



## SRE 經驗分享

從事故分析、精準監控到自動化維運

**Presented By**  
**Duran Hsieh**  
**Site Reliability Engineer**



GDG Taichung

## ABOUT ME

### Duran Hsieh (謝政廷)

專長於開發測試、效能調教、DevOps 相關技術，過去曾擔任技術顧問，提供國內外企業技術諮詢與數位轉型。目前同時為 Study4TW 社群核心成員、微軟最有價值專家、Google Developer Group Taichung 共同創辦人，曾積極參與技術社群與經營技術部落格，曾獲得三次 IT 邦幫忙鐵人賽佳作與撰寫〈動手學 GitHub 現代人不能不知道的協同合作平台〉書籍。

- Duran 技術冶煉廠 : <https://dog0416.blogspot.com/>
- Duran 速寫筆記: <https://note.duranhsieh.com/>



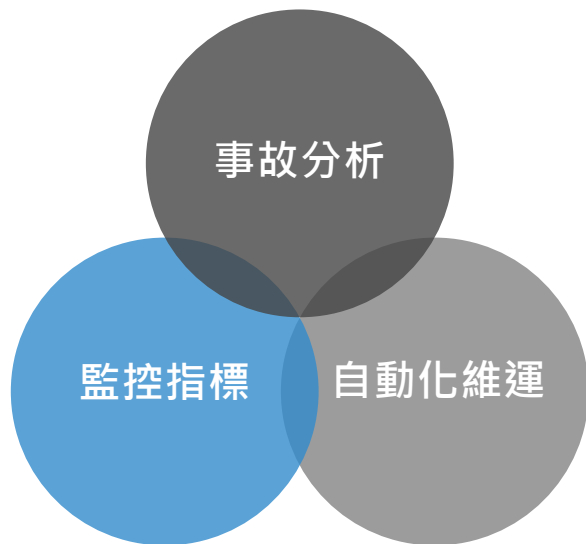
## CONTENTS

---

- 事故分析 05
- 監控指標 21
- 自動化維運 27

# SRE 重要技能

---



## 事故分析

事後分析是事件管理的關鍵部分，一旦事件得到解決就會發生



## 監控指標

用於了解歷史趨勢、比較各種因素、識別模式和異常、發現錯誤和問題的系統數值



## 自動化維運

一組技術和策略，旨在通過消除對手動重複性任務的需求來簡化您的操作



# 事故分析

## INCIDENT ANALYSIS

---

# 事故三種類型 – 效益

---



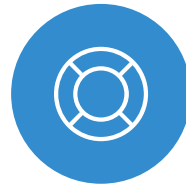
## 使用者報案

最高比例的事故回報來源尤其是基礎設施 (監控、警告) 未完善階段



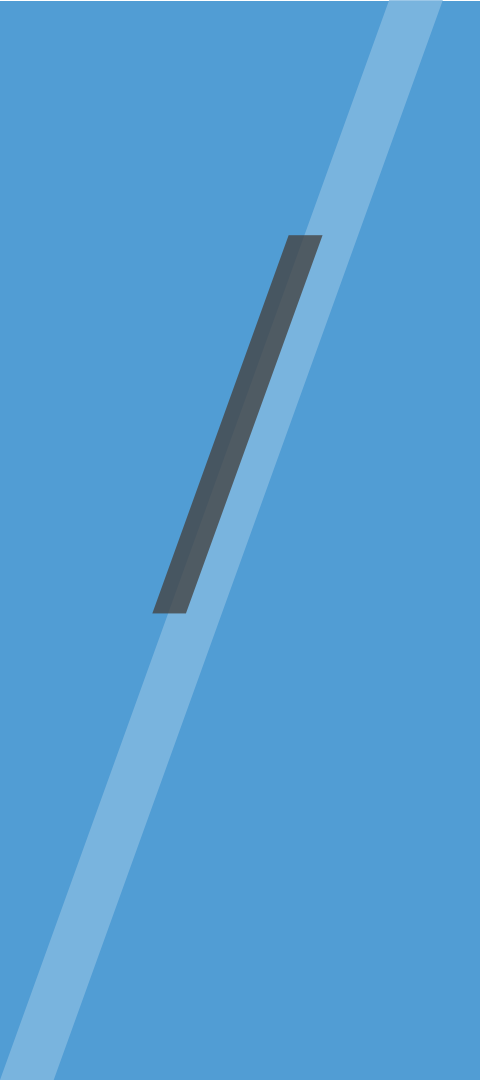
## 監控與警告

即便基礎建設完善，若未能完整掌握監控指標與過濾錯誤警告，透過監控系統仍很難發現事故



## 團隊預先察覺

成熟的監控與警告機制才有較高的機率預先察覺事故發生

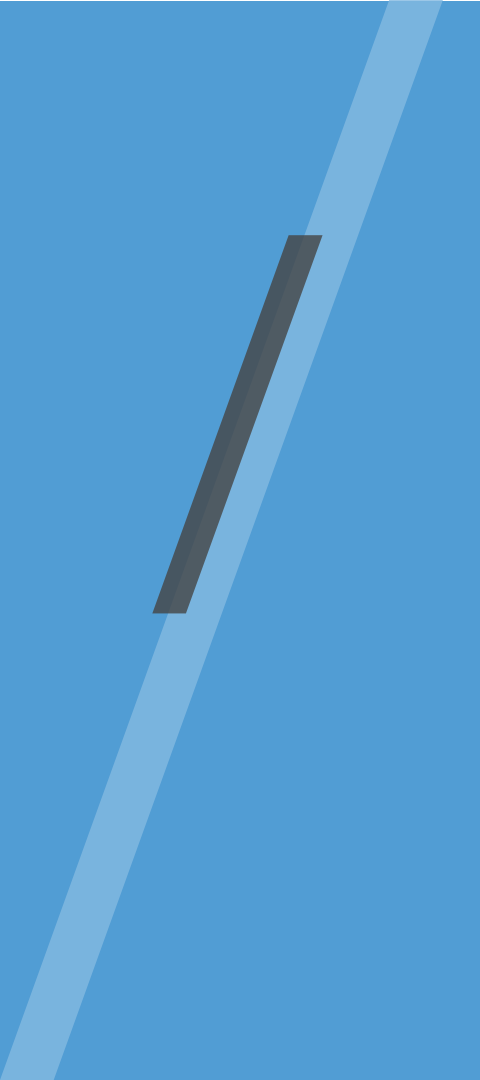


# 事故階段

## 盡可能不要採用沒測試過的 修復計畫與步驟

---

在相依複雜的系統環境  
可能導致服務中斷時間更長



# 事故階段

## 不要往極不可能的方向調查

---

不要想太多，優先考慮最簡單的跡象

# 事故分析

又稱為事後驗屍 (postmortem) 工作，完整的理解或文件化事故，找出事故根本原因，並檢視目前的解決方案是否能更好，目的是從中學習，避免事故再次發生。**其耗費的成本相當昂貴**



事故描述



根本原因



解決方案



正式文件



不責難文化



識別與預防



# 何時進行事故分析

---



## 使用者體驗受到影響 (SLO)

- 無法使用服務
- 使用者不能接受的服務效能
- 不穩定的功能



資料遺失



組織/團隊要求

# 事故分析階段

## 不要單獨/隨時對外回報調查進度

---

資訊不完整、錯誤的用詞、無法重現情境  
可能傳遞錯誤的訊息  
詳細確認與審核文件後再一致對外公布

# 檢討報告

日期時間: ....  
撰寫者: ....  
審核人員: .....  
事件指揮: .....



採取作為

- ... ..
- ... ..



時間軸



-----  
-----  
-----



執行摘要

.....  
.....



問題摘要

.....  
.....



從中學習

V ... ..  
V ... ..  
X ... ..

# 檢討與後續工作

---



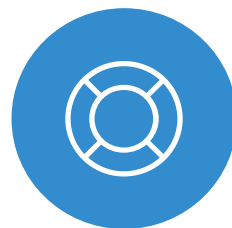
建立標準解決流程



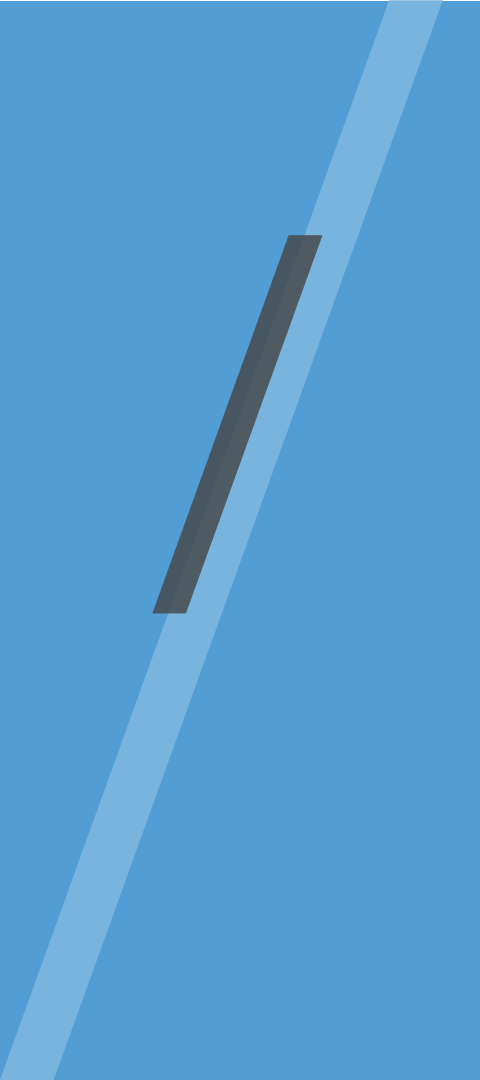
檢視事故回應  
與監控指標



主動測試



案例分享



# 事故分析分享

## 有時候會有意想不到的結果

---

除了讓其他團隊參考避免重蹈覆轍  
啟發式的討論可能出現最佳的解決方案



標準解決流程提供正確易懂的操作步驟  
有助於工程師快速釐清並修復問題

## 建立標準解決流程

文件

- 簡單易懂的用詞
- 實用：實作步驟、指令、截圖
- 共同編輯與評論 (持續維護)

團隊參與並持續演練

- 測試環境模擬
- 依據文件實作
- 持續確認是否符合需求

# 檢視事故回應與 監控指標

## 檢視監控指標

- 能否完整解決問題？
- 能否被監控？
- 能否建立預警規則？
- 能否自動修復？

## 檢視事故回應

- 事故判斷是否正確？
- 能否在第一線、第二線就解決？
- 能否快速且正確聯絡受影響的團隊？



事故後若只留下檢討報告，沒有檢視監控機制  
或改善事故回應流程，最終只有一堆瑣事



測試需要正向、反向案例，更多的測試與實作會讓你更了解系統，遇到故障的時候可以理性的作出正確的判斷

## 主動測試

- 範圍擴大 - 跨團隊測試
- 邀請使用者 - 更多的實務測試案例
- 累積操作流程與經驗 - 事故發生時擁有更多解決方案

## 案例分享

鼓勵積極參與

- 讀書會
- SRE 工作群組或會議
- 建立 V-Team



社群方式經營是不錯的方式

# 不究責文化

---



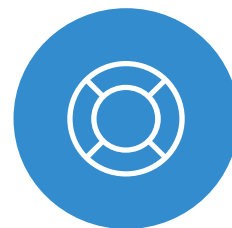
高層支持



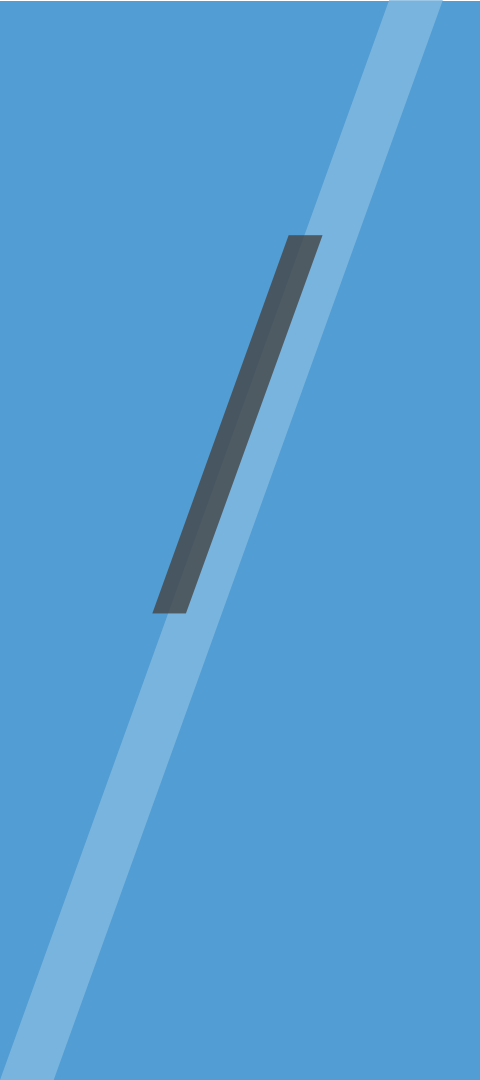
鼓勵面對錯誤



建設性批評



確保流程改善



# 不究責文化 其實有點難

---

完全不究責可能導致無法正常溝通與績效難以管理

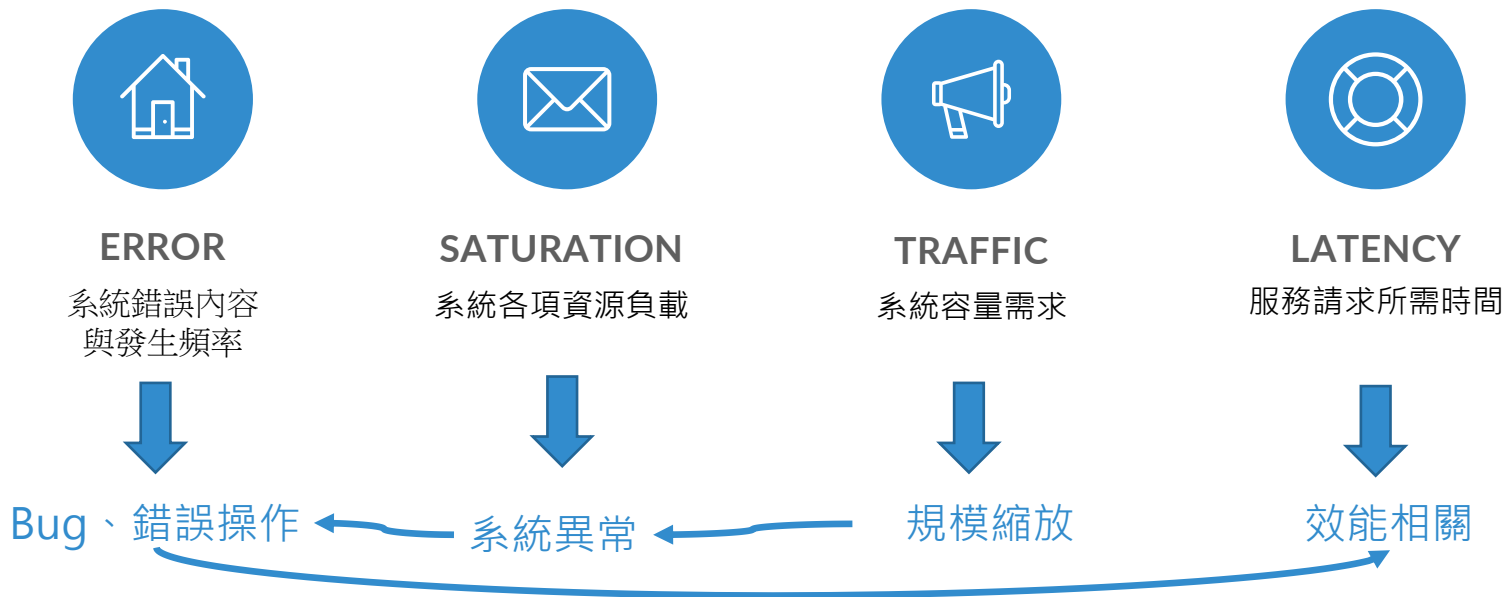


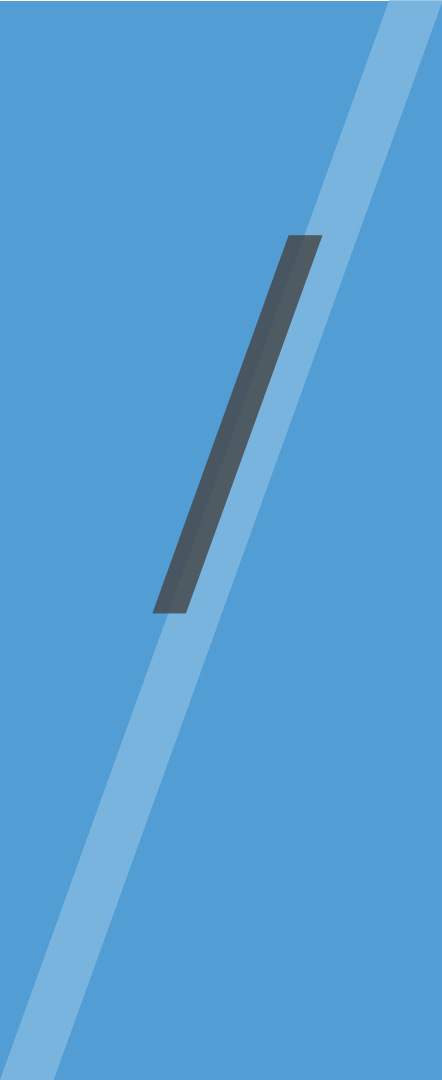
# 監控指標

## MONITORING METRICS

---

# 監控的 四個黃金信號





# 正確的監控指標 必須經過長時間觀察與測試

---

能透過監控預防事故發生或  
事故後執行自動化維運，是相當困難的



## 目標明確的指標

儀錶板上列出一堆指標相當專業，且許多樣板可以讓 SRE 快速建構監控，但是過多不實用或不清楚的指標，除了影響事故回應，最終像錯誤警告一樣被工程師忽略



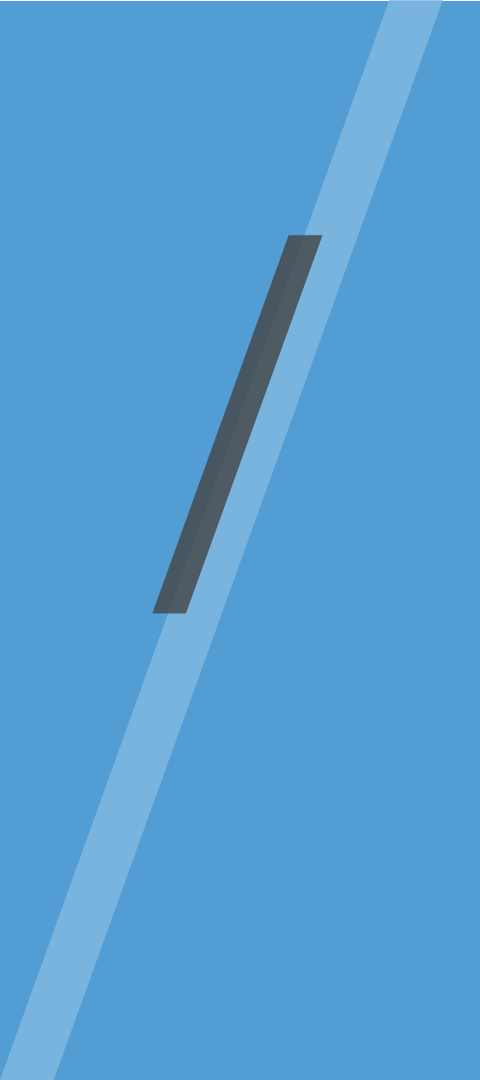
說明指標含意



數值異常時可能的影響或事故



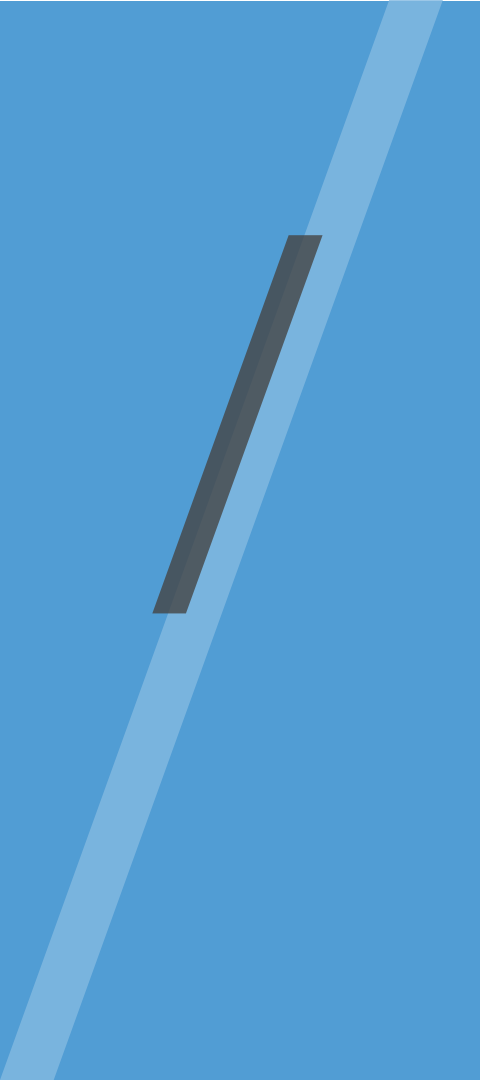
保留實用的指標



# 好的監控指標來自事故分析 很明確，但通常不容易發現

---

多數情況下，事件發生時不是呈現在單一指標  
建立良好除錯指標可以更快診斷問題所在



# 事故並非呈現在單一指標

## 很多事故無法明確監控

---

驗屍工作只有疑似原因，無法重現很常發生



# 自動化維運

## AUTOMATE IT TASKS

---

# 自動化維運目的與前提

SRE 工作不應該投資超過 50 % 在瑣事上，建立自動/半自動化維運相當重要，但前提是...

- 維運資料標準化
- CI/CD 流程標準化
- 使用一致的工具



消除無效率的操作與管理



減少人為錯誤發生



有效降低成本



減少聯繫與等待時間



# 自動化維運

---



## 減少瑣事而非簡省成本

故障排除的減少使 IT 團隊能夠在更短的時間內完成更多的項目



## 使用正確的工具

不要零碎的各自進行採用新工具進行自動化，反而讓 IT 變得複雜且難以維護



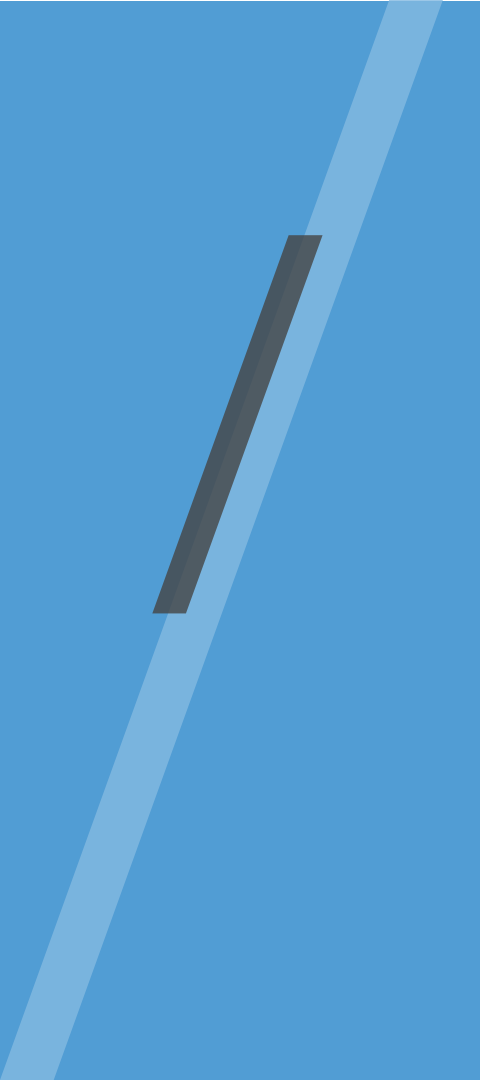
## 識別高價值流程

自動化低風險和高可見性的常規手動流程



## 檢視團隊技能

自動化維運需要對技術要求高，多數自動化流程失敗原因在於資源與技能不足



# 明確指標與消除誤報 是自動化維運的根本

---

多數情況下，事件發生時不是呈現在單一指標  
建立良好除錯指標可以更快診斷問題所在

# THANK YOU FOR WATCHING

---



若您對 SRE 相關工作有興趣  
歡迎找我討論