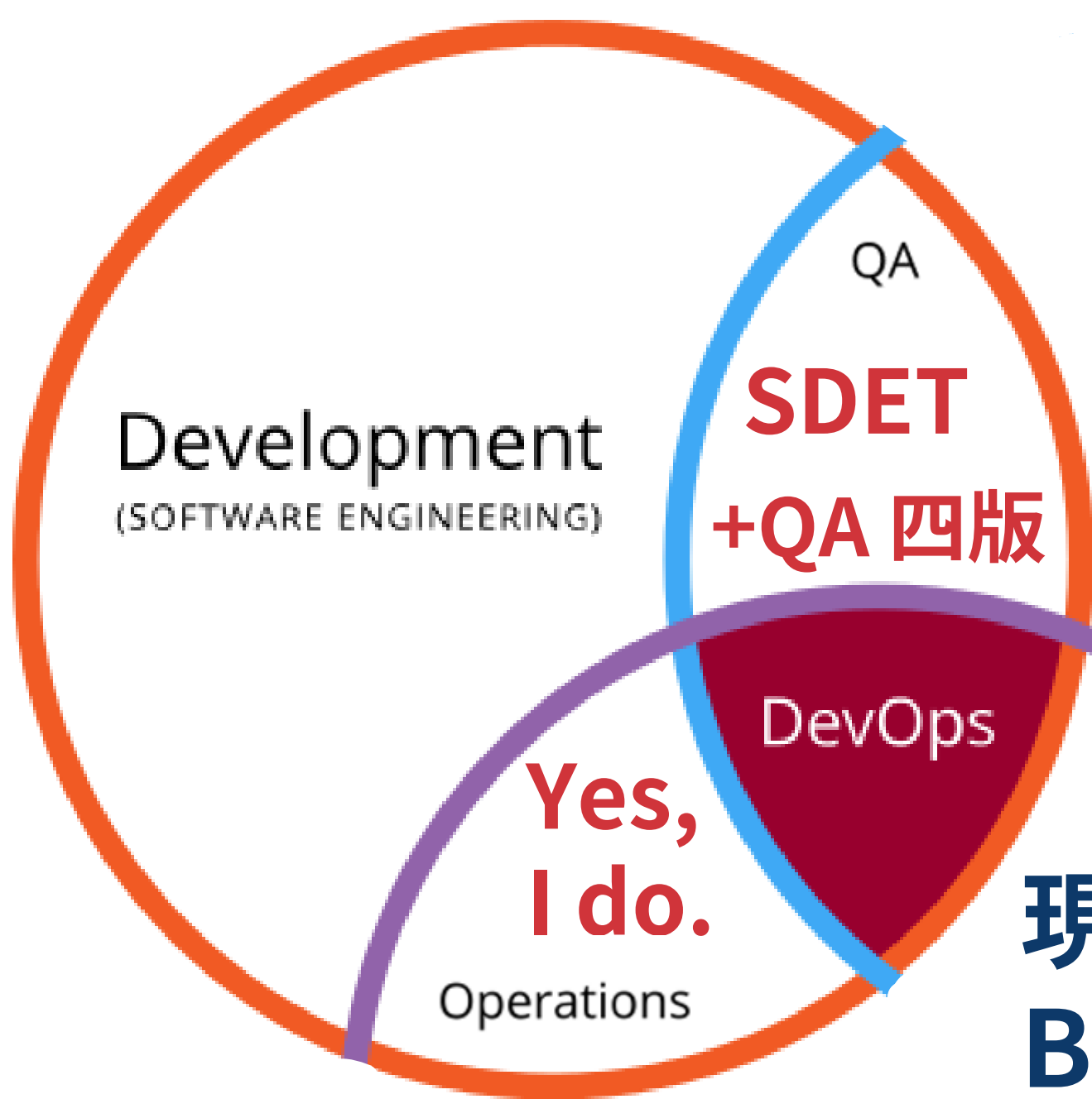
當 (S)QA 團隊無法長期投入特定軟體產品服務的持續測試時

- 迭代愈多成本愈高
- 迭代愈快錯誤愈多
- 自動化測試開發(SDET)人才非常稀缺

迭代愈快錯誤愈多
迭代交付 = 錯誤

持續測試(開發)





現實總是艱難！
But it works...



DevOps Taiwan

半個月一次算少到不行

算少

以企業來說正常

非常少

每半個月部署一次到生產環境算多嗎？
這已經不包含hotfix



每家公司的需求不同

應該不算太多，但這還是要看各種主客觀因素

除了比快還要比品質啊，持續交付 bug 不太對。

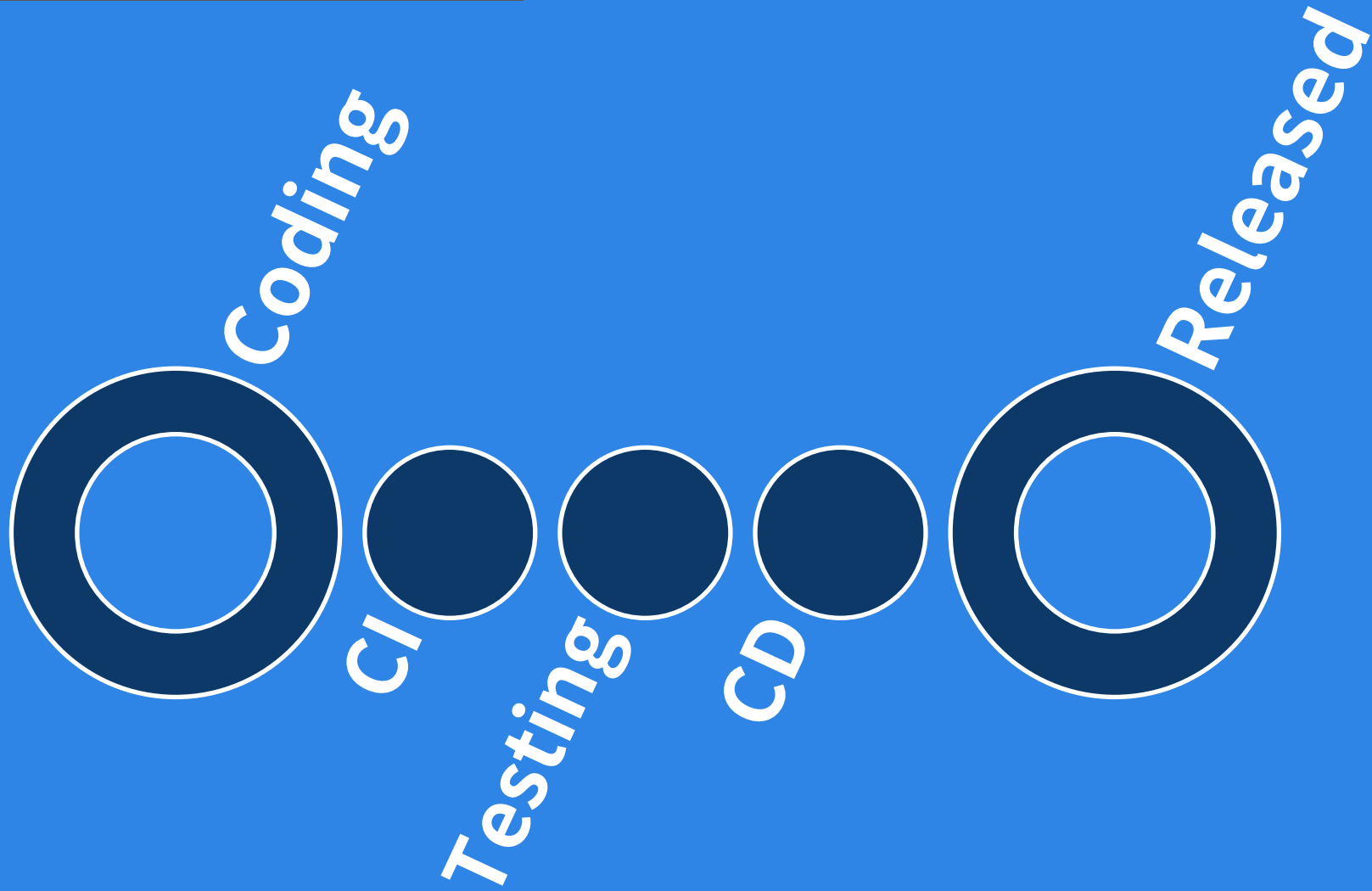
可以擁有很快的能力，但不代表需要很快。



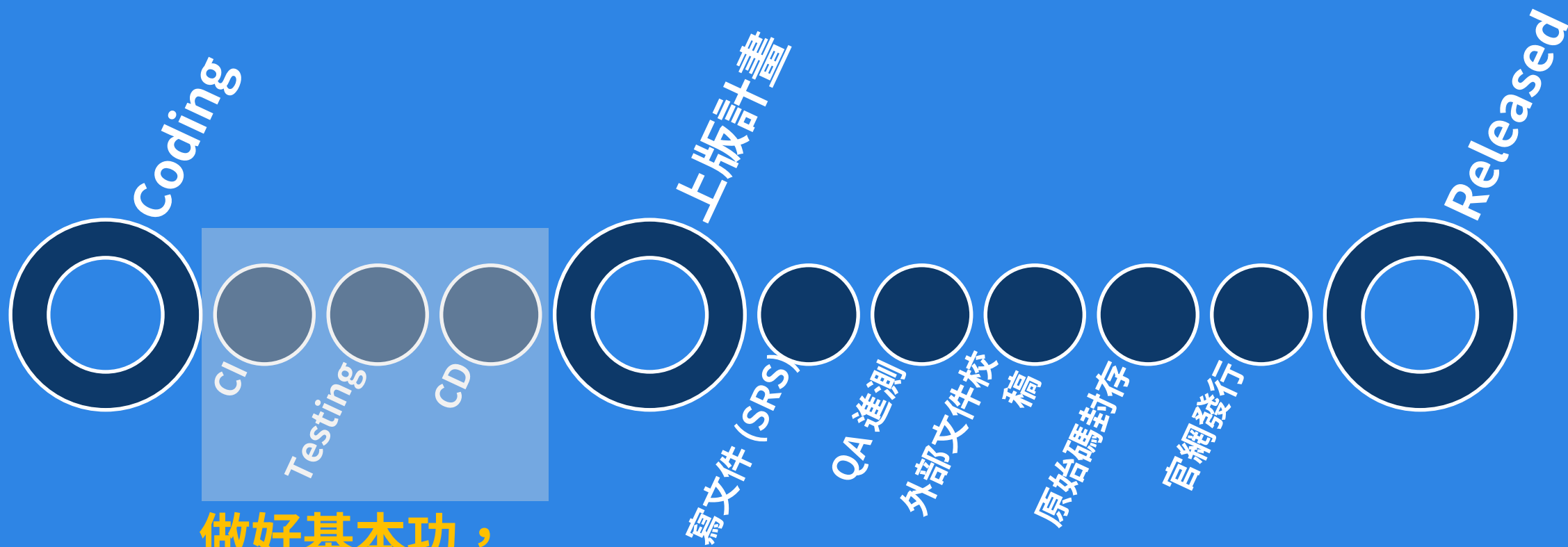
和其他36人

27則留言

2C/SaaS 說的價值流



2B/我們的價值流



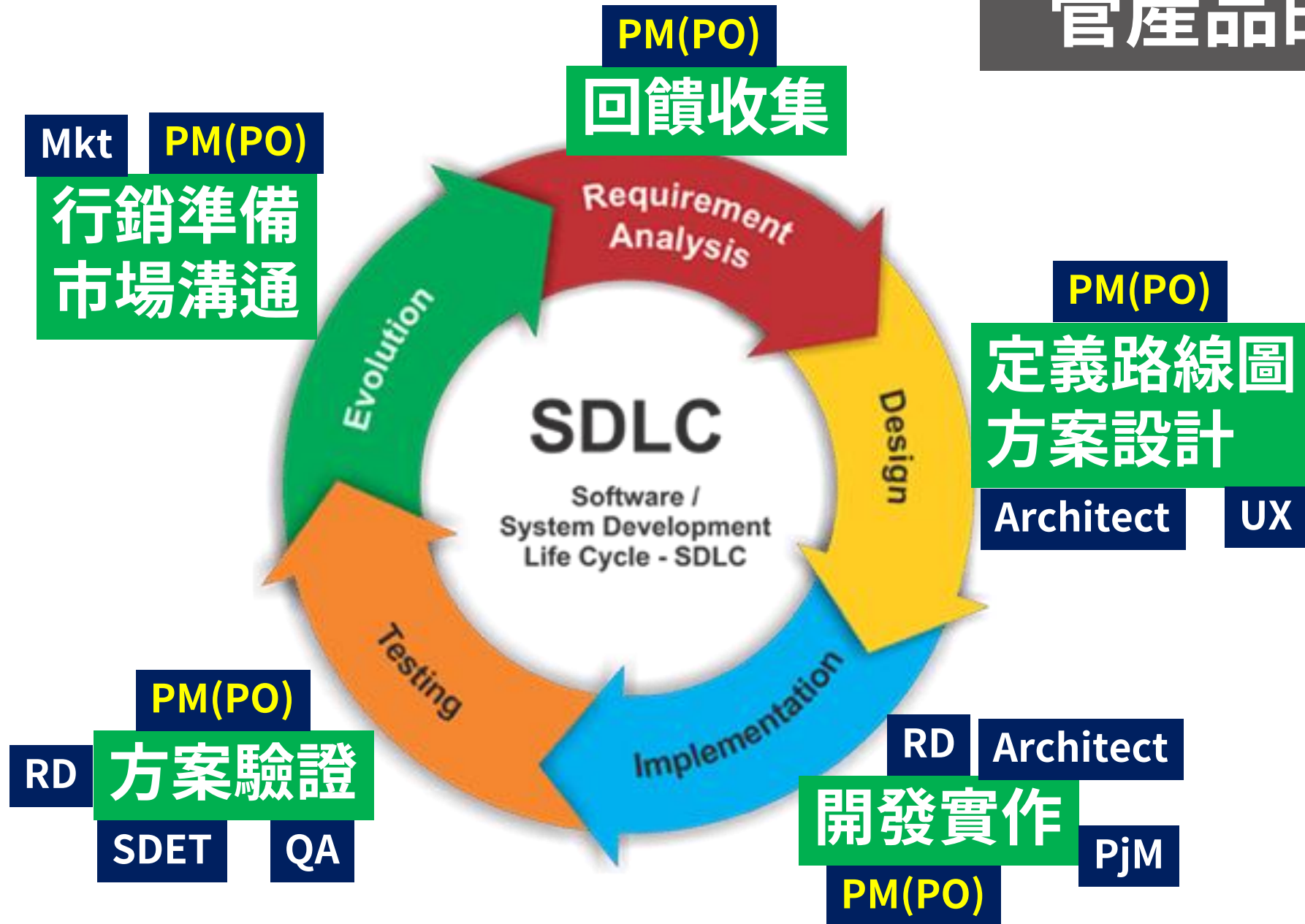
做好基本功，
尋求商業突破

#2 小結

- 實踐 DevOps 很可能改變團隊構型
- 跨單位同步目標很重要
- 盤點價值流找突破口

#3

過來人看 DevOps 如何落地 那些基本而重要的因素



帶團隊的課題

商機 / 價值

方案 / 需求

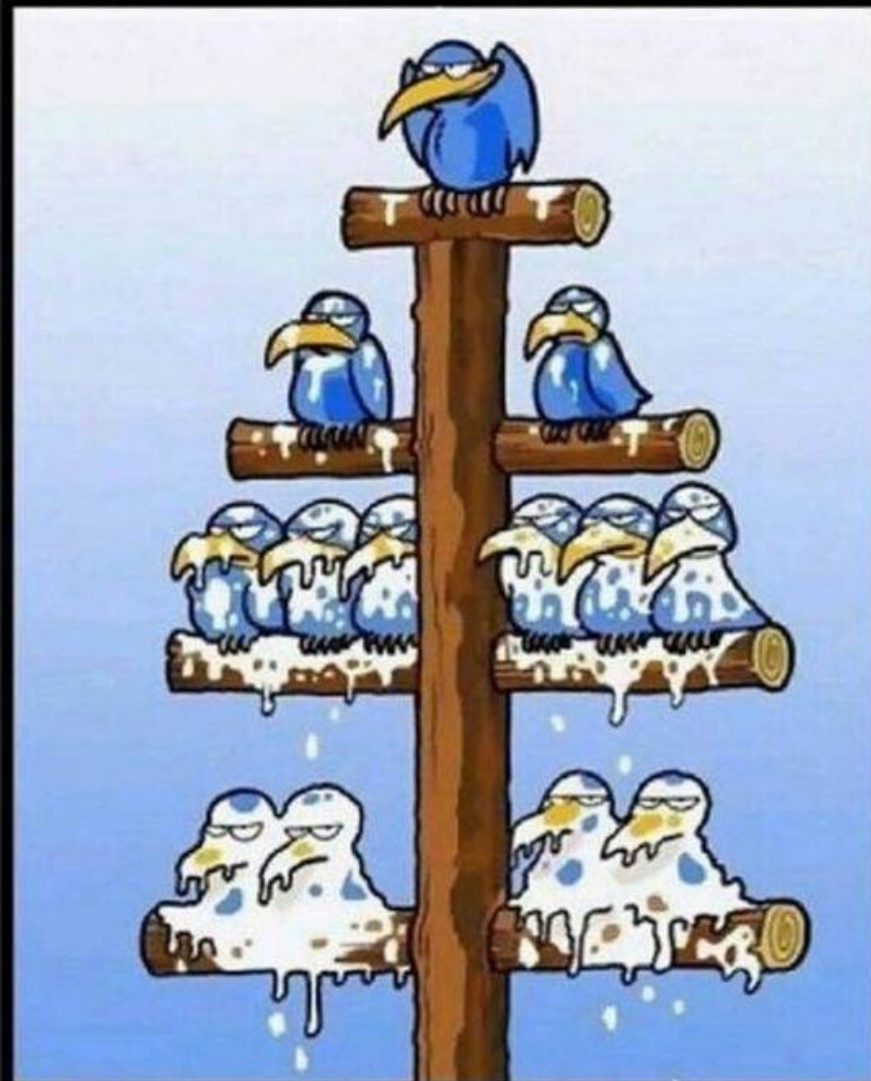
設計 / 架構

自動化省時



正向循環

When top level guys look down they see only shit.



When bottom level guys look up they see only assholes.

老闆

Sales / BD

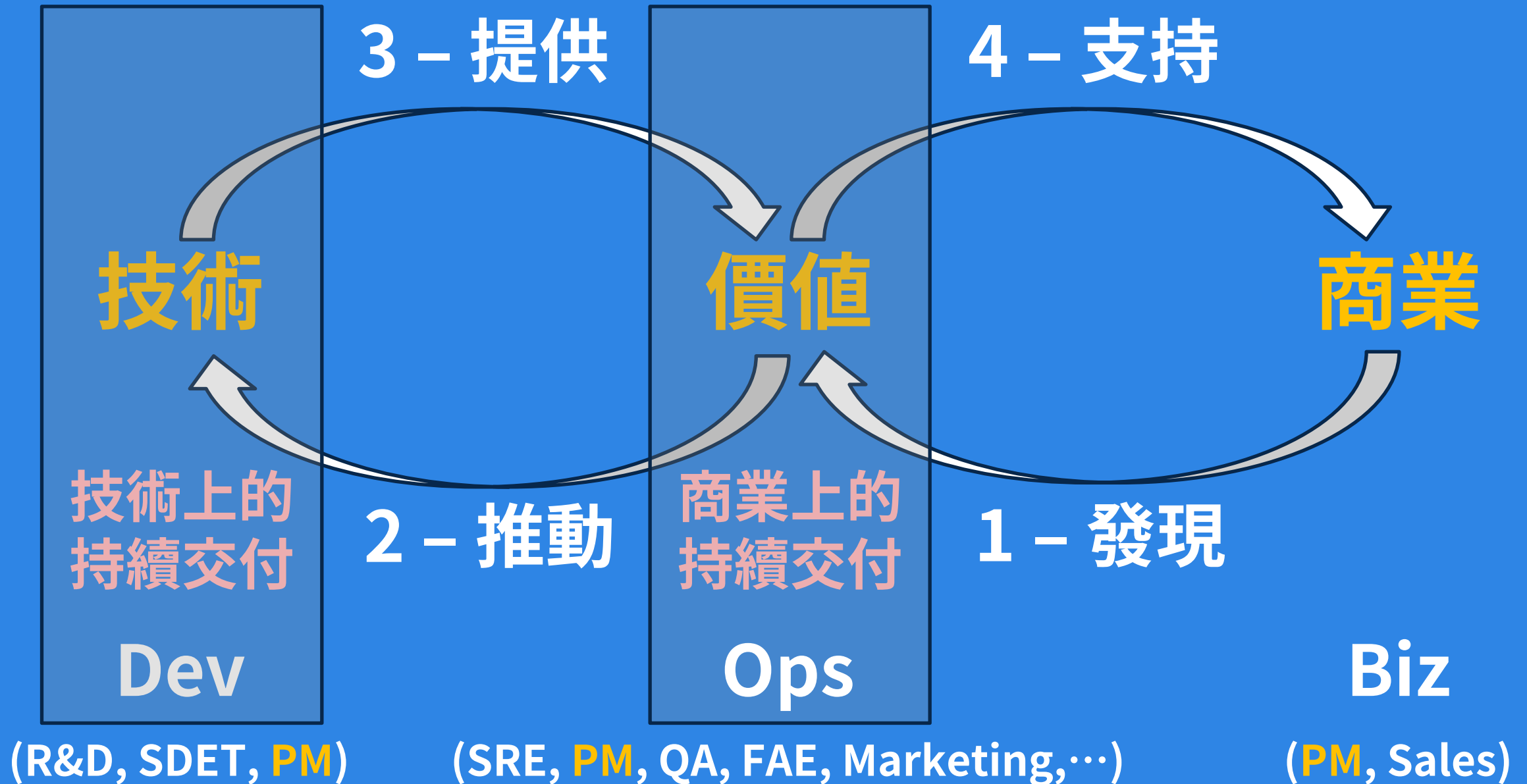
主管 / PM

基層 / RD



負向循環

Biz DevOps = 以 Biz 引領 DevOps 推動



Product Team

共同目標

Function Team (Silo?)

SRE



SDET



BE



FE



IT



QA



SW



HW



對齊商業目標組建 pilot-run team

三步工作法之一：由左到右快速流動

- 產品商務

- 對客戶**探詢價值**、對團隊**說需求故事**、將需求**轉為可行計畫**
(Impact Mapping、User Story Mapping、Event Storming)
- 支持以 **Product Team** 構型來建構團隊

- 技術開發

- CI / CD / **Test Automation**
- **Trunk-based development (TBD) & Feature Toggle**
- **Virtualization / Infra. as Code**

三步工作法之二：由右到左快速反饋

- 產品商務

- 因應業務場景(2B/2C)，設計資料收集機制及分析方法
- 帶領團隊 **Retrospective** (回顧)
- 基於各種回饋進行 **Refinement** (微調)

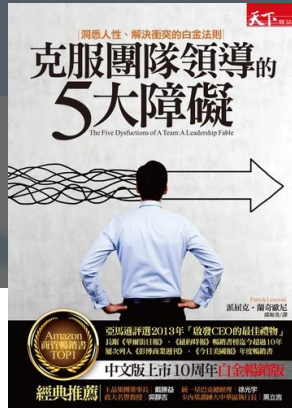
- 技術開發

- 規劃並實踐 log/trace/metric 協助資料驅動決策
- 落實知識管理 (Knowledge Management)
- **商業思維**，對於討論理解需求不排斥

三步工作法之三：持續學習

- 全體成員
- 假設(hypothesis)驅動
- 追求價值導向
- 不咎責文化
- 塑造**學習型組織**

Lencioni's Five Dysfunctions of a Team



“No matter what the problem is,
it's a **people** problem.”

—Gerald M. Weinberg

總結：

1. 處理**人與組織**的問題
2. **自動測試**是加速循環的關鍵
3. 釐清**價值流**，以**商業目標**驅動
4. **目標與溝通**重於 工具與流程
5. **持續演化**才是最佳策略

Q&A

歡迎加入 **DevOps Taiwan** 分享交流經驗
<https://www.facebook.com/DevOpsTaiwan/>

