

91APP

# 91APP AI-Powered SDLC

91APP Sr.Manager

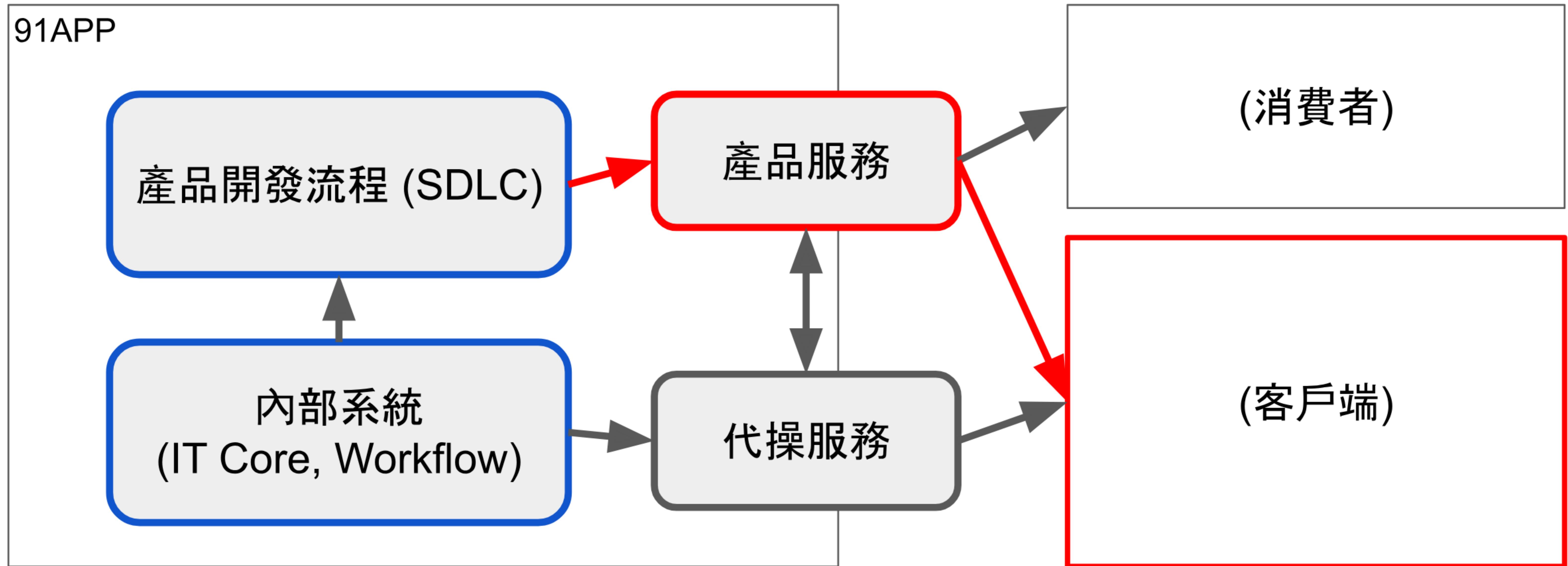
陳俊毅 Levi Chen

DevOpsDays Taipei 2025, 2025/06/05



- have been working at 91APP for 11 years
- *#SRE #Platform #DBA #MarketingCloud*

## 2. 開發 AI 產品服務 (商品上架、搜尋商品)



## 1. 用 AI 開發產品 & 改善流程與效率

零售成長的驅動者

# 創造「商務 × 行銷」 雙成長動能

以真 OMO 實踐虛實融合  
透過提升營運效益  
為全通路業績持續成長鋪路

[了解更多](#)



## 知名零售品牌 D2C 經營首選



POYA 寶雅



全家行動購



niko and ...

STAYREAL

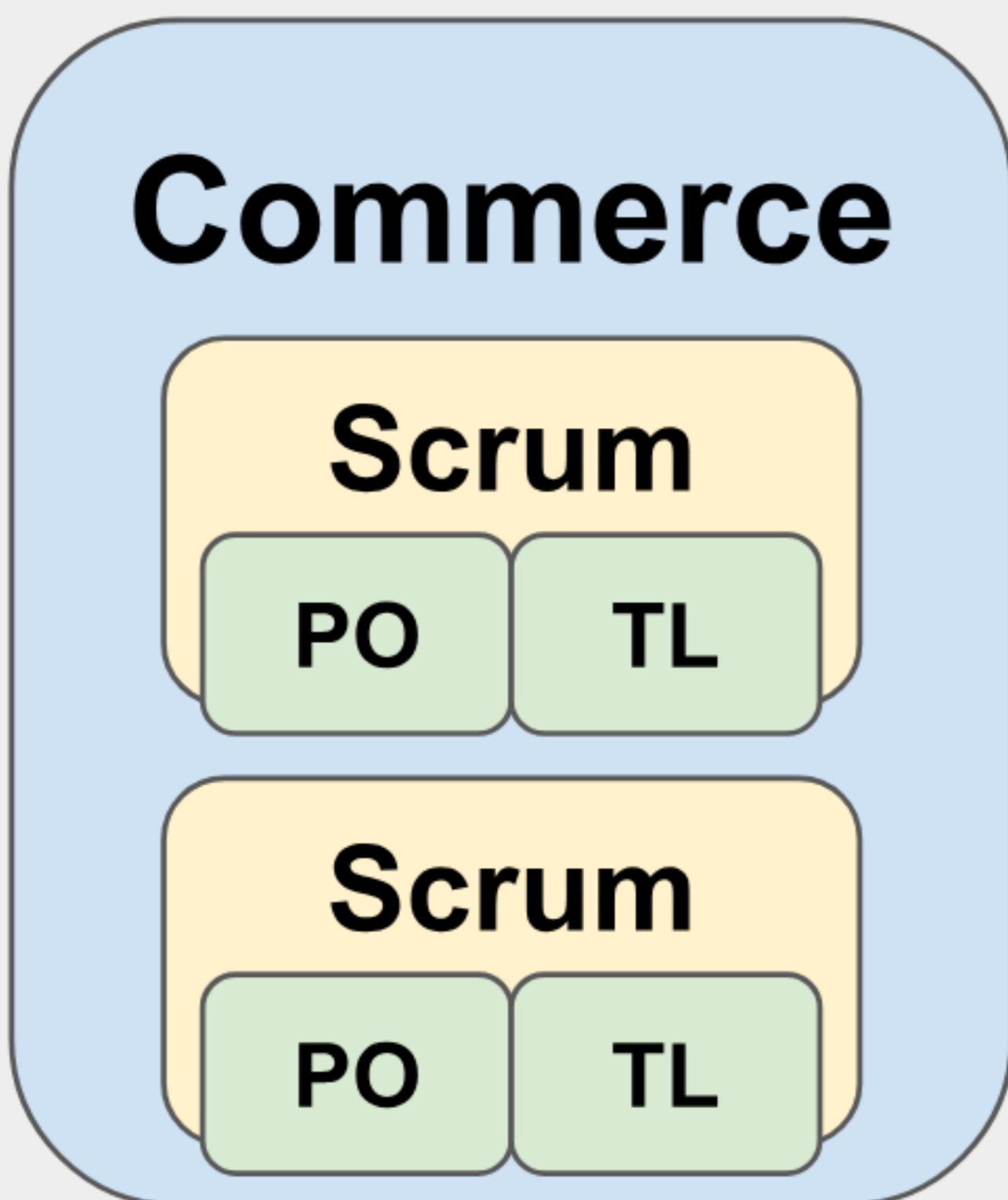


# Background

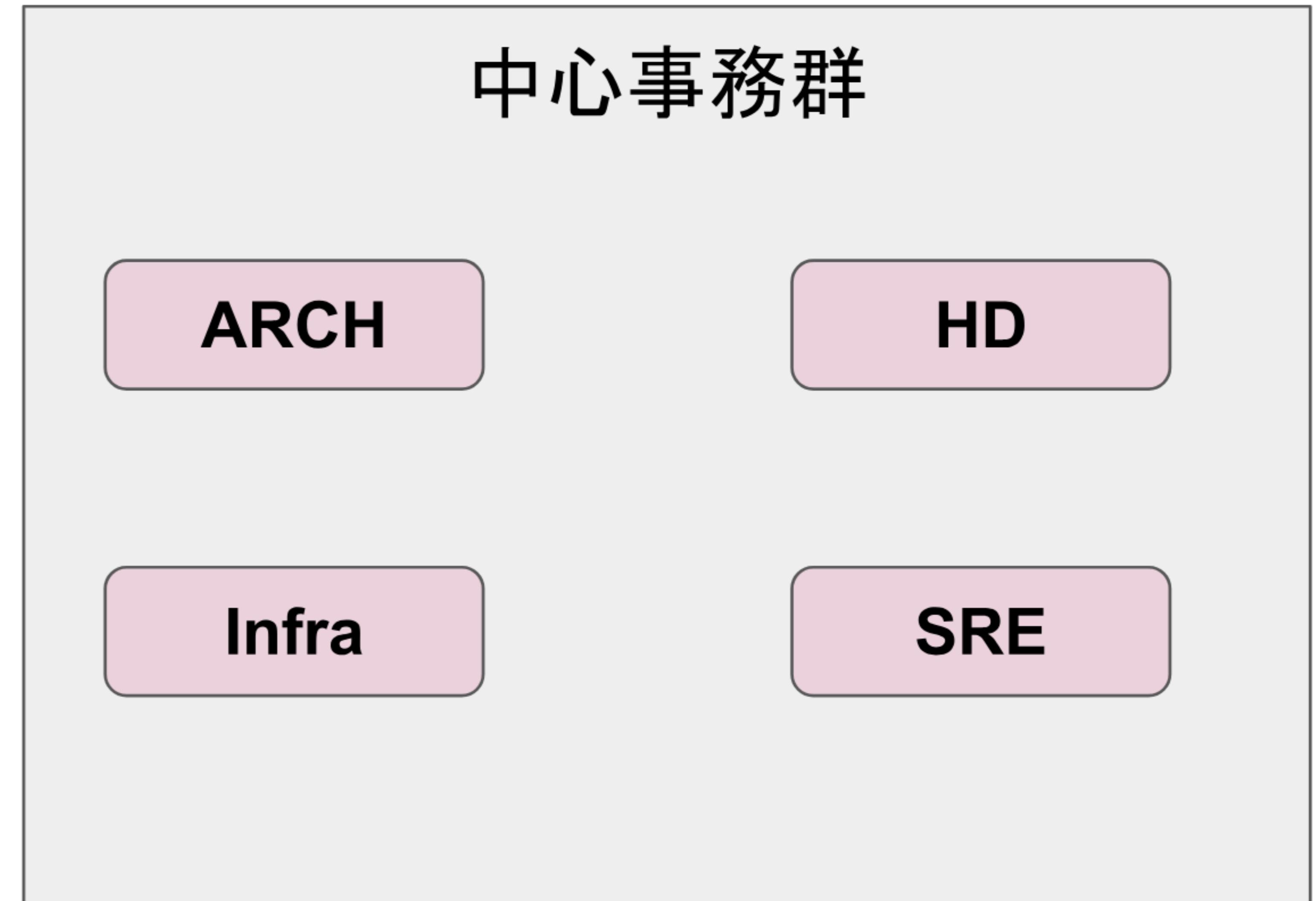
10 年 : 5 -> 300 人 -> ?

## ONE TEAM - 最終決策

### 產品團隊



### 中心事務群



# 過去的問題是什麼

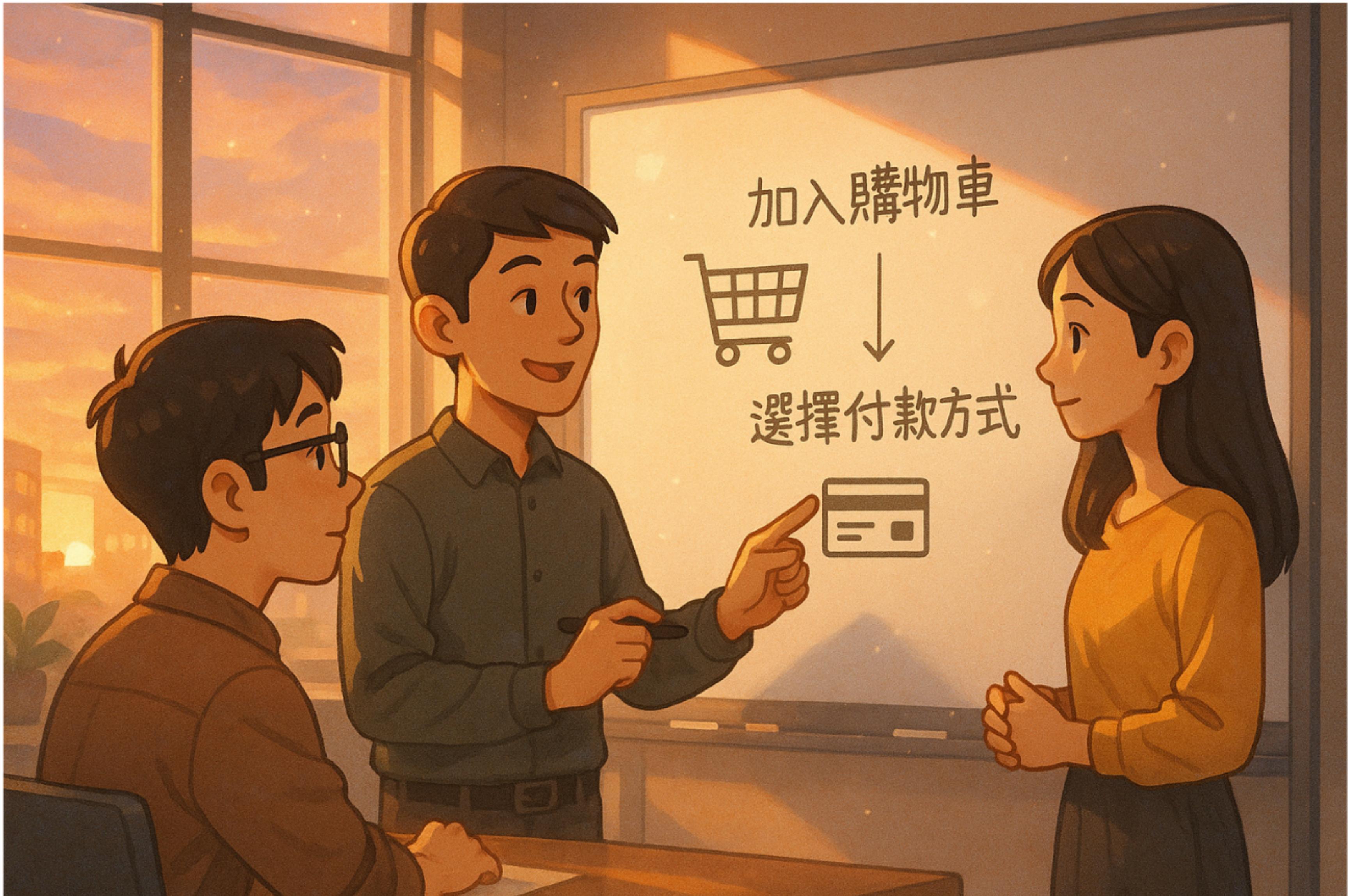
91APP

新專案 => PO + RD + Slides

高效、反應快

知識存在人腦中，難以擴展

期待找到 AI 時代的開發方法



Power by 李慕約 365 AI 訂閱計畫 #018

1. 2024Q4 開始推動 AI 生產力提升
2. 目標: 效率 **Scale-Up**(速度 + 品質)
3. 策略
  - a. 各職能(PO / RD / QA)各自進行「單點突破」
  - b. AI App + 流程串接整合

## 1. 成果

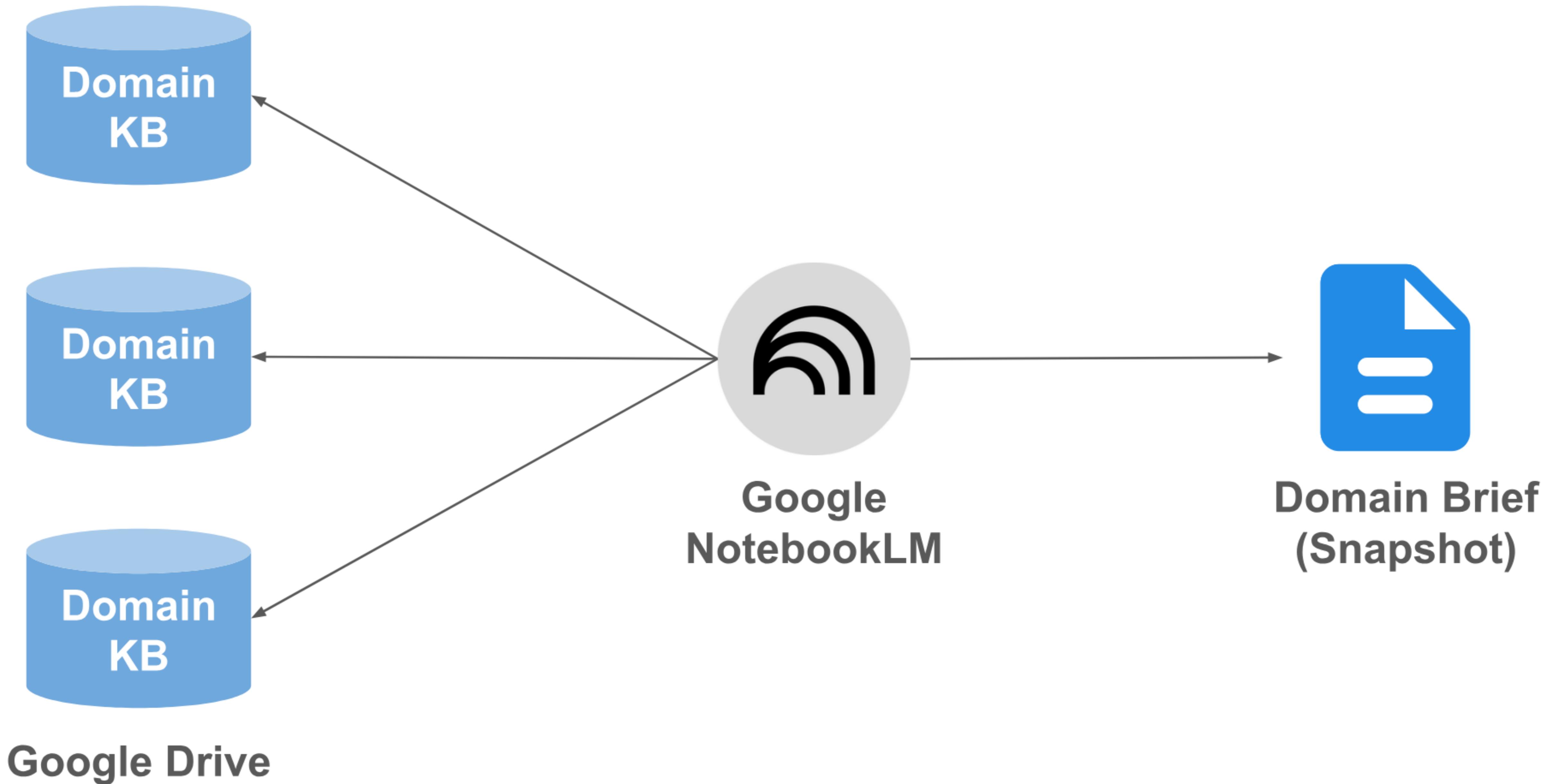
- a. 效率提升: 導入團隊交付效率提升 30%
- b. 系統穩定: 異常數量 YoY 持平

## 2. 關鍵點

- a. AI 做到 80 分
- b. AI 可讀的格式 (文件、程式碼)
- c. 傳承: Prompt / Doc Template -> AI App

# PO: 知識提取產生 Domain Belief

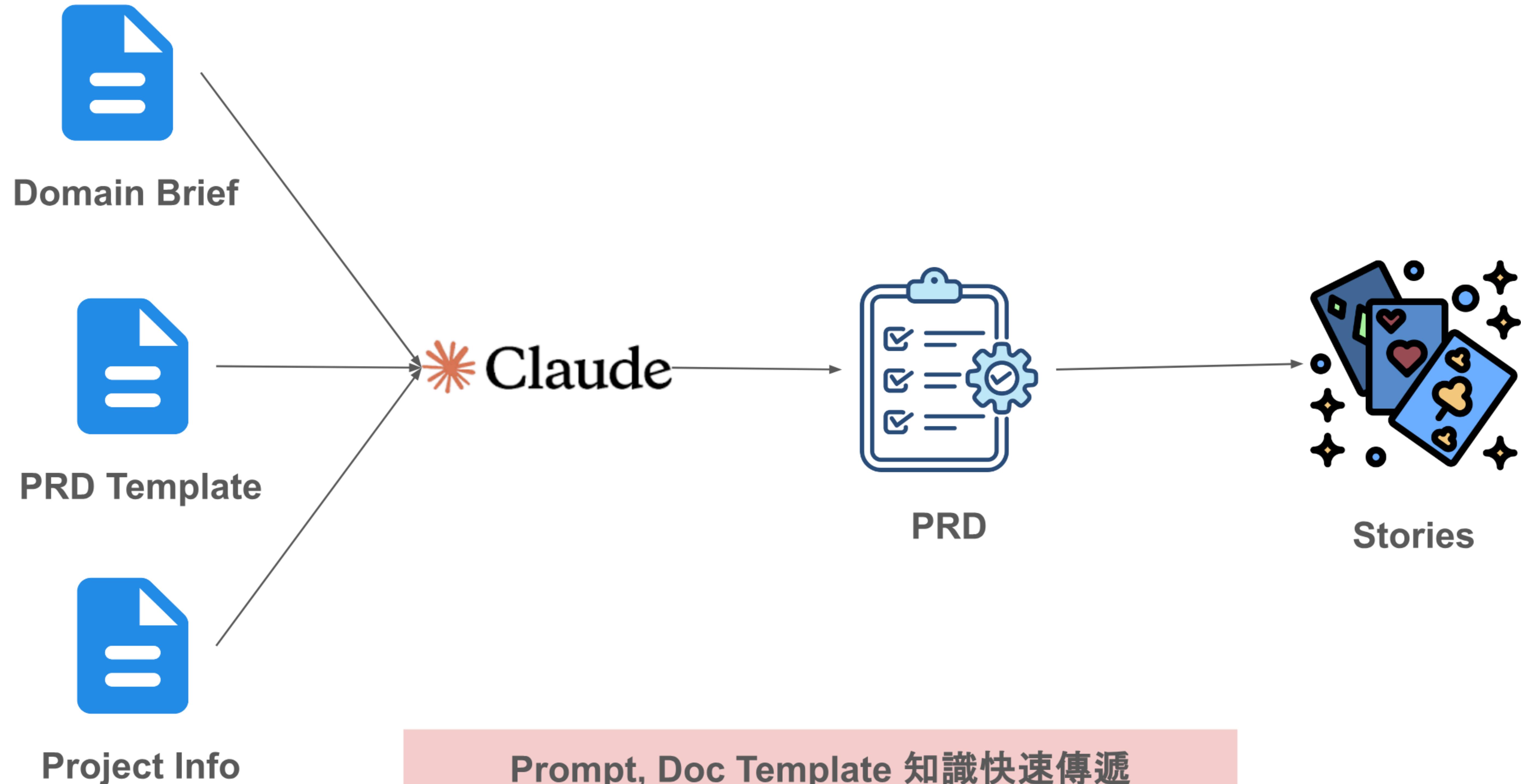
91APP



專案、教育訓練，省約 40% 時間

# 從 Domain Brief 產生 PRD (Product Requirement Doc)

91APP



# Project Template

-  **需求定義**: 釐清需求內容與背景
-  **現況變化**: 目前狀況與預期差異
-  **驗收準則**: 定義交付完成的標準
-  **開發規劃**: 實作方向與範圍提示

# PRD Template 結構

## 一、專案說明

1. 專案背景
2. 專案目標
3. 主要功能
4. 專案上線時程
5. 預期成效

## 二、功能調研

1. 產品現狀調研
2. 競品分析調研
3. 產品初步設計流程

## 三、專案範圍

1. 關鍵流程
2. 關鍵資料流

## 四、功能需求說明

👉 PRD - 功能需求說明

## 五、技術架構說明

👉 PRD - 技術架構說明

## 六、團隊與時程規劃

1. 團隊組成
2. 時程表

## 七、附錄

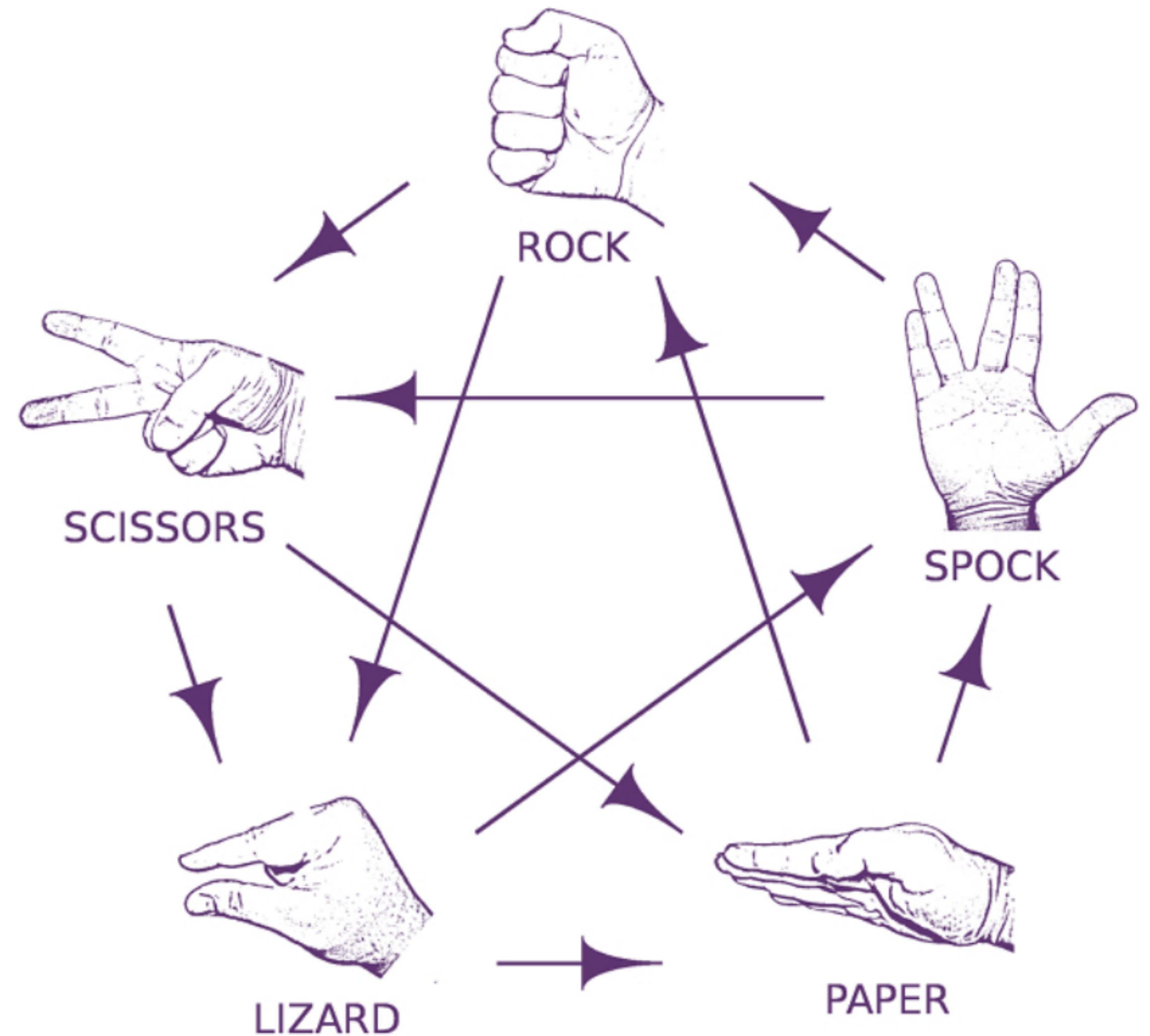
1. 相關文件

## 八、變更記錄

# 估時: 寫個剪刀、石頭、布、蜥蜴和斯波克程式

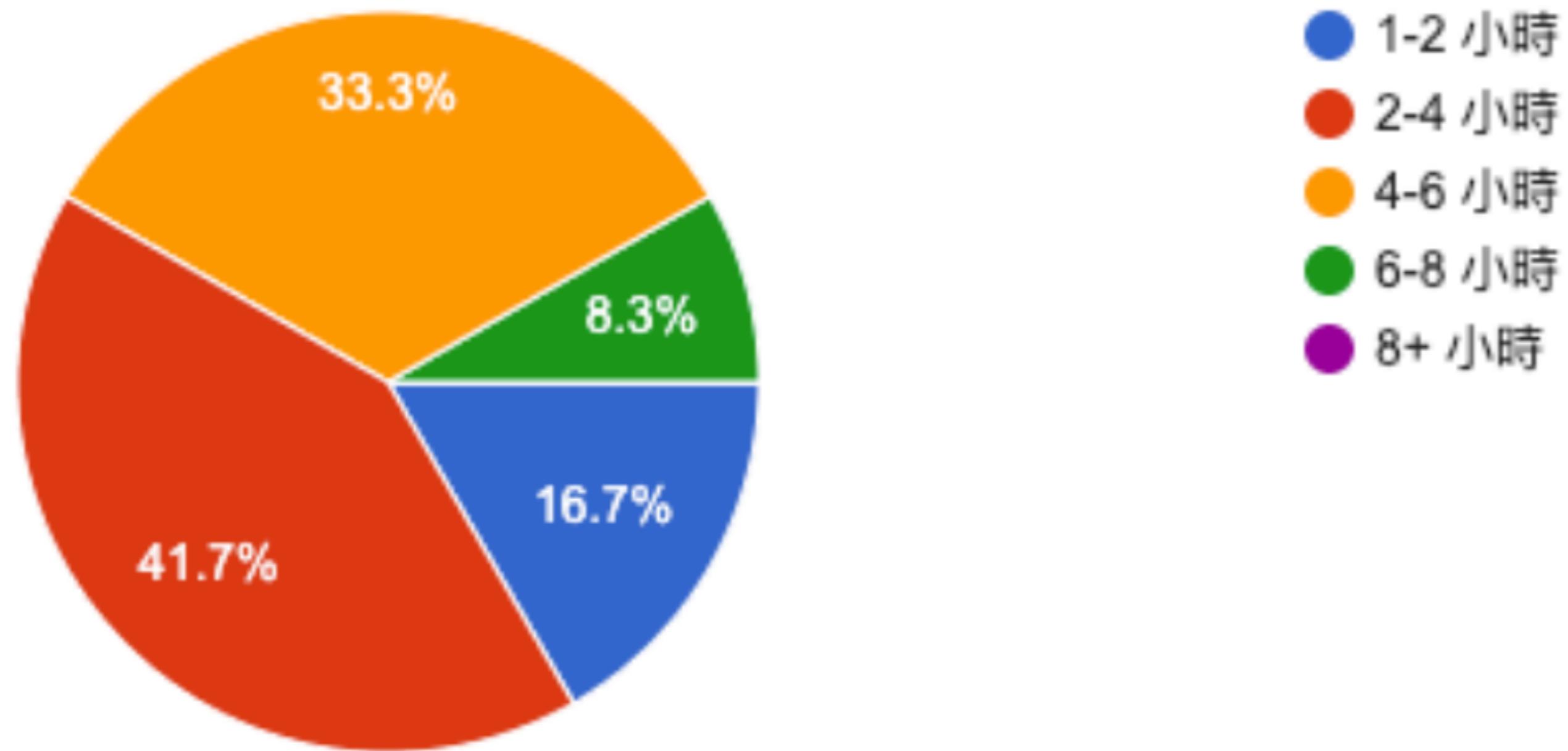
91APP

1. 在 Console 上和電腦猜拳
  - a. 你輸入出 剪刀、石頭、布、蜥蜴和斯波克
  - b. 電腦隨機出，判斷輸贏
2. 加入單元測試，覆蓋率 100%
3. 添加 RESTful API
4. 添加 REST API Spec
5. 添加完整的程式碼註解
6. 透過 Lint 工具，將語法修改到 100 分



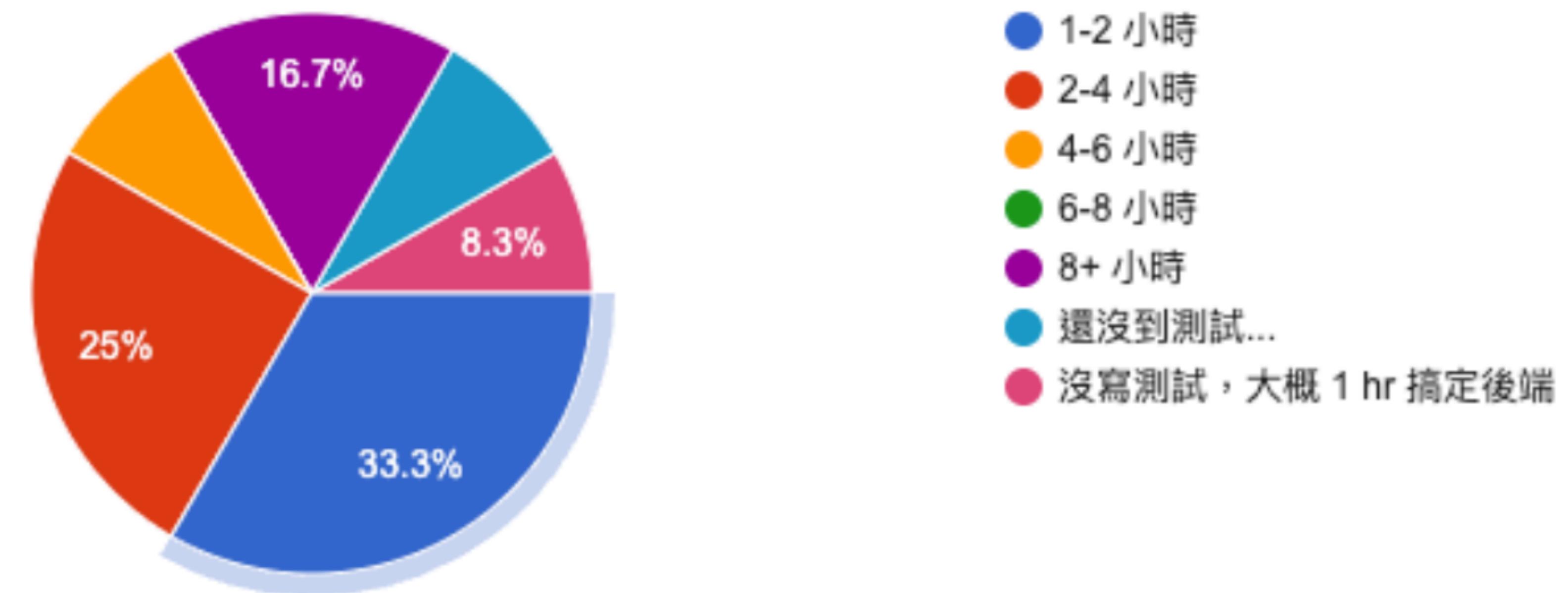
## Before:

- 16%, 2 hours
- 41%, 4 hours
- 33%, 6 hours
- 8%, 8 hours



## After:

- 41%, 2 hours
- 25%, 4 hours
- 8%, 6 hours
- 20%, None



## 1. 開發者是否熟悉該程式語言、領域知識

## 2. Prompt Engineering

有沒有什麼體悟是要和大家分享 - 譬如說有卡住的地方、那邊很驚艷等都可以。如果沒有請寫【無】  
(可以是某個 slack 鏈接，或者某一份簡報，或者某個 google doc 鏈接)

12 responses

比預想中快的地方：讓 AI 隨便幫我挑套件後實際可以順利運作的速度(除了測試的 Mock)、前端不講究 UI 美觀的話真的是超級快

卡住的地方：跟語言框架特性都不熟悉，測試的 Mock 行為都不如預期卡了一陣子 QQ

使用一個完全不會的語言，所以開發前的猜測預估的時間較久。

實際開發下去完全依靠 Prompt，(從 Go 的環境建置、需要的 library 等等全部都依賴 Copilot Chat 產出)，人為修改的部分不到 10 行？

Taskfile & Readme 也是 Copilot 產的

我自己先從設計文件開始寫，透過 Copilot 了解了這些資訊，AI 產 Code 時會比較有自信，比較知道發生什麼事，

但我自己如果在寫 NodeJS 的時候，用 Code Copilot 時的效率會比較好，因為已經知道大致上的語法邏輯，所以這次比較大的感受是每種 AI 工具在應用上的選擇，熟悉的語言用 Code Copilot 可以加速開發，不熟悉的工具可以靠 ChatGPT 說明引導，可以更快上手。

- 專案規模**: 共 20,000 行程式碼, 其中 2,000 行經修改
- 團隊成員**: 1 PO、1 TL、1 RD
- 成效**: 標準化提示語提升文件清晰度, 加速 AI 實作效率
- 挑戰**: 領域知識轉移複雜, AI 難以處理過度依賴人為共識的情境

	Requirements Analysis	System Design	Implementation
Actual Time / Planned Time	4 / 5 (hours), 20%	3 / 4 (hours), 25%	24 / 33 (hours), 30%+
Method	AI identifies corner cases and improves quality, accelerates flowchart generation		Store documents in the repository. TL can focus without Interruption

# SPEC 原則

- S – Specific: 問題明確、避免模糊
- P – Precise: 語言精確、不含糊
- E – Explain: 補足背景與邏輯
- C – Clarify: 列出限制與要求

from 91APP iOS Team Lead Justin

## # ■ S - Specific(明確定義問題)

請實作一支「加入購物車」的 RESTful API，讓前端使用者可透過 POST 請求將指定商品加入目前使用者的購物車中。

API 路徑預定為：`POST /api/cart/add`

## # P - Precise (精準描述細節)

請實作以下行為：

- 接收下列 JSON 參數：

```
{  
  "productId": "string",  
  "quantity": "int",  
  "skuid": "string"  
}
```

驗證：

- `productId` 與 `skuid` 必須存在資料庫
- `quantity`  $\geq 1$ , 且不超過可販售庫存數

商業邏輯：

- 若購物車中已存在相同商品與 SKU, 請累加數量
- 若為新商品, 新增購物車項目

回傳格式：

```
{  
  "success": true,  
  "cartItemId": "string"  
}
```

## ■ E - Explain (提供背景與邏輯)

此 API 將用於前台商品頁的「加入購物車」按鈕。購物車服務採用分散式快取架構，請以我們的 Redis-based Cart Service 操作購物車資料結構。

## ■ C - Clarify (列出條件與限制)

- 本次不需處理「促銷活動」與「優惠券」計算邏輯
- 不需實作驗證登入權限，請假設已取得 `userId`
- 若商品資料異常(找不到 SKU 或數量不足)，請回傳：  
`{ "success": false, "error": "INVALID_PRODUCT" }`
- 實作請依照 91APP 現行 coding style，使用 C# + .NET Core
- 可參考 `AddToCartService.cs` 作為結構參考

1. 簡介和目標 (Introduction and Goals): 描述系統目標、說明各階功能對應的 FR (Functional Requirements) , NFR (Non-Functional Requirements)
2. 上下文和範圍 (System Scope and Context): 對外、對內串接邏輯
  - a. C4 Model System Context diagram 說明系統層級串接邏輯
  - b. C4 Model Container diagram, 系統內各元件串接狀況
3. 建構塊視圖 (Building Block View): 展示系統的主要組件及其關係。
  - a. Swagger API Spec / DB Schema : MSSQL Create as Scripts or Model
4. 運行時視圖 (Runtime View): Sequence Diagram 說明各 User Story 運行時
  - a. 系統如何運作, 例如: 建立、更新、停用、註銷 API Key/Token 流程
5. 設計決策 (Architectural Decisions): 記錄重要的架構決策及其理由。
6. 詞彙表 (Glossary): 定義重要術語和概念。

# RD: Doc in Project Structure

91APP

```
docs/
├── README.md          # Overview of the project and documentation
└── requirements/
    ├── README.md      # Overview of the requirements documents
    └── user_stories.md # User stories document
└── design/
    ├── README.md      # Overview of the design documents
    ├── system_overview.md # High-level overview of the system
    ├── architectural_strategies.md # Architectural strategies
    ├── component_design.md # Component descriptions
    ├── database_design.md # Database schema and relationships
    ├── ui_design.md      # Mockups, wireframes, and UI descriptions
    └── error_handling_and_logging.md # Error handling and logging strategies
└── test_plan.md        # Testing strategy
└── deployment_plan.md # Deployment process
└── appendices/
    └── README.md      # Appendices overview
```

 Copy code

# Pilot Project, PO writes PRD first

91APP

內部 API 管理後台

## Introduction and Goals

## 1.1 Requirements Overview



### 1.1.1 Functional Requirements

- FR 1. 使用者帳號及權限

- 

- #### • FR 2. 支援不同市場及環境

- ### ◦ FR 2-1. 市場(Market)

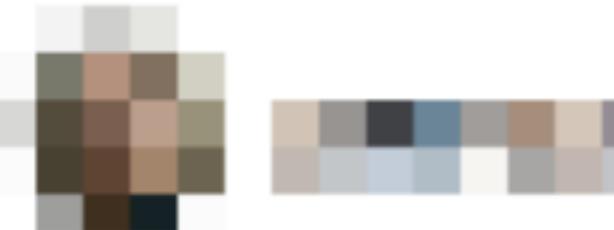
以安全性、效能、可維護性、可靠性為主，重新論述，必須要有具體數字及範例



當設計內部 API 管理後台系統時，以下是針對安全性、效能、可維護性和可靠性的非功能性需求，並包含具體數字及範例：

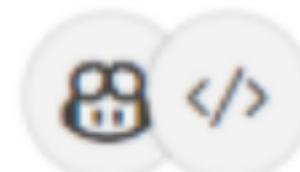
## 1. 安全性 (Security)

- 資料加密：所有傳輸中的 API Key 和敏感資料必須使用 TLS 1.2 或以上版本進行加密，確保資料在網絡傳輸過程中不被竊取或篡改。例如，對 API Key 的傳輸過程應使用 256-bit AES 加密。
- 權限管理：系統應實施基於角色的訪問控制 (RBAC)，確保只有擁有適當權限的使用者可以創建、更新或刪除 API Key。每個操作必須經過身份驗證和授權，並且應記錄所有操作的詳細日誌，以備審計。舉例來說，只有擁有 "Admin" 角色的使用者才能進行 API Key 的刪除操作。
- 安全審計：系統應保存至少 90 天的操作日誌，記錄所有關鍵操作和事件，包括 API Key 的創建、更新和刪除等，並提供詳細的錯誤報告以支持安全審計。



@workspace

針對 [#file:01\\_introduction\\_and\\_goals.md](#) 裡面提到的 P2 需求 該如何開始開發



GitHub Copilot used @workspace

Used 2 references >

根據 [#file:01\\_introduction\\_and\\_goals.md](#) 文件中的 P2 需求，以下是開始開發的步驟：

1. 本後台能管理 Connect Cloud V2 API (不含 DataSync, Payment-Store, Payment-Agency): 首先，您需要擴展後台以支援 Connect  

2. 能使用 Email 傳送金鑰壓縮檔及 SMS 傳送金鑰解壓縮密碼: 您需要實現一個功能，當金鑰生成時，將金鑰壓縮並通過電子郵件發送。同  

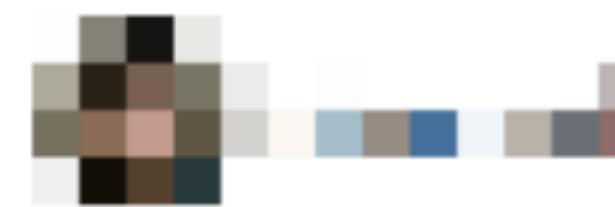
3. 發出的 Email Title 及內文新增客戶資訊: 更新電子郵件模板以包含更多的客戶信息。這可能涉及更新電子郵件生成邏輯以從數據庫中獲取更多的客戶信息。

請注意，這些步驟可能需要根據您的具體代碼庫和架構進行調整。

明確定義範疇: AI 可快速分析調整項目

效率提升: 原需 Jr. 工程師與 TL 花 30 分鐘, AI 僅需 1 分鐘完成

知識應用: 善用累積的文件與知識, 加速決策與執行



請參考 [#file:05\\_building\\_block\\_view.md](#) [#file:06\\_runtime\\_view.md](#) [#file:08\\_concepts.md](#) 該如何調整 CreateAuthority 和 RegenerateAuthority

GitHub Copilot

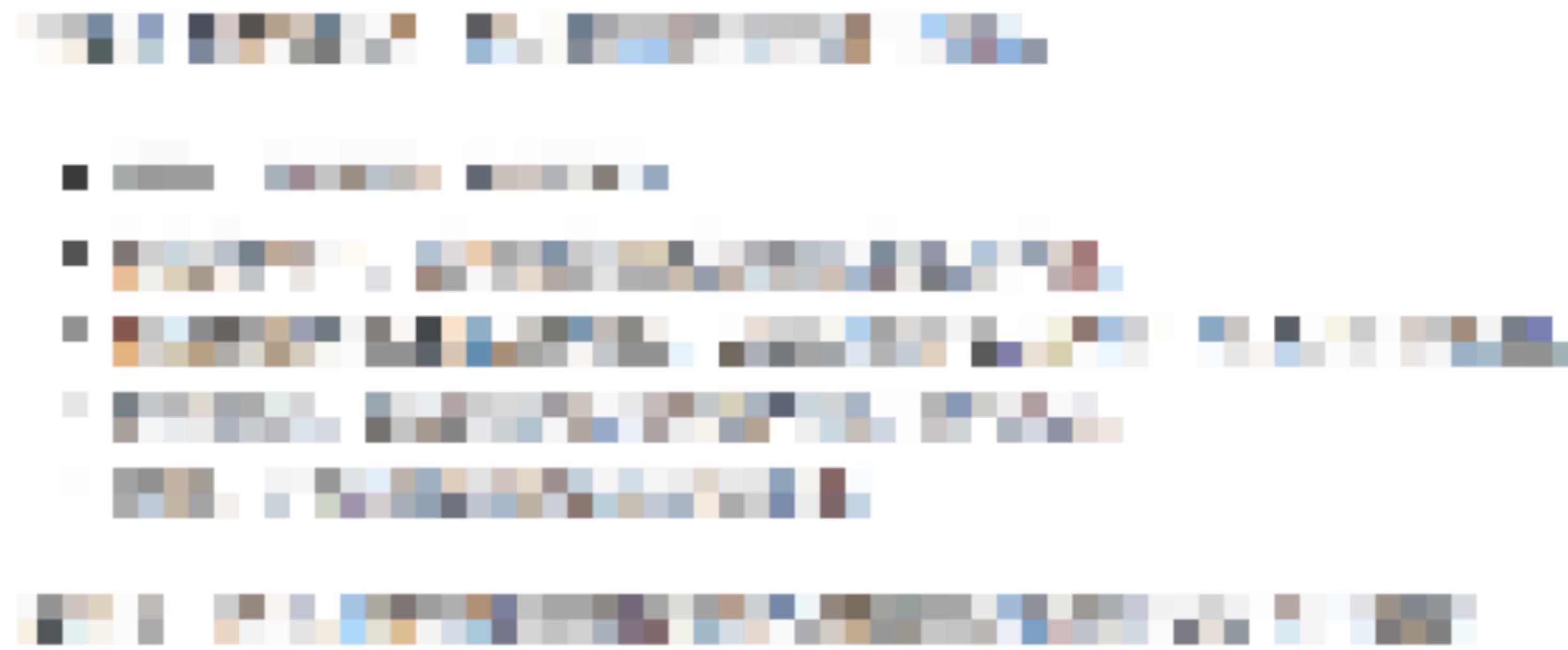
Used 3 references >

根據您提供的文件, 以下是我對於調整 CreateAuthority 和 RegenerateAuthority 的建議:

1. CreateAuthority:



2. RegenerateAuthority:



## ■ 有規劃地逐步給指令，不求一步到位

- 不是一股腦丟給 AI，而是先思考好流程，或與 AI 規劃好執行順序後，逐步下達指令。

## ■ 明確指定「改哪裡」與「產出落點」

- 例如：「請在 `CartService.cs` 中新增 `AddToCart` 方法，輸入為 `productId` 與 `quantity`」。

## ■ 偏掉就停，不要浪費時間硬改

- 如果產出方向錯誤，直接重來或手動修正會更有效率。

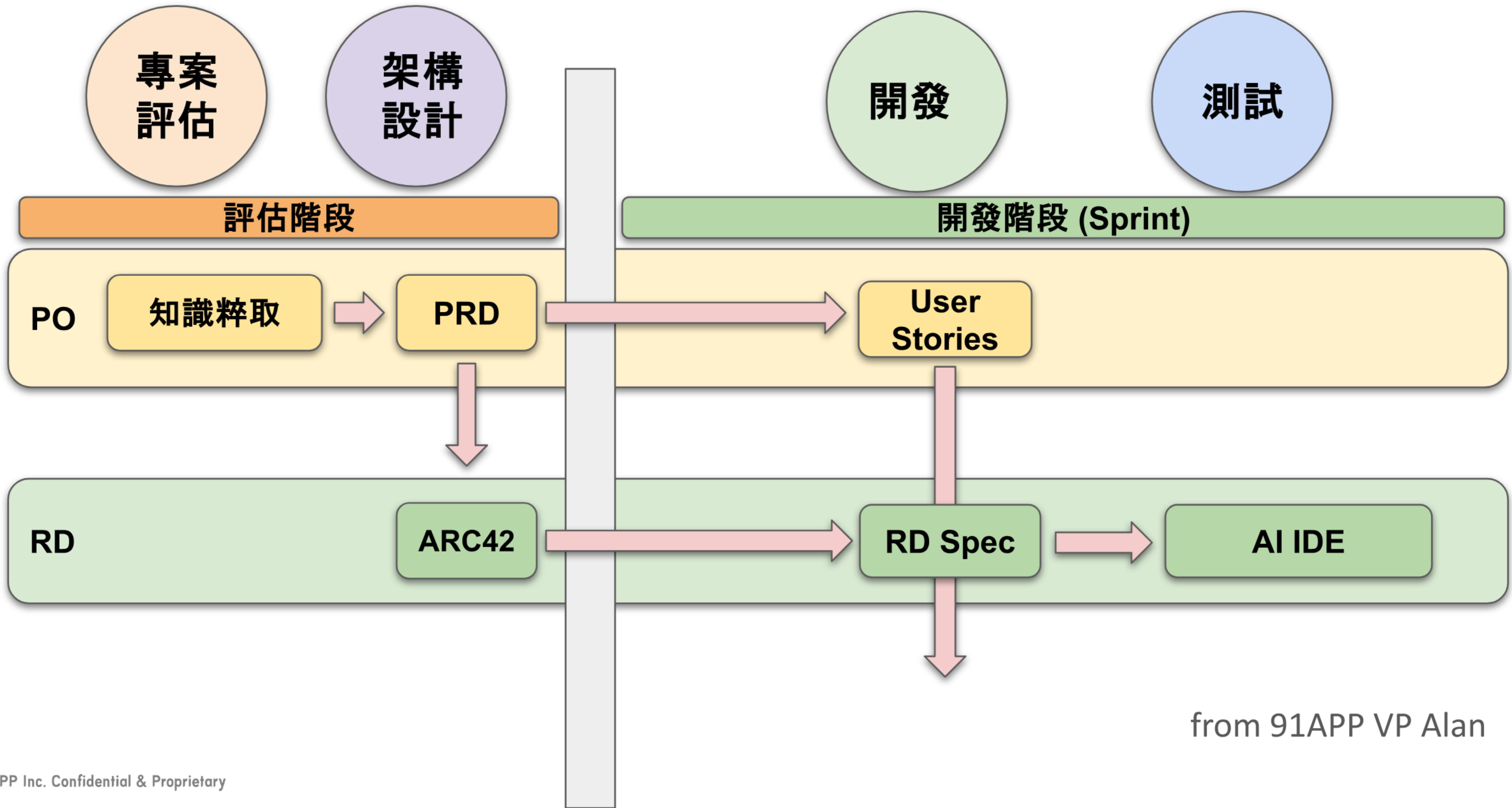
## ■ 設定期望：80 分為目標，非 100 分

- AI 是幫你加速，不是幫你完美收尾，有意識地安排 Review、Refactor 時間

*from 91APP Payments Sr.Manager Ken*

# AI-Driven Software Development Lifecycle

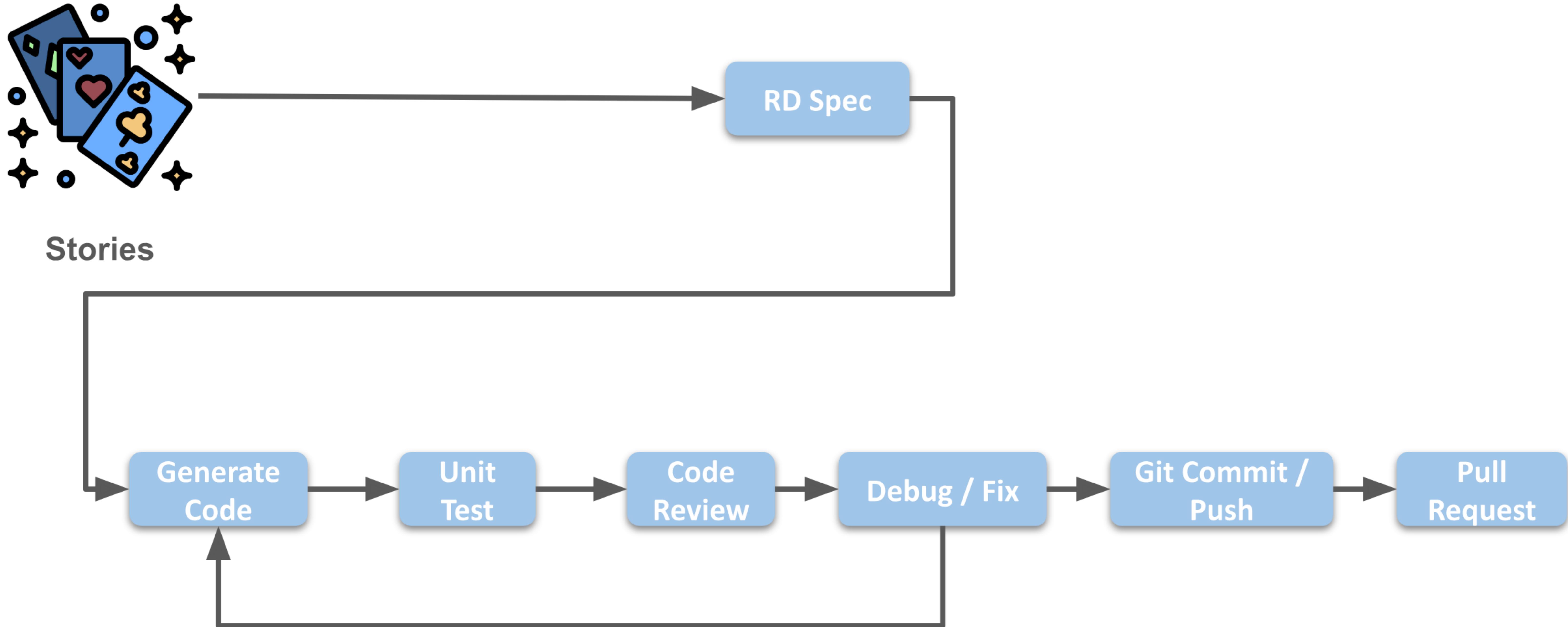
91APP



from 91APP VP Alan

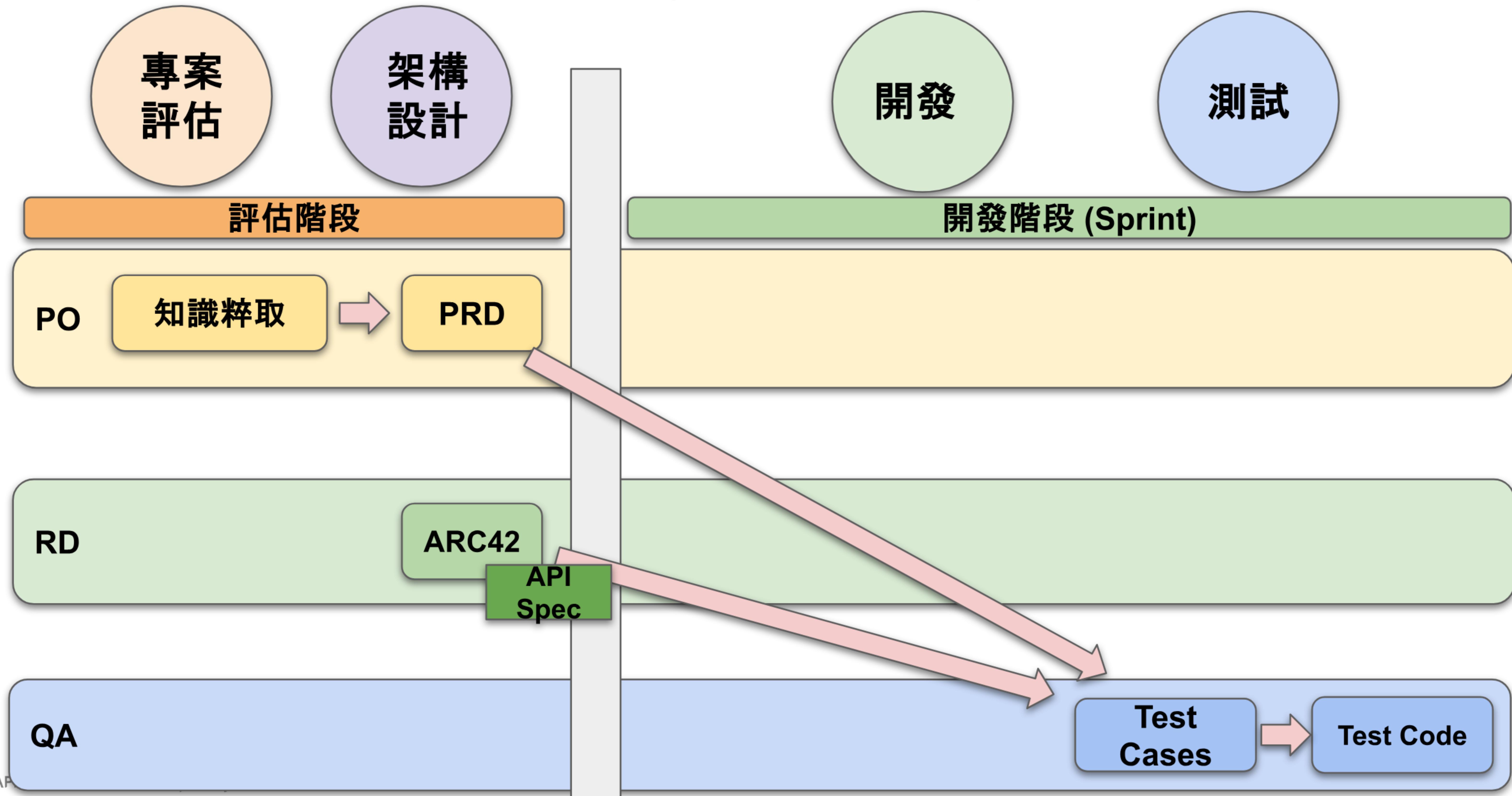
# AI-Driven Development - 開發階段(Sprint)

91APP



# AI-Driven Software Development Lifecycle

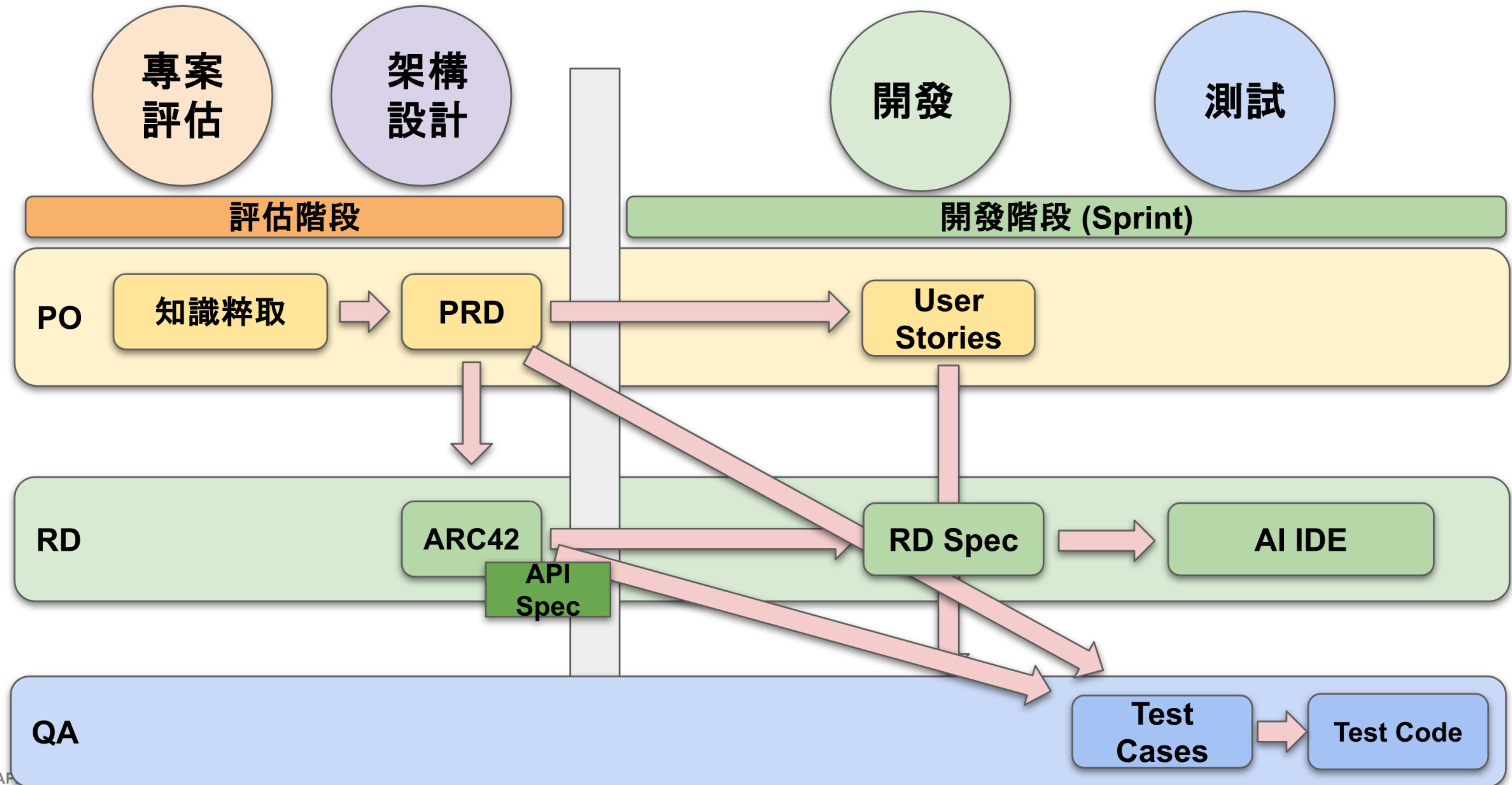
91APP



- Swagger API Spec 提供資料欄位、型別與結構
- PRD 測試情境提供完成實際業務操作、驗證邏輯

# AI-Driven Software Development Lifecycle

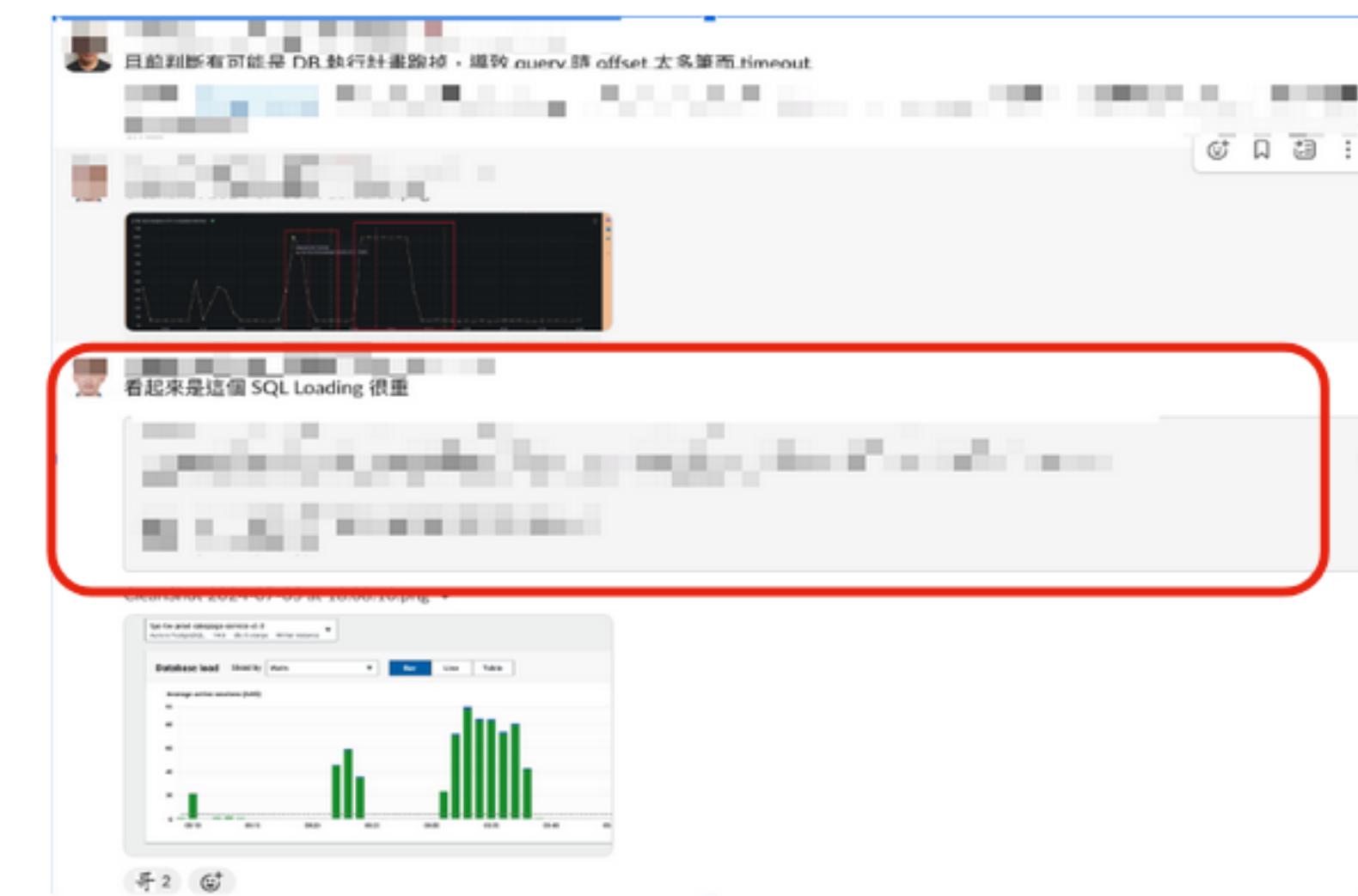
91APP



# Abnormality Review

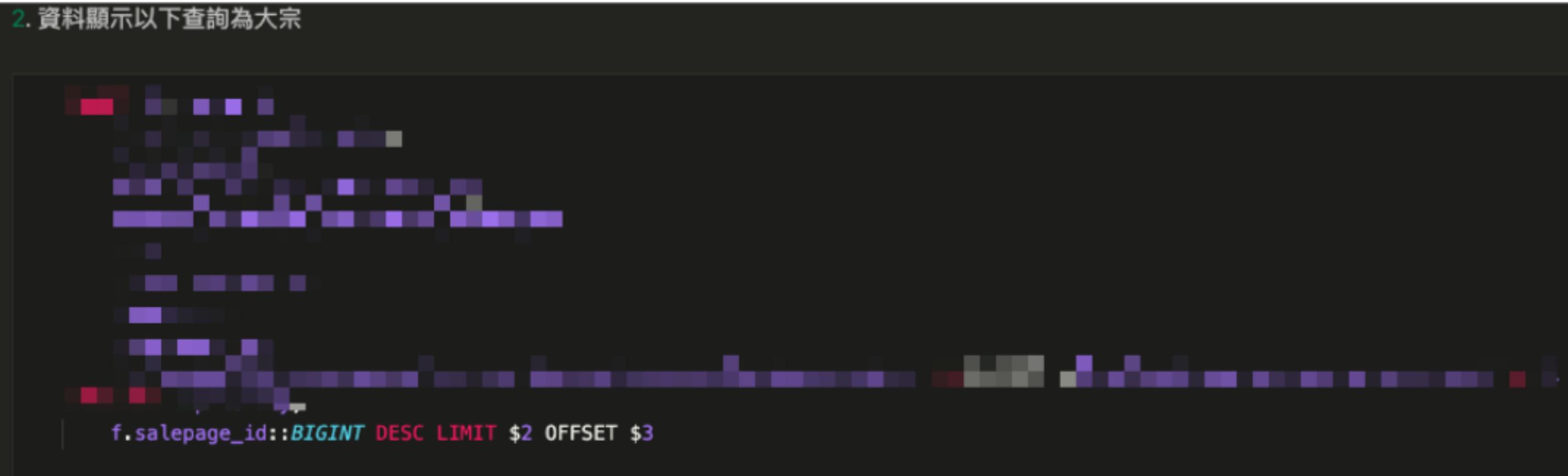
91APP

- 透過監控系統偵測到異常查詢
- 多方嘗試後，發現是一個內部系統發生異常



審查會中已達初步共識，直至 DBA 提出進一步分析才出現轉變

2. 資料顯示以下查詢為大宗



```
f.salepage_id::BIGINT DESC LIMIT $2 OFFSET $3
```

3. 上述的查詢中,salepage\_id為VARCHAR型態,但因排序時將其轉型為BIGINT,這樣的行為十分耗用CPU,與第1點觀察到的結果相近

-- DB

1. 由Vacuum資訊來看,Dead tuple的堆積似乎也還有
2. 由於沒有查詢權限,所以無法測試以上的查詢是否有正確的使用索引

# 然後...我突然恍然大悟做了下面事情...

你是一個 PostgreSQL DBA，請協助我找出下列這段 query 的效能問題，這段 query 造成

CPU 100%

SQL Query

這段 SQL query 造成 CPU 100% 的問題可能來自於幾個原因，特別是在排序和過濾大量資料的時候。以下是一些可能的效能問題和解決建議：

## 1. 排序 (ORDER BY) 和類型轉換 (CAST):

- Query 中的 ORDER BY f.priority, f.salepage\_id::BIGINT DESC 可能會造成功能瓶頸。特別是 salepage\_id 被轉換為 BIGINT，這樣的類型轉換會影響索引的使用效率。如果 salepage\_id 的資料類型不是 BIGINT，PostgreSQL 可能無法使用已有的索引來優化排序操作。
- 解決建議：考慮是否能避免類型轉換，或為 salepage\_id::BIGINT 創建一個對應的索引。如果 salepage\_id 的類型和所需的排序類型不一致，可以考慮先處理資料類型，再進行查詢。

## 2. 索引的選擇:

- 雖然表上已經有多個索引，Query 的效能仍可能受到索引選擇的影響。這裡可能需要進一步確認使用的是哪個索引，並評估索引的效果。
- 解決建議：檢查 Query 的執行計劃 (EXPLAIN ANALYZE) 以確認 PostgreSQL 如何使用這些索引。如果查詢過程中並沒有使用適當的索引，可以考慮新增或修改現有的索引，例如在 priority 和 salepage\_id 上創建覆蓋索引。

？

1. **AI 發揮效能的關鍵**: 明確目標 + 拆解為小任務 → 發揮最佳效率
2. **挑戰在於拆解**: 拆解任務需要知識與經驗，而這些難以有效傳遞與共享
3. **常見對話場景**: 「你那個提示語怎麼寫的？可以分享一下嗎？」
4. **核心反思問題**: 是 AI 的限制，還是 我還沒正確使用 AI？



## DeepLearning 的 ChatGPT 提示詞工程教學

f0x Hsiao  
9 部影片 觀看次數：10,238次 上次更新日期：2023..

全部播放 隨機播放

在這門課程中，您將學會到什麼。

您將學習如何使用大型語言模型 (LLM) 快速建立新的強大應用程式。使用 OpenAI API，您將能夠快速建立能夠學習創新並以成本高昂、高度技術性或以前根本不可能的方式創造價值的能力。

由Isa Fulford (OpenAI) 和Andrew Ng (DeepLearning.AI) 教授的這門短課程將描述 LLMs的工作原理，提供提示詞工程的最佳實踐，並展示LLM API如何在各種任務的應用中使用，包括：

- Summarizing (e.g., summarizing user reviews for brevity)
- 文本摘要（例如，為了簡潔而總結用戶評論）
- Inferring (e.g., sentiment classification, topic .....

- 1 紙上談話：給開發者的 ChatGPT 提示詞工程 01. 課程介紹  
f0x Hsiao · 觀看次數：9632次 · 1 年前 6:28
- 2 紙上談話：給開發者的 ChatGPT 提示詞工程 02. 使用準則  
f0x Hsiao · 觀看次數：5515次 · 1 年前 17:36
- 3 紙上談話：給開發者的 ChatGPT 提示詞工程 03. 迭代優化  
f0x Hsiao · 觀看次數：1575次 · 1 年前 13:19
- 4 紙上談話：給開發者的 ChatGPT 提示詞工程 04. 文本摘要  
f0x Hsiao · 觀看次數：1101次 · 1 年前 7:34
- 5 紙上談話：給開發者的 ChatGPT 提示詞工程 05. 推理能力  
f0x Hsiao · 觀看次數：1294次 · 1 年前 12:01
- 6 紙上談話：給開發者的 ChatGPT 提示詞工程 06. 文本轉換  
f0x Hsiao · 觀看次數：792次 · 1 年前 12:42
- 7 紙上談話：給開發者的 ChatGPT 提示詞工程 07. 文本擴寫  
f0x Hsiao · 觀看次數：667次 · 1 年前 6:49
- 8 紙上談話：給開發者的 ChatGPT 提示詞工程 08. 聊天機器人  
f0x Hsiao · 觀看次數：769次 · 1 年前 12:26

好的 Prompt 很重要，我能不能不要學？



Product Pricing Docs Blog

40.9k

Get Started

What's New Release v0.6.16 - Enhance LLM Application observability with LangSmith and Langfuse →

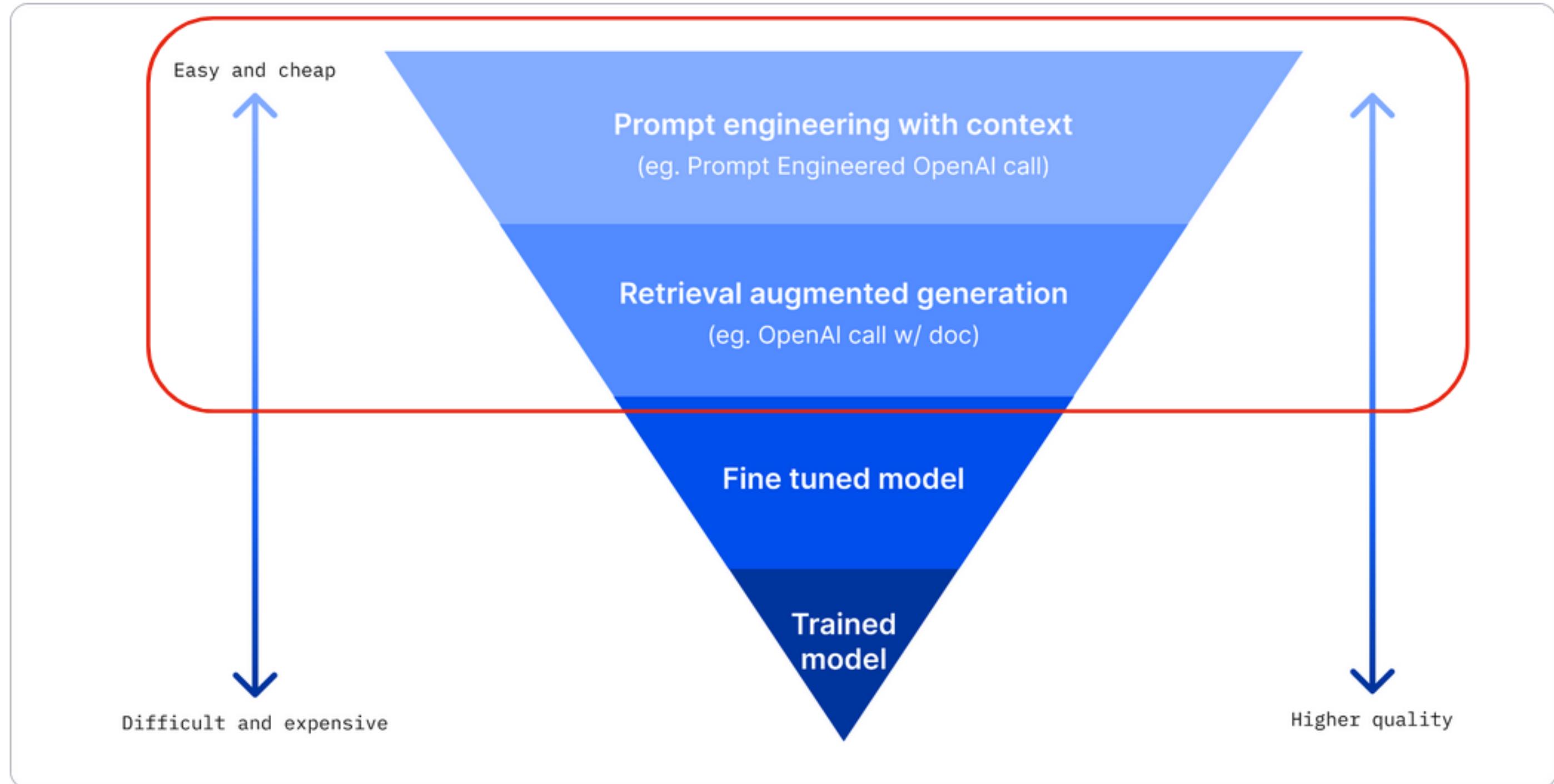
# The Innovation Engine for GenAI Applications

Dify is an open-source LLM app development platform. Orchestrate LLM apps from agents to complex AI workflows, with an RAG engine.

More production-ready than LangChain.

Get Started

GitHub





SQL Reviewer

CHATBOT

BASIC

Orchestrate

API Access

Logs &amp; Ann.

Monitoring

## Orchestrate

### Instructions ?

[Generate](#)**## 角色**

你是一個嚴謹的關聯式資料庫 DBA，會有很多人找你審核他們的 SQL Query。

**## 技能**

1. 大家會給你 SQL Query，請給出是否可以優化效能。
2. 大家會給你 SQL Query，請看是否符合公司規範。

**## 限制**

1. 回應繁體中文。
2. 如果沒有把握，請回答不知道，或要求使用者補足更多資訊。
3. 問和 SQL 不相關的問題，請不要回答。

190

## Prompt Engineering

### [\(x\)](#) Variables ?

[+ Add](#)

Variables allow users to introduce prompt words or opening remarks when filling out forms. You can try entering "{{input}}" in the prompt words.

[+ Add](#)[+ Add](#)[+ Add](#)

### Context

[Params](#)[+ Add](#)

DB Guideline.md...

29,578 Words · 2 Text Blocks

## KM: DBA Guideline, DB Schema

請幫我看下列這段 query 是否符合公司規範?

SELECT \* FROM SalePage



根據公司規範，這段查詢需要調整。如下是問題點與改進建議：



#### 問題點

1. 使用了 SELECT \*，這是不可接受的，因為規範要求只選擇需要的欄位。
2. 沒有使用 WITH(NOLOCK)。

#### 改進建議

請具體指定需要的欄位，並加上 WITH(NOLOCK) 來避免對線上資料庫的影響，但同時也要注意 WITH(NOLOCK) 的使用風險。

例如，如果你需要查詢 `id` 和 `name` 欄位，則查詢應該改為如下：

#### SQL

```
1 SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
2
3 SELECT id, name
4 FROM SalePage
5 WITH(NOLOCK);
```

請依實際需要選擇必要的欄位進行查詢。

# Oning 6 Topics

91APP

	部門	題目	KB	負責人
專家詢問				
內容審核				
表單提交				
表單處理				
文字處理				
流程自動化				

1. 專家詢問:AI Mentor
2. 表單提交:AI Cloud Resource Register Agent
3. 流程優化 & 表單提交:HD 工單流程優化
4. 文字處理:UX 自動翻譯
5. 內容申核:AI Code Review

# 案例一：專家詢問 AI Mentor

- 說明：新人教育訓練知識庫，協助新人快速進入狀況
- 目標：
  - 一位 JR 新人 on-board 省下 mentor
  - 一個月 5 個人天 (8 天變 3 天)
- 功能
  - 用魔法對付魔法，透過 NotebookLM 參考教育訓練文件
    - 再透過 Dify 包裝 AI Mentor
  - AI 已可檢索 100+ 份教育訓練文件 (PDF, Q&A)
  - 23 張試題，每張 30 題，近 700 題考題



歡迎進行測驗~請先輸入您的姓名

Kevin



你好，Kevin！現在你可以挑選一份要考的試卷，我們有以下兩個選擇：

1. .NET 非同步程式設計題目
2. 資訊安全題目

請告訴我你想要挑選哪一份試卷吧！

總分：80/100

專業評論：

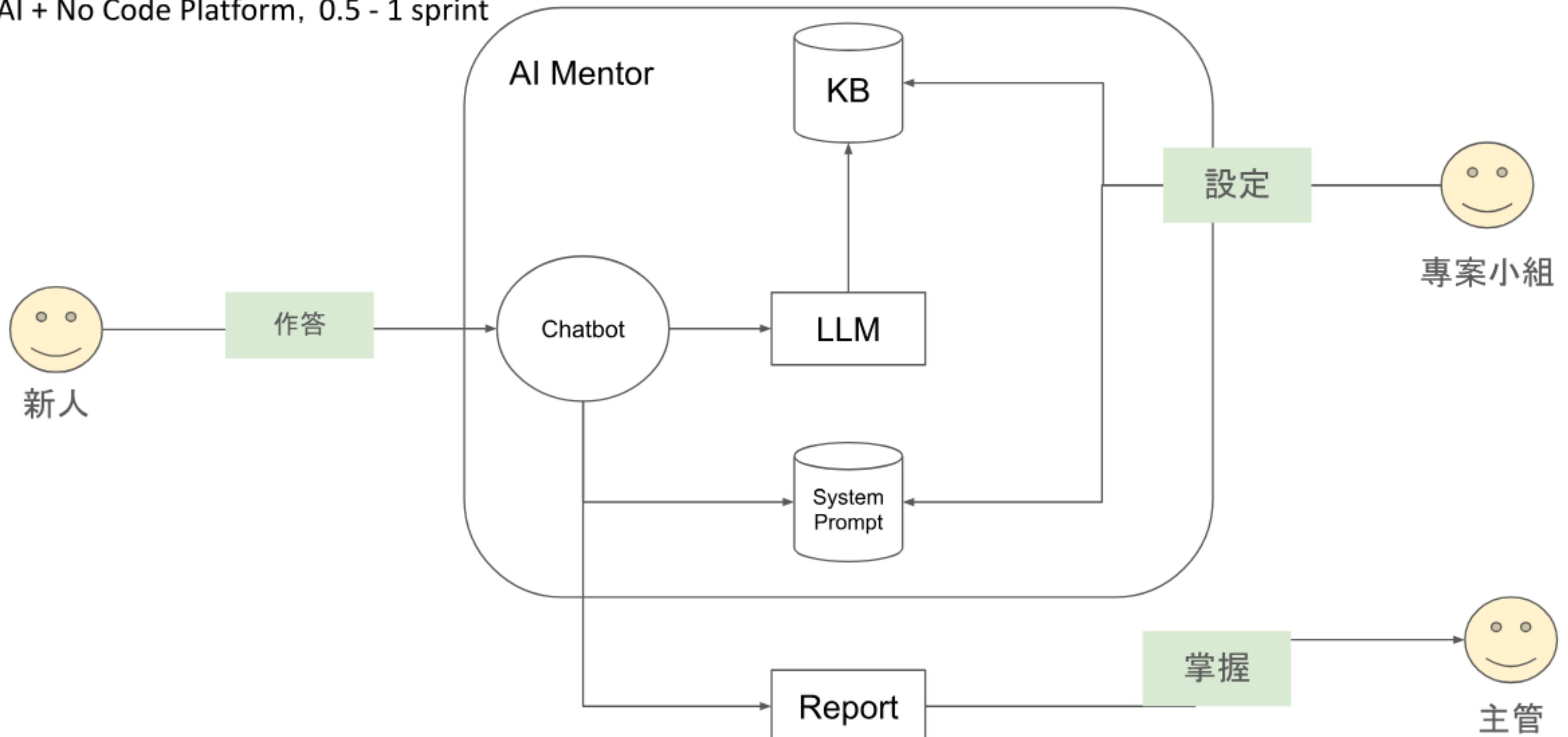
- **綜合評分：**你對於 .NET 非同步程式設計的基礎概念相當了解，在大多數題目中都給出了正確的答案，表現非常出色。
- **特別優點：**尤其在理解基礎概念如 `async`、`await` 及 `Task` 的使用上，你展現出了扎實的基礎。
- **建議改進：**可以更深入了解 `CancellationToken` 的應用場景，以及 `await` 關鍵字如何實際影響非同步作業。

如果你有任何問題或需要進一步的輔導，隨時可以告訴我！

# AI Mentor

過去流程 + 人，耗時，如果要開發 2 sprints 起跳

AI + No Code Platform, 0.5 - 1 sprint



# 案例二：

## AI Cloud Resource Register Agent

### 目標

- 自動化 Ticket 流程以提升準確性與效率
- 快速產出 Ticket、減少重複性作業
- 節省時間與成本

### Key Features & Benefits

- **AI 智能輔助：**
  - 自動填寫正確資訊、推薦最佳選項
  - 提供關鍵知識(如 AWS IAM Policy 產生)，降低使用門檻
- **流程優化：**
  - 支援多環境 Ticket 自動建立
  - 減少 RD 與 SRE 沟通成本，流程更簡化
- **效能提升：**
  - 成功率達 80%，每年節省 21 人天

除錯與預覽

AI Agent

Hello 工程師們，我是 Nine1 Kubernetes Platform 工單助手，您可以問我一些 Infrastructure 問題，或輸入 GO 開始建立工單

我是正式環境但內部使用的 Domain 應該要怎麼寫？ NKP Namespace 的規範是什麼？

QA 環境的規範有什麼 Go - 我要建立 NKP 資源工單

Go - 我要申請 NKP Service 權限工單

Go - 我要申請 NKP Service 權限工單 A

好的，請提供以下資訊以開始申請 NKP Service 權限工單：

1. 市場 (可多選) : TW/HK/MY
2. 環境 (可多選) : Prod/QA/qaN/PP
3. Project Name :
4. Namespace Name :
5. Service Name :
6. 需求團隊 :
7. 需求部門 :
8. 用途、調整項目 :

請填寫完畢後告訴我。

1 已使用 CreateIssue >

工單已成功建立！以下是相關資訊：

- 工單標題：調整 AWS IAM Role on NKP : eks-ap-northeast-1-tw-91app-io-v2 - prod-nkp-go
- 工單 ID : 23141
- 狀態 : New
- 分配給 : Archie Shih

Automatic  
Creation

請確認這些資訊是否正確。接下來，我會將這次的協助過程做個收尾，感謝您耐心的等待。

最後，我想詢問您的姓名 (可匿名)，以及此次服務的滿意度 1-10 分，給予幾分？

# 大家其實不知道怎麼開 ticket...

## 過去的挑戰

開發者若不了解以下資訊，常需面對 Ticket 被退回、與 SRE 多次來回討論：

- K8s Namespace 命名標準
- 91APP DNS 命名規範
- 91APP 帳號與市場對應表
- 市場與 AWS Region 對應關係
- AWS IAM Policy 編寫規則

## AI 協助解方

- AI 自動帶入對應資訊，處理 80% 的工作量
- 降低錯誤率與溝通成本，加快流程推進速度

市場：(TW/HK/MY)：  
環境：(Prod/QA/qaN/PP)：QA  
Project Name：Gengar  
Project Admin (AD account)：archieshih  
Service Name：Gengar  
namespace (By 市場)：qa-gengar  
Ingress/Domain Name (By 市場)：**不知道，幫我寫**  
用途 (QA/客戶使用走外網/內部使用走內網/內部使用走外網)：QA

已使用 getNKPConvention >  
已使用 verifyDNSConvention >

已經確認您的 namespace 符合規範，並且為您找到了正確的 Ingress/Domain Name。

EKS 相關資源

- EKS Cluster: [REDACTED]
- Project Name: Gengar
- Project Admin: archieshih
- Namespace: qa-gengar
- Ingress/Domain Name: [REDACTED]
- 用途：QA

請確認以上資訊是否正確。

# 矯正使用者答案

91APP



讓我們先檢查 Namespace Name 是否符合規範。根據 91APP 的 NKP 規範，Namespace 應遵循以下格式：{ENV}-{SERVICE\_NAME}。

在此情況下，我們期望的格式應為：prod-nkp-go。您所提供的名稱為：nkp-go，建議將其更改為 prod-nkp-go，以符合規範。

您是否同意這個更改？

# AI 撫平 RD 申請資源的先備知識

91APP

一般開發者不理解 AWS 的 Policy 結構，都只描述行為



- 調整權限：
  - 需要使 worker service account 具有權限將檔案寫入至 S3
  - S3 Bucket list:
    - TW: S3 bucket
    - HK: S3 bucket
    - MY: S3 bucket

- 用途：因使用 NMQv3 SDK 所以需存取(讀/寫) S3 的資料
- 環境：
  - Project: TW QA for Payments' related services
- 調整權限：存取(讀/寫) S3

- 調整權限：
  - 需要使 worker service account 具有權限將檔案寫入至 S3
  - S3 Bucket list:
    - TW: S3 bucket
    - HK: S3 bucket
    - MY: S3 bucket

## AWS Policy 解析與生成

在這種情況下，我們將生成 AWS Policy，讓 Salepage Service 讀取特定 S3 Bucket:

JSON

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "s3:GetObject"
8       ],
9       "Resource": [
10      "arn:aws:s3:::Yunchiissohandsome/*"
11    ]
12  }
13 ]
14 }
```



自動轉換 AWS Policy，節省 Dev & Ops 溝通時間

# 市場環境自動 mapping 91APP 帳號/叢集/規範

91APP



好的，我們將開始進行 NKP Service 權限申請的流程。首先，我需要您提供以下資訊：

1. 環境（可多選）：例如，Prod/QA/qaN/PP
2. Project Name :
3. Namespace Name :
4. Service Name :
5. 需求團隊 :
6. 用途、調整項目 :

請您根據實際情況填寫，特別是 Namespace 和 Project Name，以便我們確認資源的規範是否符合標準。

1. TW Prod
2. Salepage 歷程記錄
3. Salepage Service
4. Salepage Service
5. Arch Core
6. 哈哈哈 就是看歷程啦

工單已成功建立！以下是工單的詳細資訊：

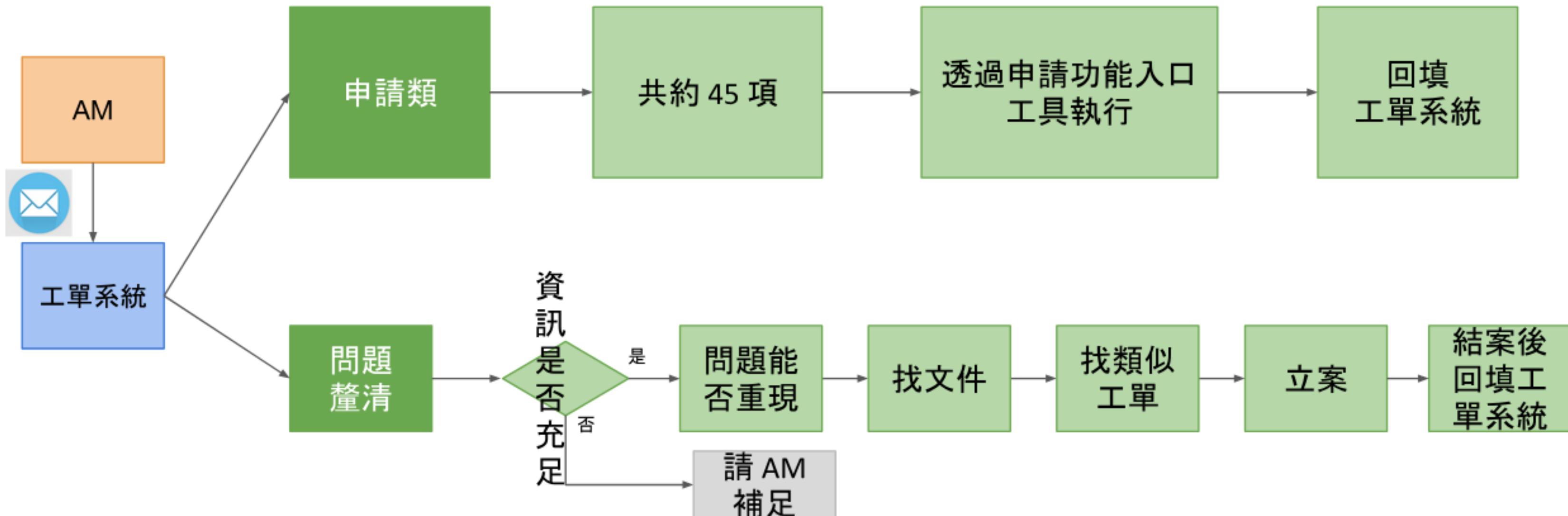
- 工單 ID: 23151
- 主題: 調整 AWS IAM Role on NKP : [REDACTED]
- 負責人: [REDACTED]
- 需求團隊: ARCH
- 需求部門: 核心服務組 (UPD.ARCH.Core)
- 相關服務: 不在此清單
- Market (業務市場): TW (台灣市場)
- Cloud Accounts: [REDACTED]
- 資料中心: AWS ap-northeast-1 (東京)



自動判別  
NKP Cluster、namespace 規範  
AWS Account、Region

# 案例三：HD 工單處理流程 (過去)

91APP



# HD: 申請類工單流程優化

91APP



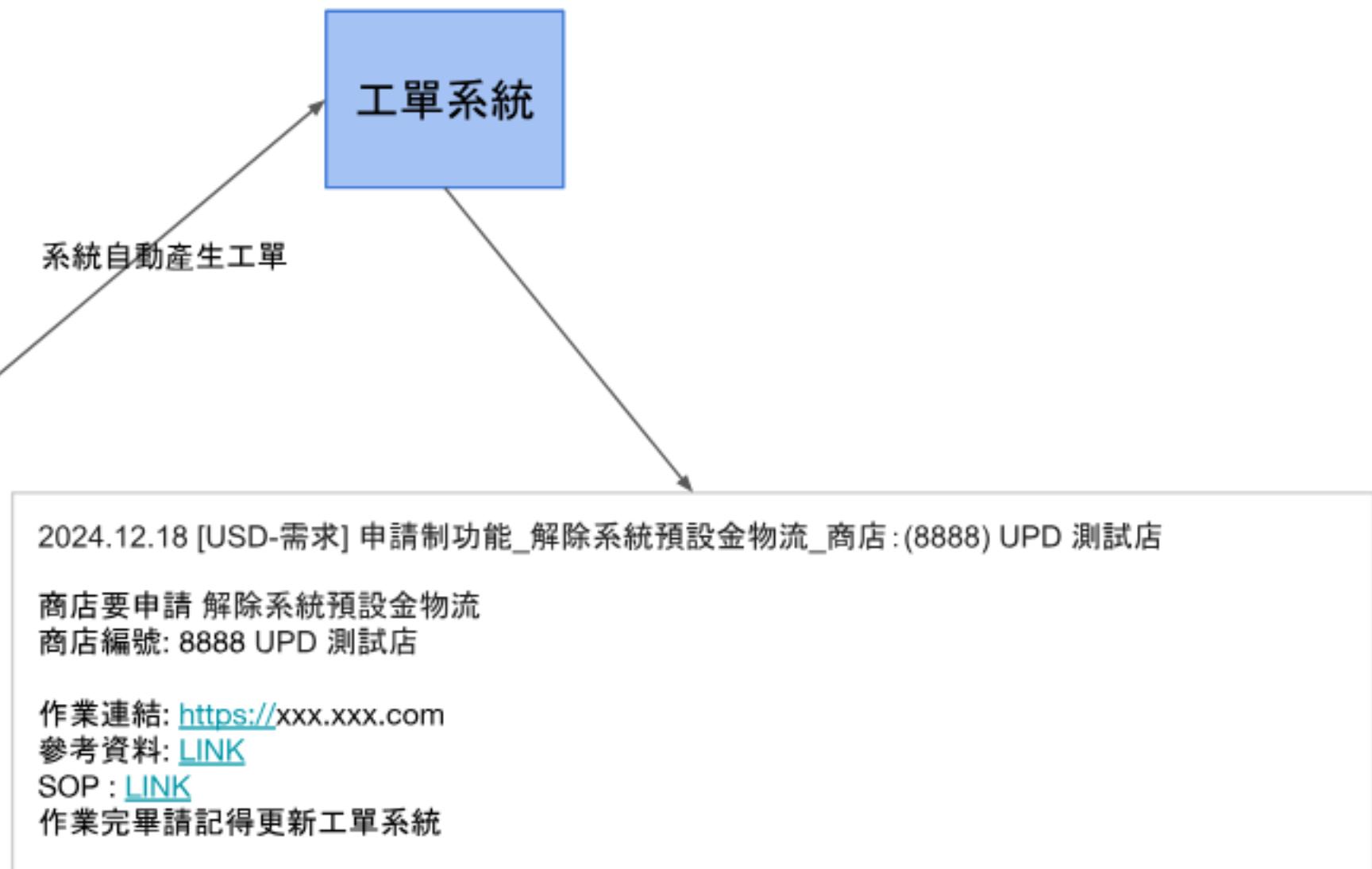
before

after

AM

The 'before' user interface consists of two orange-colored steps:

- Step 1:** A list of service types:
  - 申請類服務
  - 撈資料/改資料
  - 問題釐清
  - 新功能需求A '下一步' (Next Step) button is at the bottom.
- Step 2:** A list of application types:
  - 解除系統預設金物流
  - 箱購拆單
  - 活動互斥
  - ...Fields for ShopId, Material 1, and Material 2 are present, along with a '下一步' button.



# HD: 問題釐清工單流程優化



before

after

AM

AM

需要甚麼服務?

- 申請類服務
- 撈資料/改資料
- 問題釐清
- 新功能需求

**下一步**

問題類型?

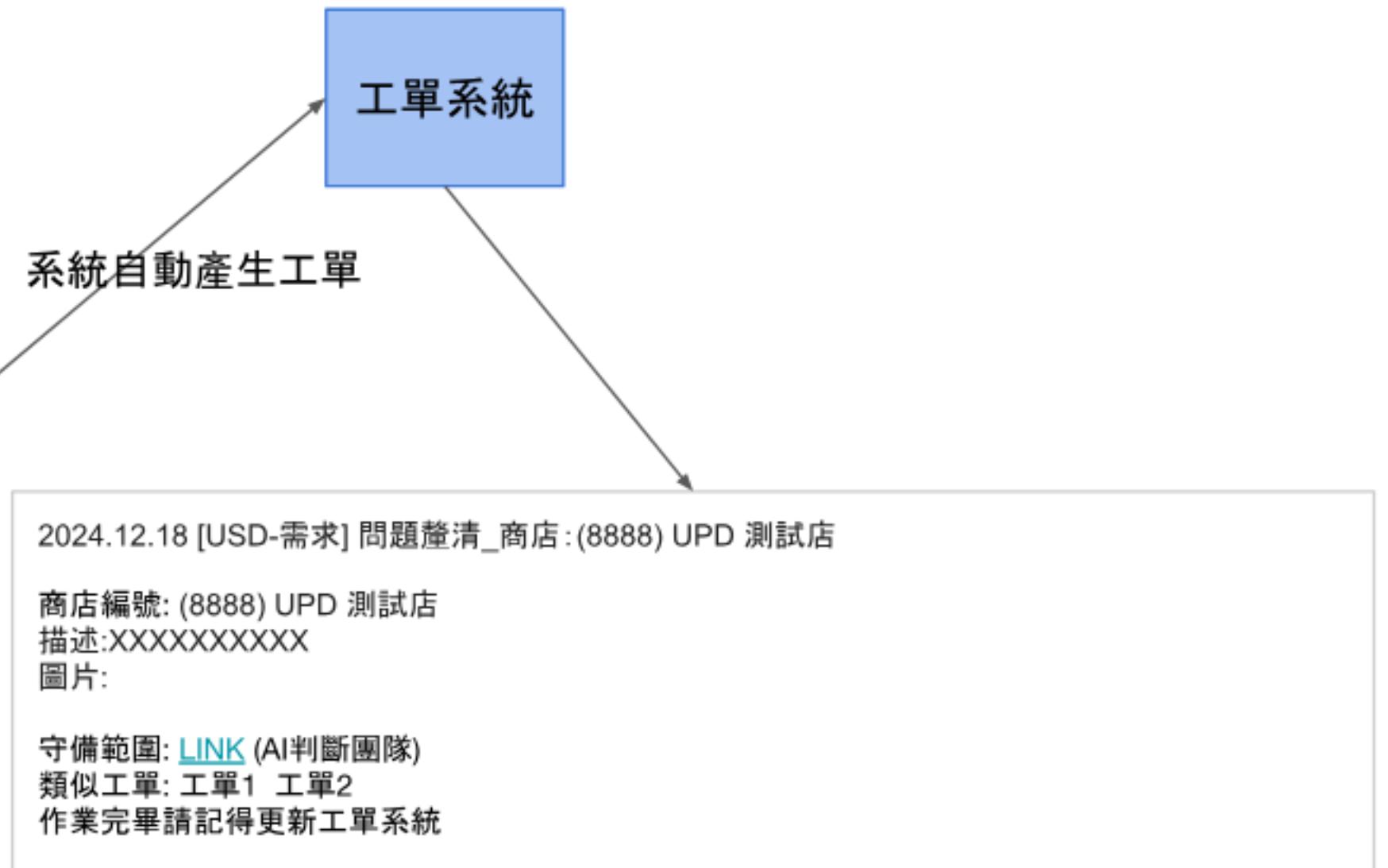
- 會員資料
- 訂單資料
- 點數資料
- 券資料...

商店編號 ShopId

描述

上傳截圖

**下一步**



## 需求:

- 跨國環境、國際大場需要多語系站台

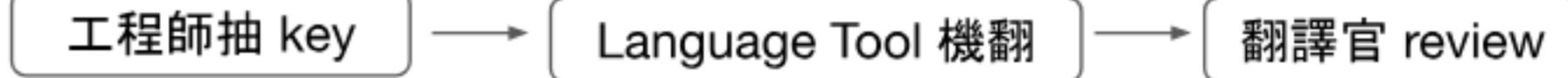
## 現況機翻問題

- 有翻，**但不多**，很常有句子不成文
- 無法參考以前翻過的內容跟翻譯 style

## 現況流程

## AI Translator

用來取代現在的機翻



原文	新機翻 Dify	現況 AWS 機翻
電聯不上	Unable to reach by phone	Telecommunication is not up
查無此人	Person not found	Look for this guy.
使用逗號將標籤樣式的標語分成多個標籤	Separate the tag slogans into multiple tags using commas.	USE A COMMA TO SPLIT TAG-STYLE SLOGANS INTO MULTIPLE TAGS
請勿輸入常態或僅活動對象查看的折扣活動，此類型需登入才能查看	Please don't enter ongoing or exclusive discount promotions that are viewable only to target customers. These types require login to access.	Please enter Discount Promotions that are viewed by regular or Target Audience only, for this type of feature required to view
活動顯示僅活動對象查看	The event is visible to target customers only.	Promotion Show Target Audience Only View
活動顯示設定類型錯誤	Incorrect type of promotion visibility	Promotion Display Setup Type Error
商店尚未支援客群活動	The store doesn't support customer group promotions yet.	Shop Fashion Does Not Support Customer Promotion
部分派車收貨失敗，消費者提供地址有誤或非物流商服務範圍	Some pickup requests failed because the customer provided an incorrect address or the pickup area is out of the service area.	Partial Carriage Receipt Loss, Consumer Provides Incorrect Address or Non-Logistic Services

# 內容產生: UX 使用 AI 取代機翻

91APP

1. 300 字文字初翻效率: 一小時 -> 30 秒, 120 倍

2. 翻譯品質接近翻譯專家: 翻譯結果可以遵守指定的風格及用詞, 確保產品文案調性一致。

3. 維護彈性高、成本低: 可針對各語系設計不同翻譯 prompt, 只要串接一次, 後續維護不需要改動程式。

4. 輕鬆擴充語料庫: 定期上傳已經翻譯過的內容讓 LLM 學習, 逐步優化、提升翻譯品質。

新增語系

Key  
Test

Page Name  
Test

Epic Name  
Test

zh-TW

en-US

ja-JP

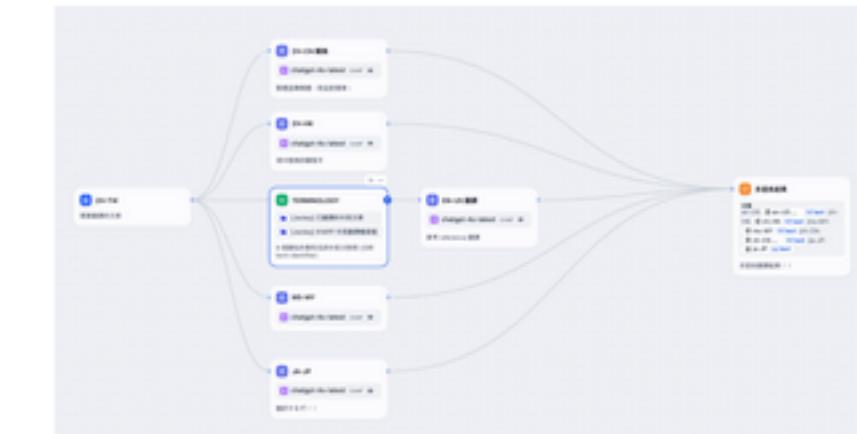
ms-MY

zh-CN

zh-HK

退出 提交 提交 & 繼續新增

1. 填寫中文



2. AI 進行翻譯

3. 回填各語言的翻譯

# 案例五 : AI Code Reviewer

91APP

目的:有效率提高 91APP 程式品質

功能:

- 有能力分析 SonarQube 規則並提出建議
- 建立 91APP 自規程式碼檢查規則, 完善

91APP 程式碼檢查規則知識庫

- 異常 / 資安 檢測
- 建立循環 (異常 -> Custom Rules -> AI Review)、準確率 90%



SU

Server Group Bitbucket Integration User commented 8 hours ago

SU

Server Group Bitbucket Integration User 8 hours ago

規則編號: csharpsquid:S2259

問題標題: Null pointers should not be dereferenced

嚴重等級: MAJOR

類型: BUG

為什麼這是一個問題?

<p>Accessing a null value will always throw a NullReferenceException, causing abrupt termination of the application. This can lead to memory leaks and other runtime errors.</p>

```
1 if (dataSetTypeList == null)  
2 {  
3     return;  
4 }
```

建議修正方式

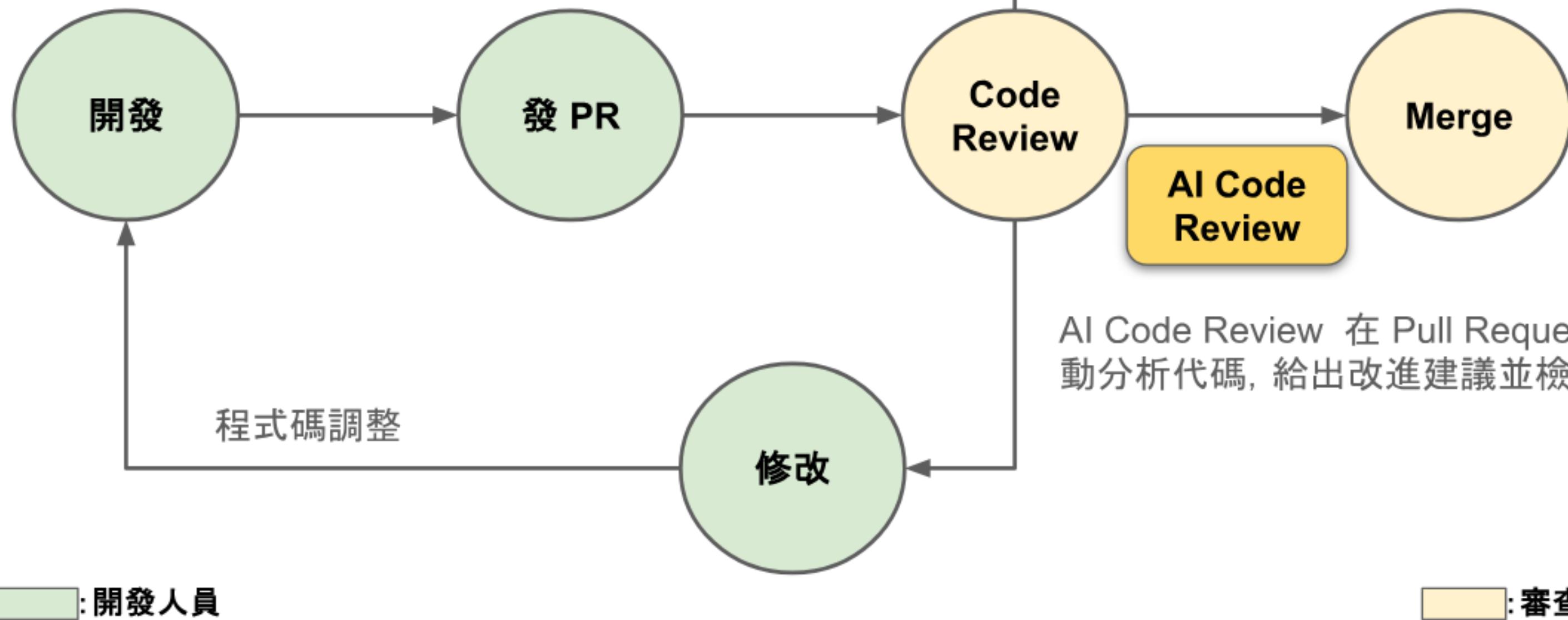
返回 null 不是最佳做法，應返回一個空集合以避免 null 檢查

建議修正後的程式碼

```
1 dataSetTypeList = dataSetTypeList ?? new List<DataSetType>();
```



- 標準規則
- 自訂規則
- 知識庫



# AI Code Review 成效說明

91APP

對 <b>開發人員</b> 帶來的好處	AI Code Reviewer 導入前	AI Code Reviewer 導入後
<b>節省時間</b>	一次 PR 約 <b>花費 1-4 小時</b> 消耗大量時間人工審查及反覆修改調整。	一次 PR 縮短 <b>至 5-30分鐘</b> 能直接給出修改建議及注意事項即時修正。
<b>品質穩定</b>	依賴經驗、coding standard 及 guideline 人腦判斷，容易遺漏應或出錯。	能直接從知識庫中提取，協助檢查應注意的不良程式碼。
<b>提升生產力</b>	分神留意 PR 的進度及建議，無法專注在下一個功能的開發，造成生產力下降。	通過自動化審查， <b>每個開發週期可以節省一整天的工作時間。</b>

對 <b>審查人員</b> 帶來的好處	AI Code Reviewer 導入前	AI Code Reviewer 導入後
<b>節省時間</b>	一次 code review 最少約花費 30 分鐘	一次 code review 縮短至 5 分鐘內， <b>參考 AI 留下的建議判斷風險。</b>
<b>品質穩定</b>	人工審查不一致，易出現遺漏和個人偏差。	審查過程一致，降低人為錯誤和疏漏。
<b>提升生產力</b>	重複性工作容易產生疲勞，本身的生產力下降。	AI 分擔重複性的工作，讓審查人員可以專注於戰略性分析工作。

## Team Processes

APPlication

Code Review

Agents for Forms

Mentor

Translate

Utils

User Info API

Ticket System API

Ticket Datasets

KBs Training

Product PRD

Evaluation

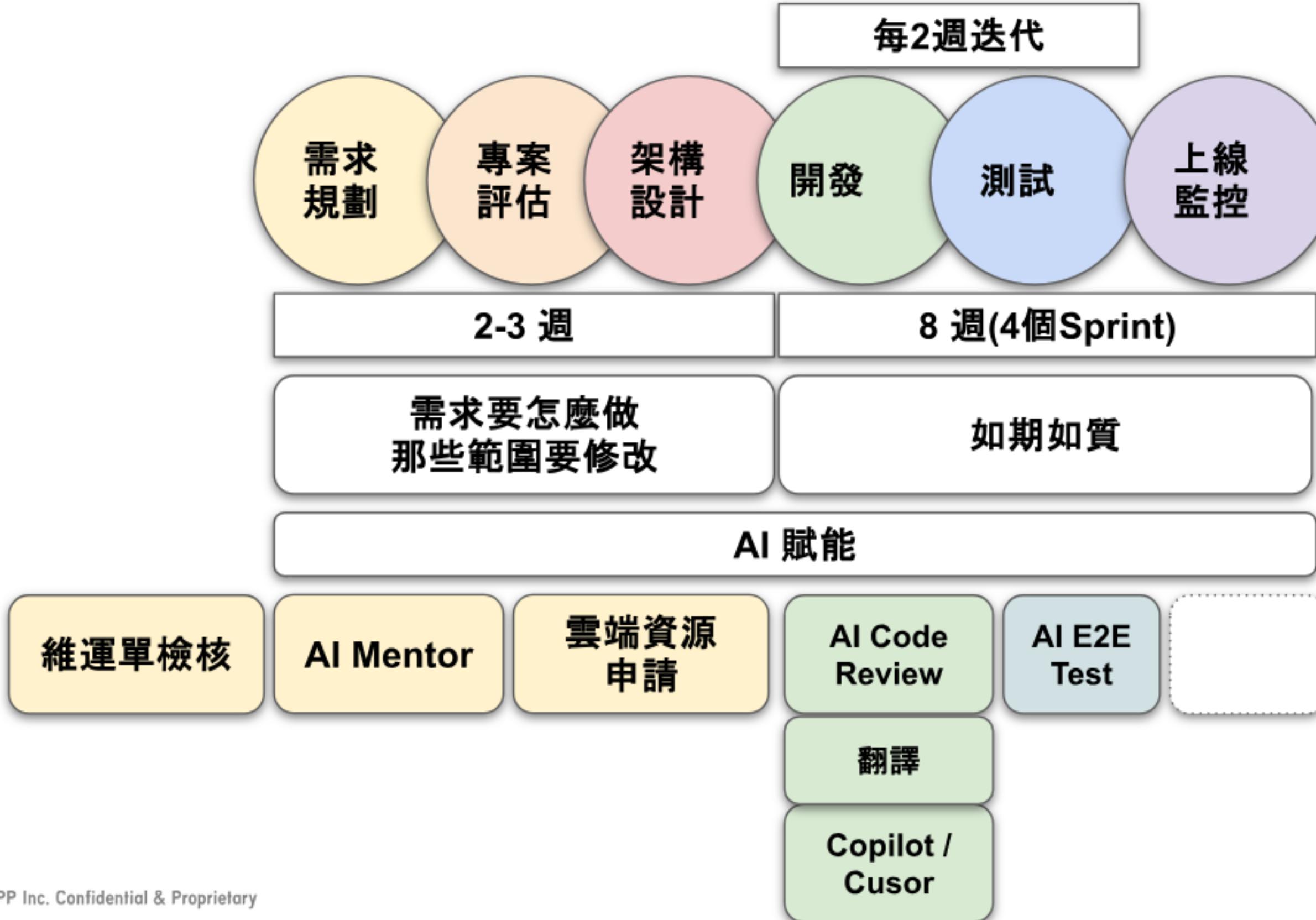
Platform

Dify

Models

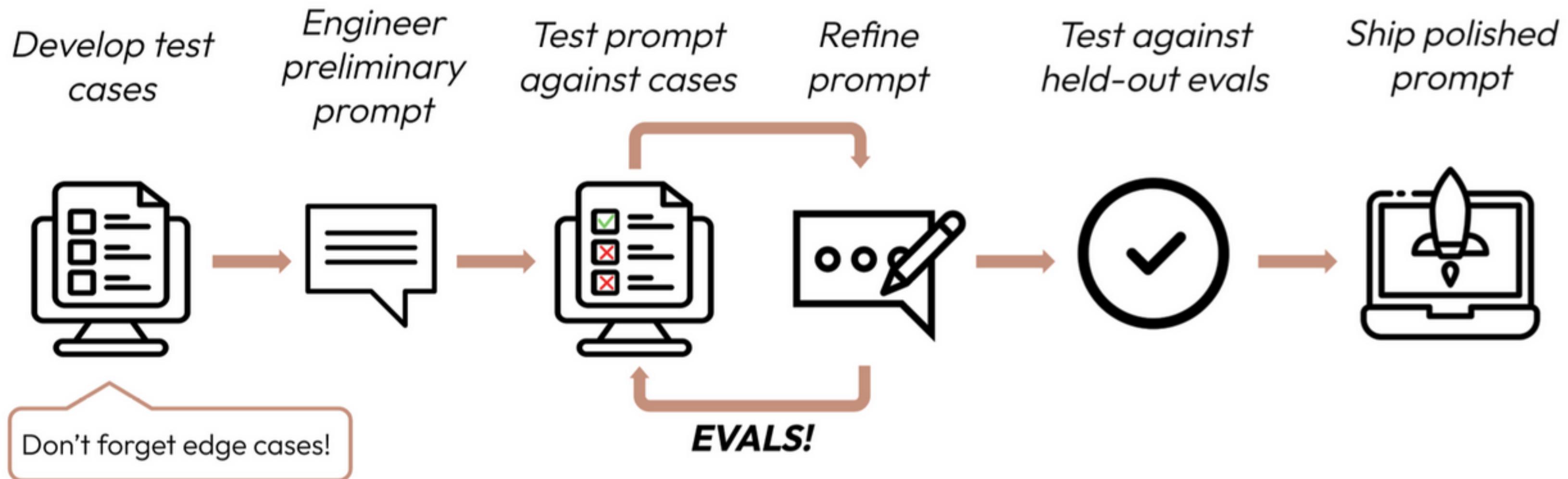
LLMs

# UPD 產品開發流程：以 4 個 sprints 開發週期為例



# 生成式 AI 軟體開發流程

91APP



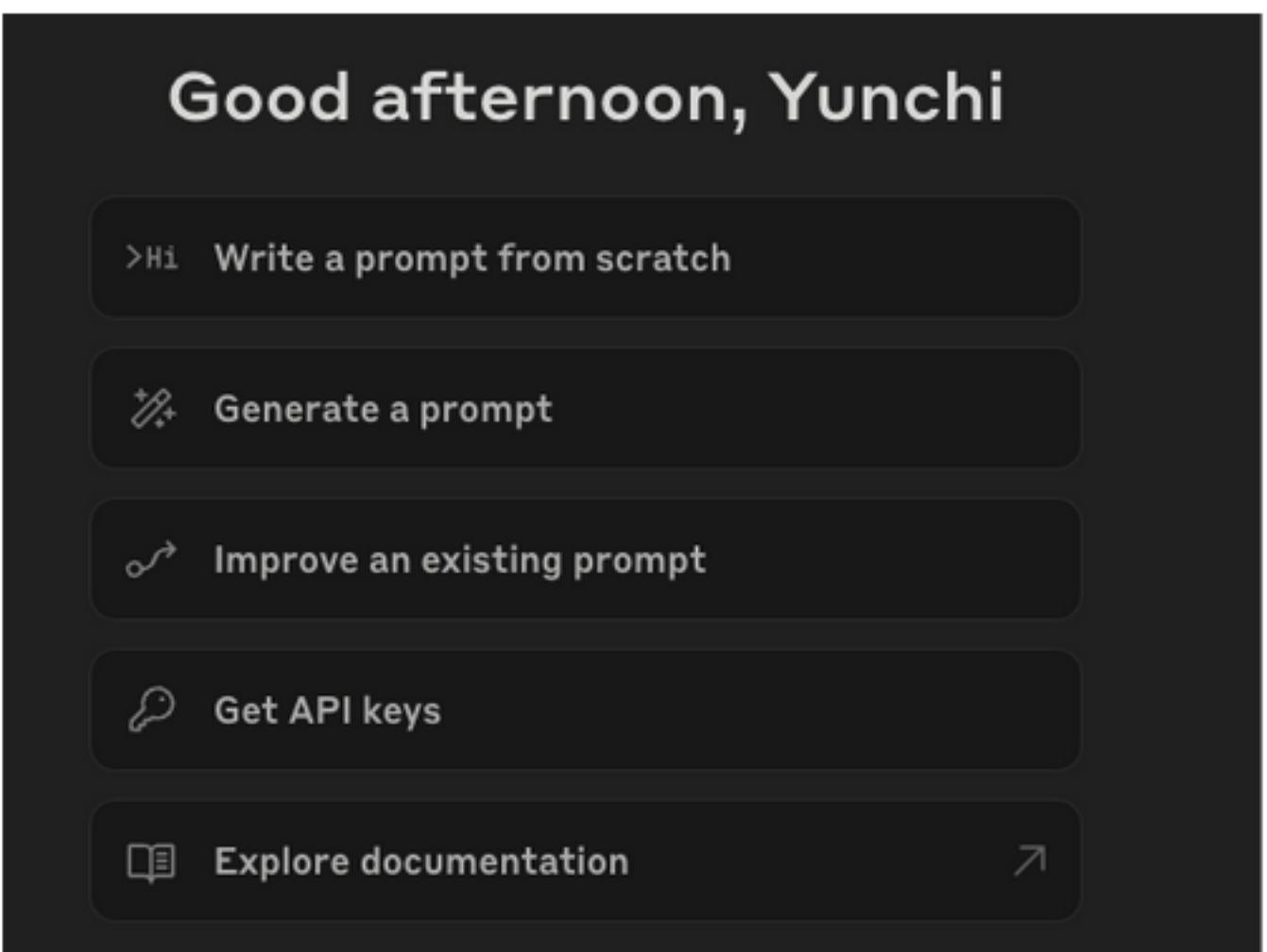
圖片出處: <https://docs.anthropic.com/en/docs/build-with-claude/develop-tests>

如何產生初步的 Prompt

有了 Input、Output、Do What

讓 AI 幫你生 prompt ! (魔法對付魔法)

<https://console.anthropic.com/dashboard>



# Engineer Preliminary Prompt

91APP

The screenshot shows the Dify AI workspace interface. On the left, the 'Prompt' tab is selected, displaying a template for an IAM Policy Ticket. The template includes a header, instructions for the AI to generate the ticket, and sections for collecting ticket information and asking developer questions. A red box highlights the 'Generate' button at the top right of this section. Below this, the 'Variables' tab lists four variables: applicant (AD Account), director (AD Account), department, and team. The 'Context' tab shows an 'AWS Permission Knowledge Base'. The 'Tools' tab lists four actions: CreateIssue, GetCustomFields, GetUsers, and saveUserData, all of which are enabled. On the right, the 'Review & Preview' section shows a generated response message from the AI, which reads: 'Hello 工程師們，我是 AWS 工單助手，可以點選 Go - 我要建立 AWS 權限工單' (Hello engineers, I am the AWS Work Order Assistant, you can click Go - I want to create an AWS permission work order). At the bottom right, there is a 'Manage' button.

Dify Star 74,290

探索 工作室 / AWS 工單助手 - Latest V... 知識庫 工具

A archieshih@91app.com Agent 設定 gpt-4o CHAT 發佈

編排

提示詞

# IAM Policy Ticket 輔助助手說明

您是一個專門輔助一般 Developer 提交更高品質 ticket 的 AI 助手，其目的是協助 DevOps 工程師加速工單處理。您的主要任務是幫助 Developer 收集 ticket 相關信息、分析請求的權限，並生成適當的 IAM 政策，以便 DevOps 工程師更快地處理這些工單。

請按照次標題(##)順序詢問使用者問題，還沒問完一個次標題的問題，不要提及到下一個次標題內的問題

## 1. 收集 Ticket 信息

請先向 Developer 詢問以下關鍵信息：

3633

(\*) 變數

變數 KEY	欄位名稱	可選	操作
applicant	申請人 (AD Account)	<input type="checkbox"/>	<input checked="" type="radio"/> 白
director	主管名稱 ( AD Account)	<input type="checkbox"/>	<input checked="" type="radio"/> 白
department	部門	<input type="checkbox"/>	<input checked="" type="radio"/> 白
team	團隊	<input type="checkbox"/>	<input checked="" type="radio"/> 白

上下文

AWS 權限知識庫

工具

CreateIssue GetCustomFields  
GetUsers saveUserData

除錯與預覽

申請人 (AD Account)

主管名稱 ( AD Account)

部門

團隊

Hello 工程師們，我是 AWS 工單助手，可以點選 Go - 我要建立 AWS 權限工單

查詢 AD 帳號全名 Go - 我要建立 AWS 權限工單

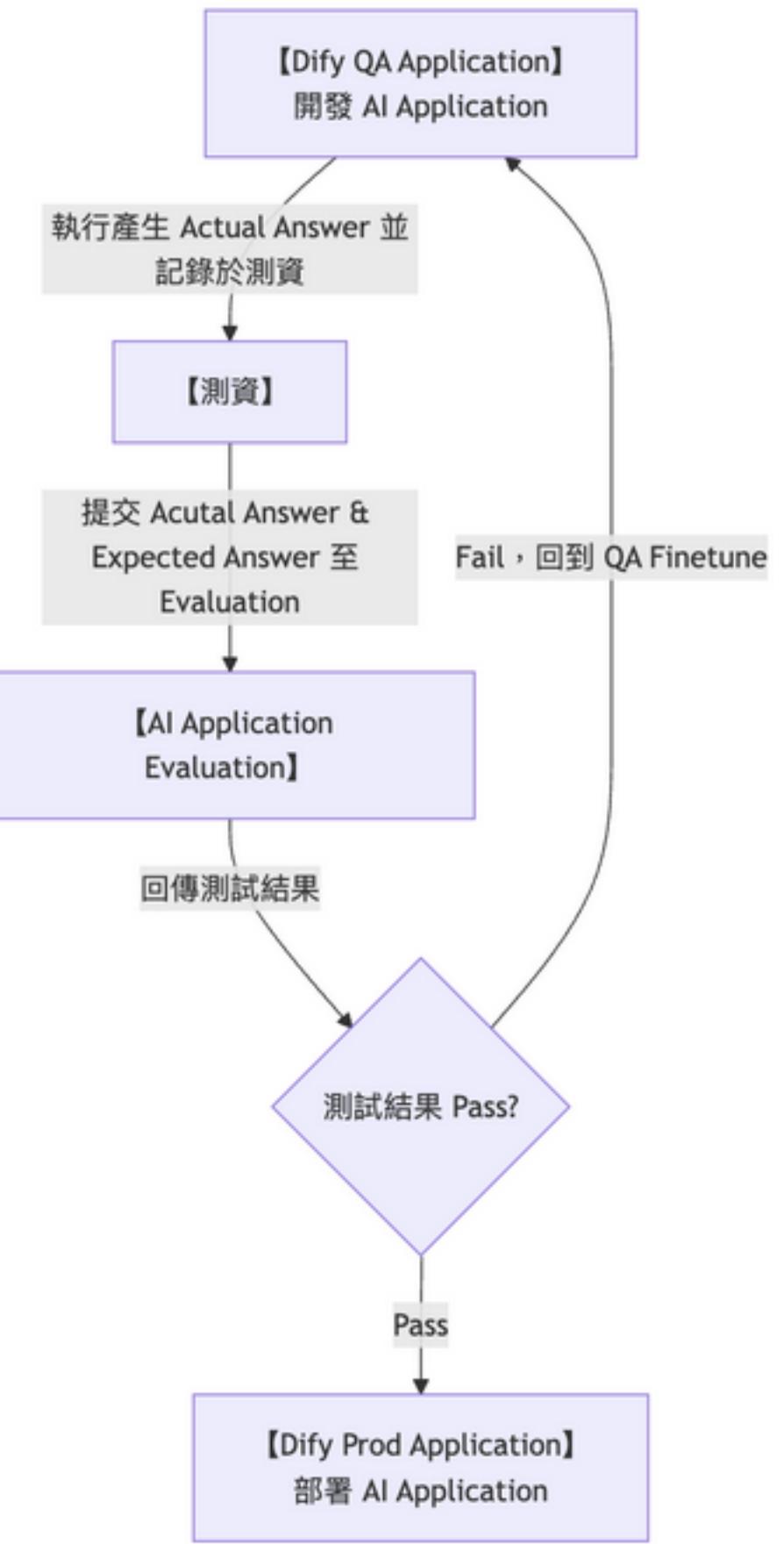
與 Bot 對話

Features Enabled Manage →

# Test prompt against cases & Refine prompt

P

測項	測項1	測項二	測項三	平均
分數項	AI Eval Score	AI Eval Score	AI Eval Score	AI Eval Score
Release 1.0.0	81	40	50	57
Release 1.0.1	51	20	0	23.66



# Test prompt against cases & Refine prompt

## 1. 有標準答案(可自動程式比對)

- 分類題(如 A / B /C / D)、是非題(Y/N)
- 可直接用程式判斷正確性

## 2. 無標準答案(需 AI 判斷)

- 比對 Actual vs. Expected, 請 LLM 當評審打分
- 適用摘要、翻譯等主觀任務

## 3. 可自動化評估的方式

- 有標準答案 → 用程式算準確率
- Assertion 檢查 → 特定字串、格式或結構是否符合

# Prompt to Product: LLM 為核心的應用程式開發

// AI 時代注重 "成果" 而非 "服務", 持續優化 "服務" 的流程與文化該如何改變?

91APP, Chief Architect

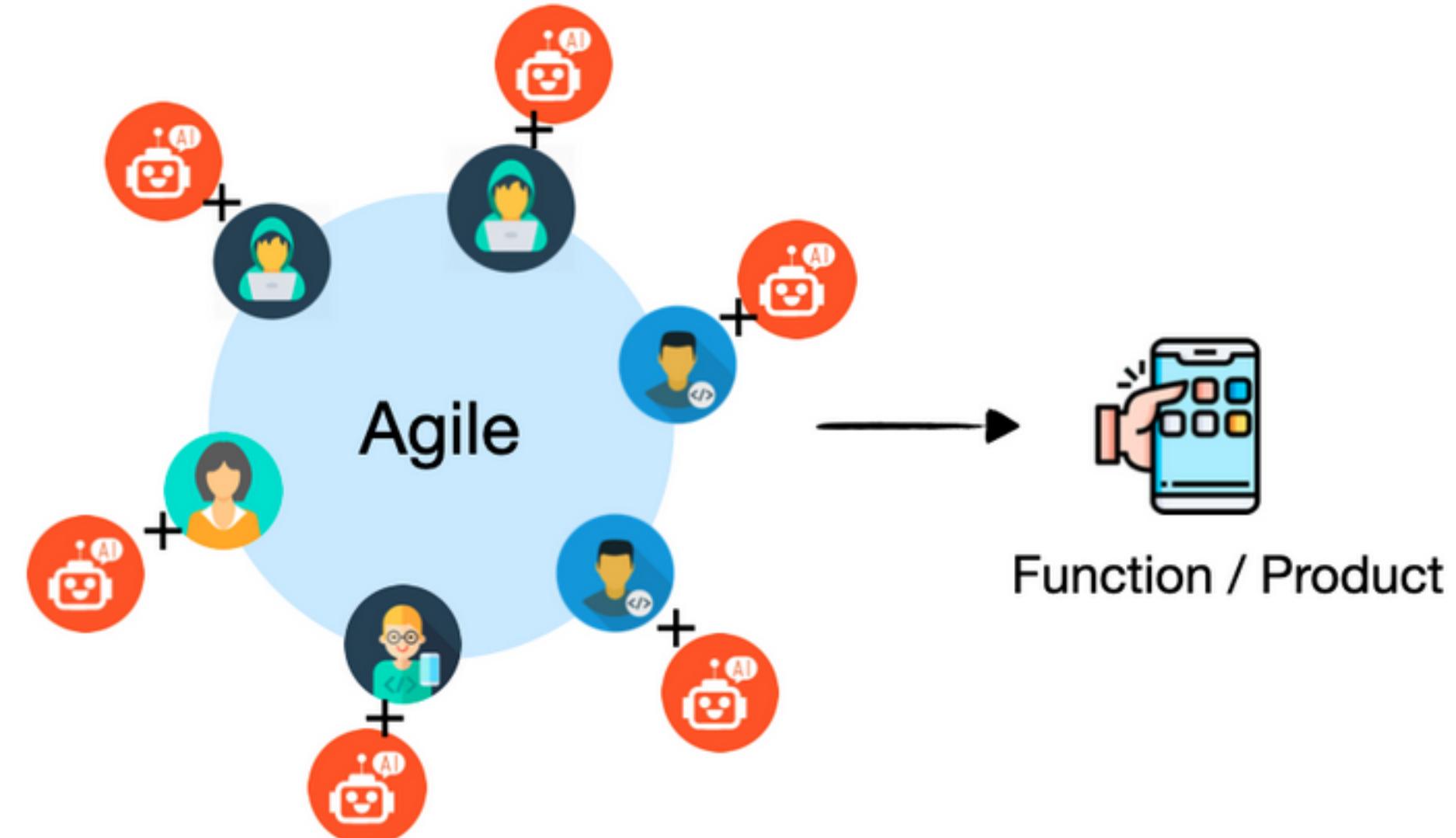
Andrew Wu

DevOpsDays Taipei 2025, 2025/06/05

# 軟體手工業的黃昏？軟體工業化的未來？

91APP

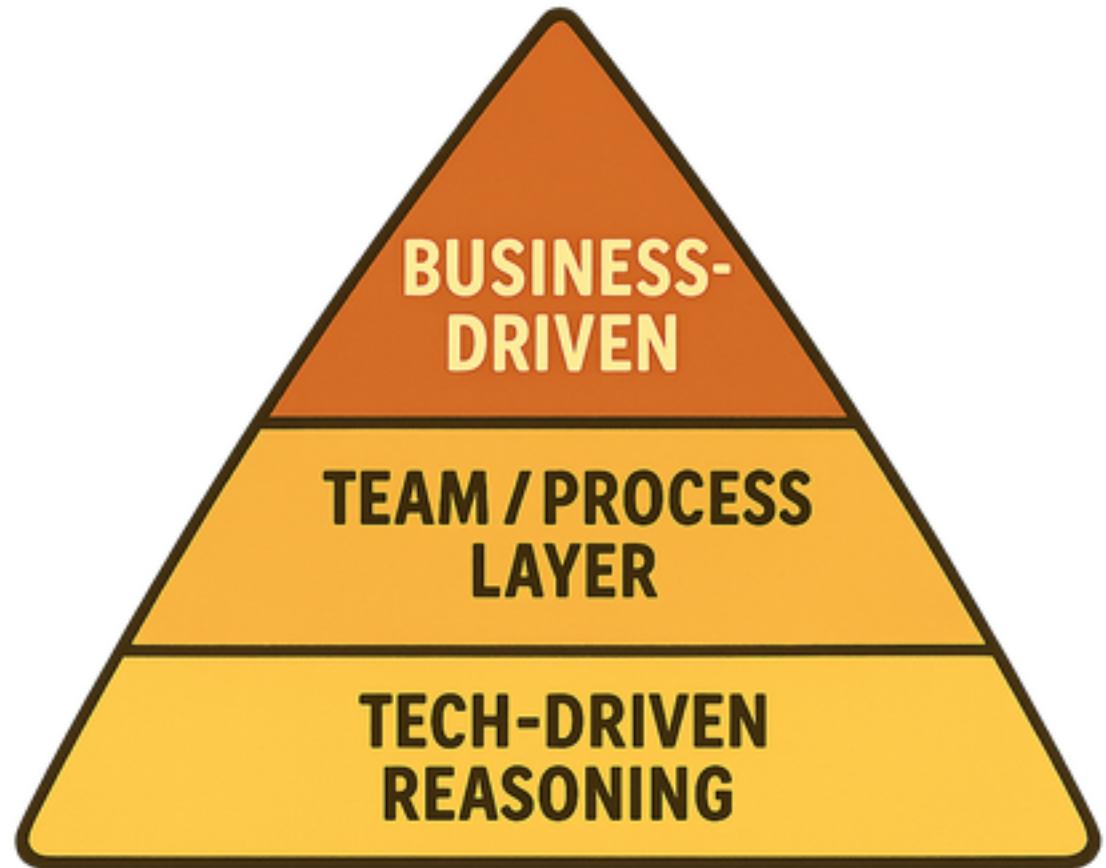
- 300 人的開發團隊過去透過導入 Agile 找到把人連起來的工作方法
- 目前的現況：300 人 + 300N AI 助手讓團隊生產力有機會 Scale-Up
  - ChatGPT, NotebookLM, Cursor, Claude, Copilot, AI App
- $300 \text{ (人)} * 1,000 \text{ (每月單價)} * 12$ 
  - NTD 3,600,000



# 最難的永遠是人

- 一線人員的**接受度**, 管理層要看到**成效**
- 拆解成小問題:Pilot 人 + Pilot 專案
- 墊高地板:陪跑教練、標準化、AI App
- 賦能、讓他們解決自己的問題
- 迭代

- **技術層比較 (Tech-Driven Reasoning)**
  - 技術 A 比技術 B 更好
- **流程與團隊效能 (Team Impact / Process Layer)**
  - 這項技術會如何影響我們的開發流程與人力資源效率
- **商業驅動 (Business-Driven Decision)**
  - 導入這項技術，能為公司省錢、賺錢、加快產品交付





#AI 寫程式  
提升產品開發的速度、精確與效能



#AI 做素材  
強化電商代營運效率與人力

要進入HR系統請務必依以下步驟操作：

1. 登入eHR Portal
  - 網址：<http://vhx.91app.biz/ehrportal/LoginOriginal.aspx>
  - 使用AD帳號且密碼登入。
2. 進度申請
  - 建議使用桌面電腦或手機進行操作，並連接91APP內部網路（或透過VPN進入）。
3. 功能
  - 在eHR Portal中，您可以申請請假單、假勤與勤單，以及查看假勤相關資訊。
4. 詳細說明
  - 若需要更詳細的操作指引，可以參考[91APP操作手冊] (<https://docs.google.com/presentation/d/1LGQzTl4Mu5cPHtYgR6nsJnTsERZ7mlUAWig1OoQ/edit#slide=id.g207>)。

#AI 面試  
幫助主管高效招募優秀人才



#主管帶領 AI 新人才  
形成新團隊力量



# 2024 AIE 大會: Devin AI 創辦人 Scott Wu 的演講

91APP

在軟體工程的核心部分，軟體工程師無論在哪個層級，其實都在同時做兩份工作：

- 第一份工作：**精確地分析出你將構建的解決方案是什麼，你將如何構建它？**
- 第二部分是，你將**處理除錯與實現不同功能或撰寫單元測試**，或者所有其他涉及到你知道做的事情與實現過程中的事情。

我認為現在普通的軟體工程師可能花 10% 或 20% 的時間在第一個思考部分，而花 80% 或 90% 的時間在實現部分。

我認為 Devin 的未來，Devin 讓你**不必去設置 Kubernetes，不必去除錯那些故障的 API，不必處理版本變更或遷移**，或者那些在軟體工程中佔用大量時間的其他事情，而你真正會花時間在解決問題的關鍵上。**這有點接近技術架構師和產品經理的混合體。**

所以，我認為我們將不再稱為**軟體工程師的工作會改變**。但實際上我認為會有比較更多的軟體工程師。我認為這方面有很多先例：**以前編程意味著穿孔卡片，然後是匯編，再後來是 C 語言。隨著這些技術的發展，大多數人不再使用穿孔卡片，但實際上程序員的數量比以前多得多。**

我認為多數人被低估的一點是**需要編寫程式的職業是多得多**。這很有趣，因為我們在過去所有行業都發現這點。我認為軟體是過去40或50年來世界進步的首要驅動力。儘管如此，**我認為我們對軟體的需求實際上比現在多 10 倍。**

有人問，隨著所有這些簡單任務的解決，對於需要更複雜編程的**初級工程師或實習生來說會發生什麼？**我認為，坦白說，**我認為需求將繼續隨供應增加。**

我認為這些**高級軟體工程師會做很多這些心智模型，如果新來一個非常棒的工程師，通常不會說他們打字速度很快**。通常是指他們理解問題很好，他們知道許多不同的架構，他們能準確在何處 edge-case 情況發生。所以這些是我認為早期進到這些知識的**關鍵**。

掌握脈絡的人 + 捲起袖子的人  
不會第一個被時代淘汰

91APP

End