

AI 時代為什麼需要產品思維？

從前有需求時

軟體公司想產品，工程師寫程式

買 ERP
買 Salesforce
買現成工具

不合用
客製要花大錢
或無法客製



現在有需求時

你想產品，AI 寫程式

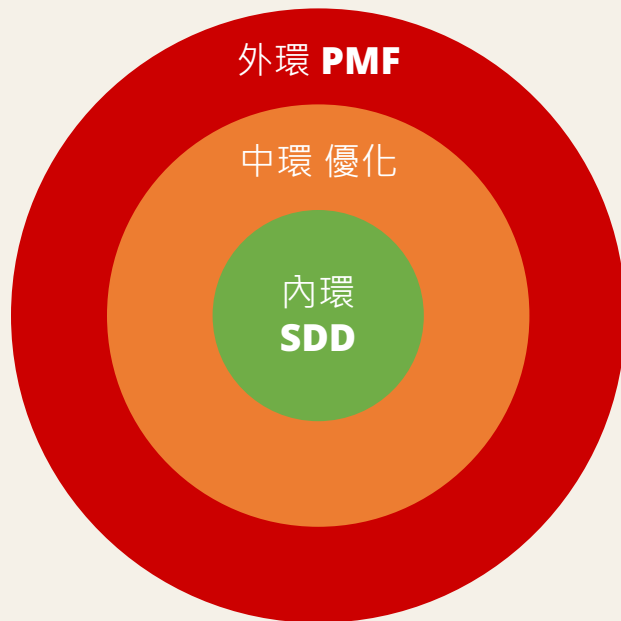
做 ERP
做 CRM
自己開發工具

你有產品思維嗎？

要做什麼？
給誰用？
解決什麼問題？

商業到產品的多層「碎型」迭代

OODA



內環 規格實作

這次做出來對嗎？

301-D1

中環 產品優化

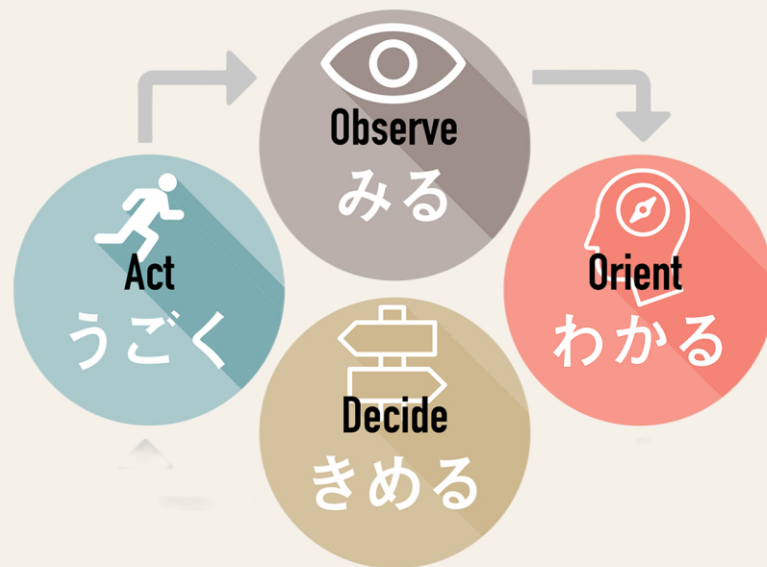
產品功能對嗎？

301-D2

外環 PMF

產品方向對嗎？

老闆定義、你達成



空戰（未知）：**OODA** (Observe, Orient, Decide, Act)

#敏捷

#速度

#適應變化

#競爭優勢

製造（可控）：**PDCA** (Plan, Do, Check, Act)

#準確

#穩定

#減少變異

#品質提升

需求書的前世與演變



- 從前，需求寫成一本厚書招標（RFP）
- 經過五次重寫、拆解，才到執行
- 每段溝通一點點誤差，就放大成災難

SDD：人人能懂的用戶需求，AI 才會懂

Spec Driven Development — 你不說清楚，AI 就亂做

EPIC (史詩)

- 要較長時間完成、跨越多個衝刺 (Sprint) 的使用者故事
- 難管理

STORY (使用者故事)

- Epic 分解，較小、可獨立完成、可在一個衝刺完成
- 好管理

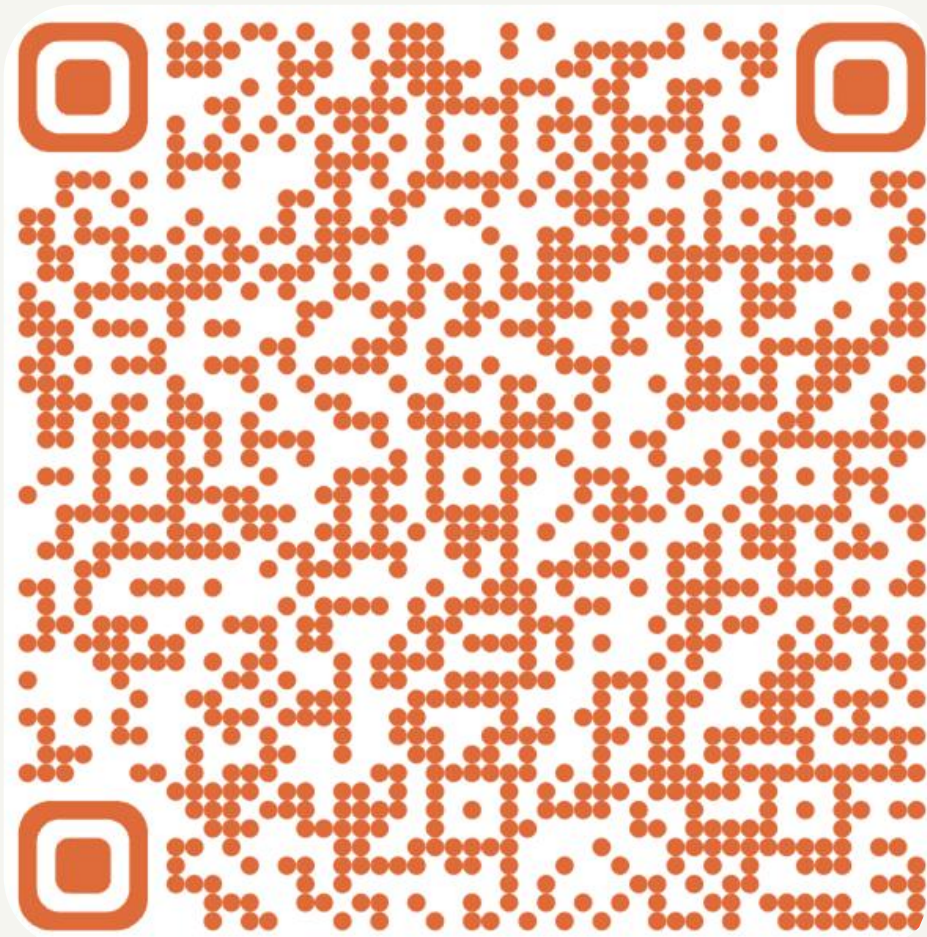
角色 Persona	功能 Function	好處 Benefit
身為 {角色}	我想要 {某功能}	以便 {對我的好處}
As a {Persona}	I want {Function}	So that {Benefit}

SDD：總目標 ⇒ 史詩 ⇒ 故事

- **Epic 1**：線上購物車功能。
 - **Story 1:** 身為顧客，我想要能夠將商品加入我的購物車，以便於我可以在之後一起結帳。
 - **Story 2:** 身為顧客，我想要在購物車中看到商品的數量和價格小計，以便於我隨時了解花費。
 - **Story 3:**
- **Epic 2**：使用者註冊與登入系統。
- **Epic 3**：.....

1 The Mapper

- **User Story 轉化師 (The Mapper) 是 Spec Driven Development (SDD) 第一個環節。**
- **將產品總目標分解成史詩 (EPIC) 和更小、可獨立完成的使用者故事 (STORY)，確保用戶需求「人人能懂」，讓 AI 理解並執行。**

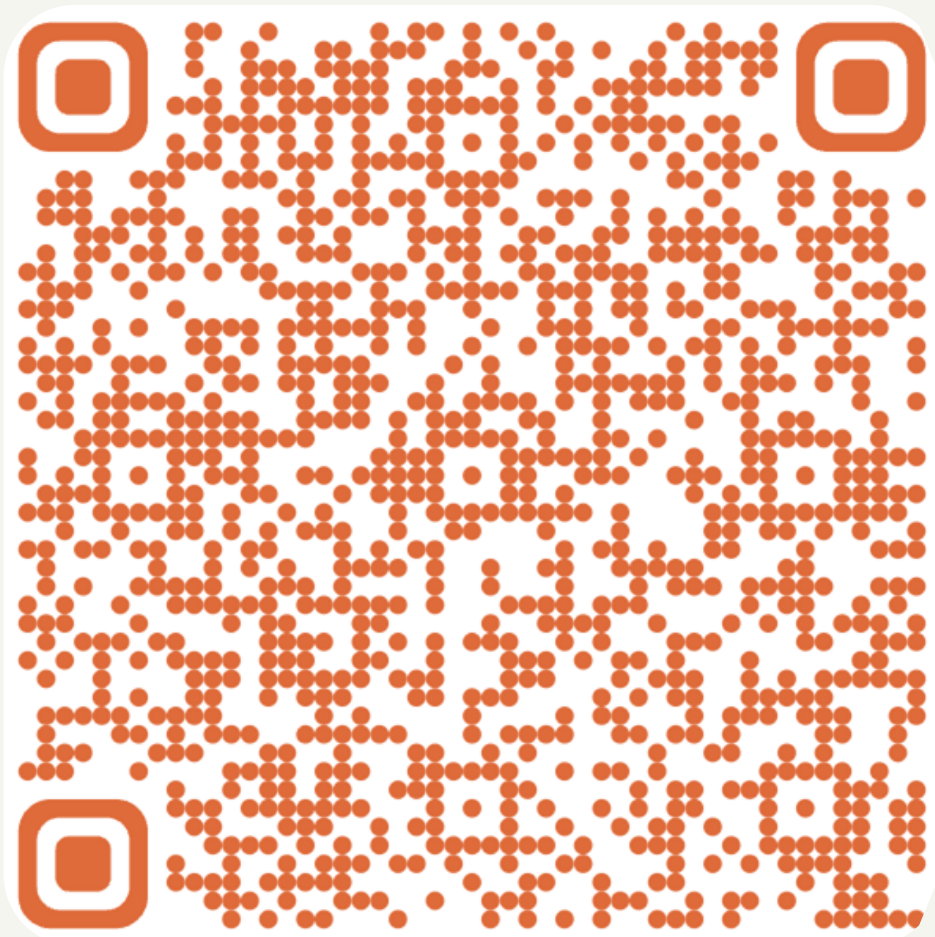


SDD：故事 ⇒ 驗收標準EARS

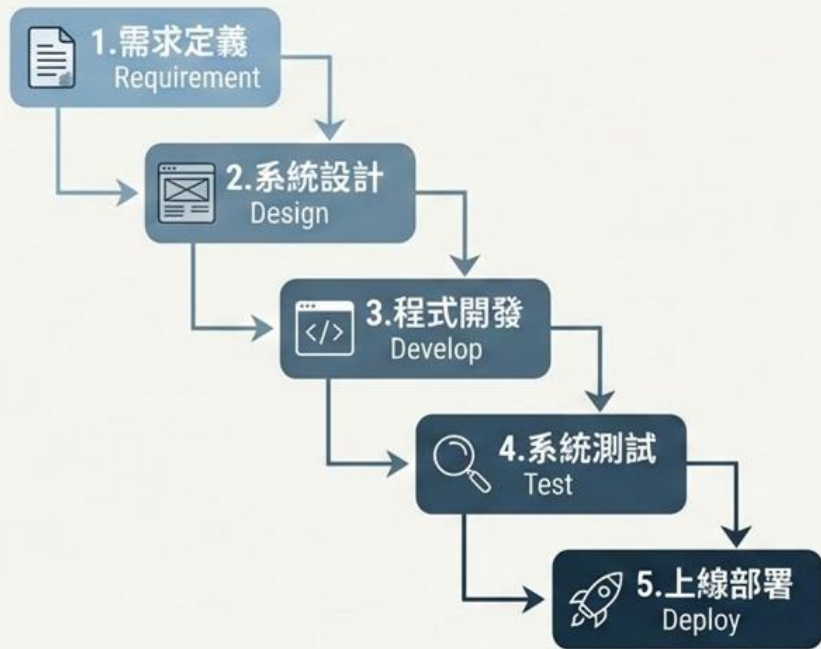
模式名稱	描述
普遍需求 Ubiquitous	系統 始終存在 的基本屬性，沒有觸發條件
事件驅動 Event-Driven	系統在 特定事件發生時 必須如何回應
狀態驅動 State-Driven	系統在 特定條件持續存在時 必須保持的行為
可選功能 Optional Features	只有在系統具備某項可選功能或組件時才適用的需求
非期望行為 Unwanted Behaviors	系統對 非預期情況 （如錯誤、故障、使用者偏差行為）的回應
複雜需求 Complex	結合多個關鍵字，表達更複雜或組合的邏輯

2 The Specifier

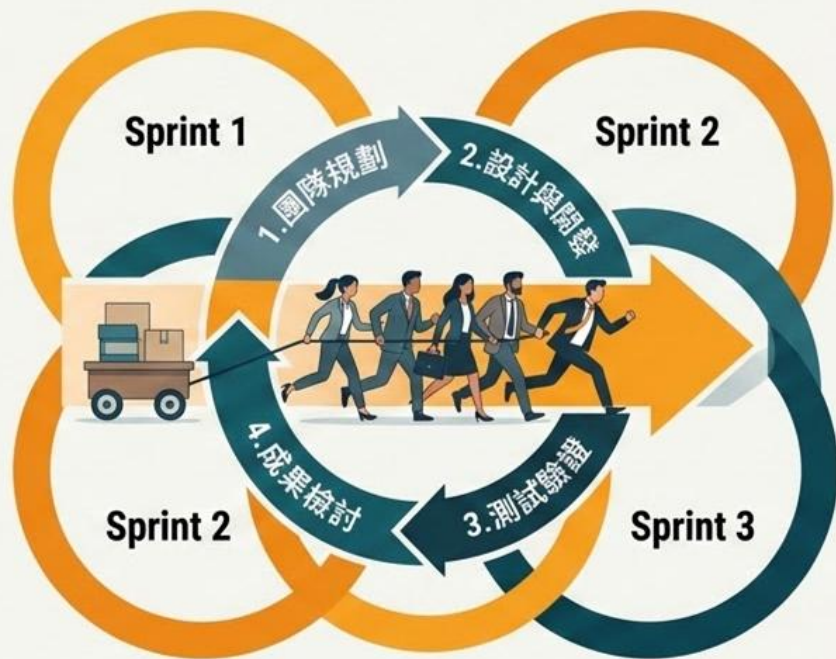
- EARS 規格工程師 (**The Specifier**) 是 SDD 流程中的第二個環節。
- 把 **User Story** 轉化師建立的 **使用者故事 (STORY)** 轉化為 **EARS 驗收標準**。



SDD：驗收標準 ⇒ 增量待辦事項



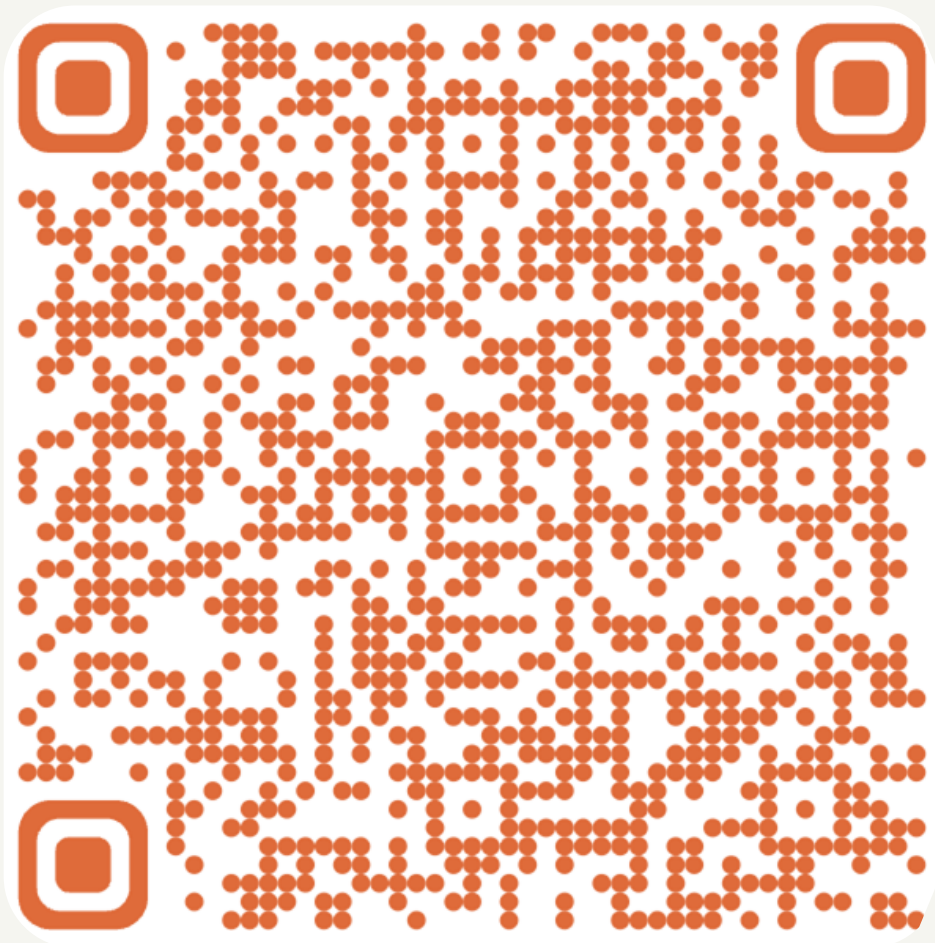
「線性」、「一關接一關」、「單次發布」



「迭代」、「團隊共同前進」、「持續交付」

3 The Planner

- 任務拆解師 (The Planner) 是 Spec Driven Development (SDD) 流程中的第三個環節
- 它將 EARS 規格工程師建立的 EARS 驗收標準，轉化為適用於增量開發的增量待辦事項
- 為敏捷開發中的 Sprint 規劃提供了執行清單



The SDD Generator

連續動作

- 分解動作太慢？
- 這個一次產生全部 SDD
- 每階段都需你同意「人類參與循環」(HITL , Human-in-the-Loop)

