



Prompt to Product: LLM 為核心的應用程式開發

91APP, Chief Architect

Andrew Wu

STUDY4LOVE 與大師對談, 2025/06/28

講者簡歷, 個人部落格 (<https://columns.chicken-house.net/about/>)

- 現任 91APP, 首席架構師
- 榮獲 Microsoft MVP 微軟最有價值專家 獎項, 2016 ~ 2025+
- 擔任多次 Microsoft Azure Cafe, TechDays, TechEd, DevOpsDays Taiwan, .NET Conf 等大型研討會講者, 與相關 Azure PaaS 雲端系列課程與企業內訓講師

演講經歷

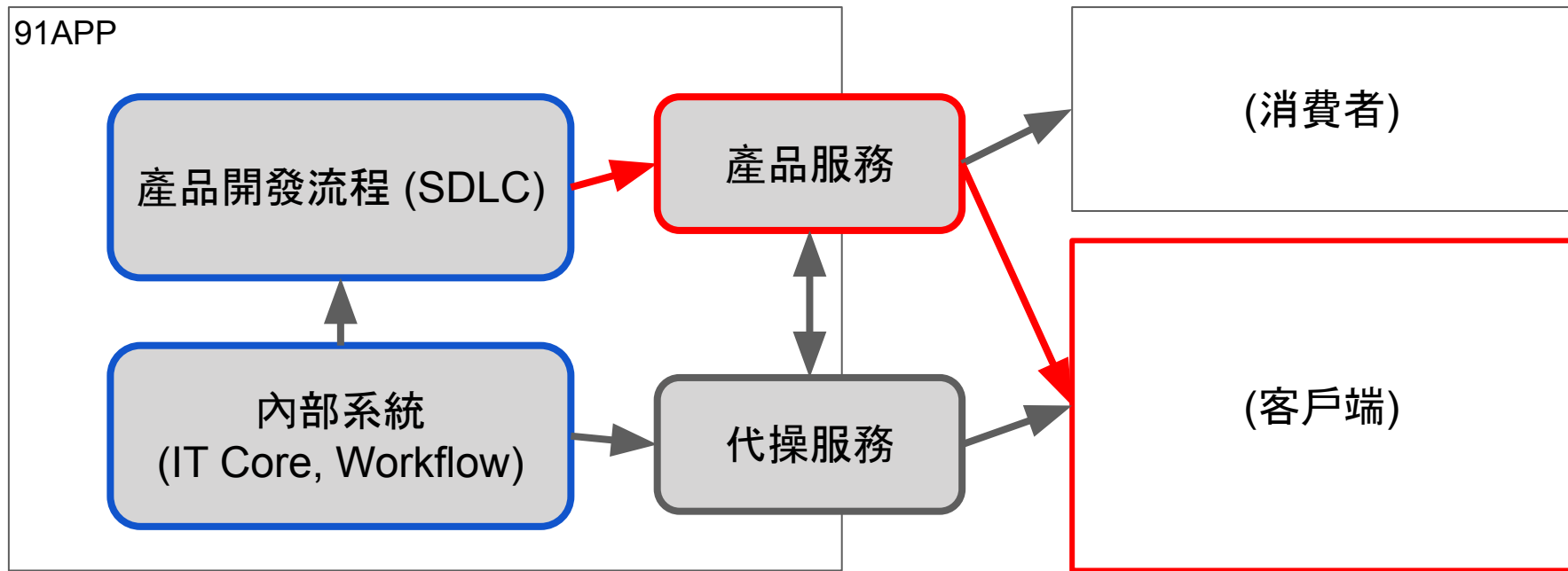
- DevOpsDays 2025, 從 Prompt 到 Product, LLM 為核心的應用程式開發
- Generative AI 2025 開發者年會, 從 Prompt 到 Product, 掌控 AI APP 的正確率
- DevOpsDays 2024 Keynote, 從 API First 到 AI First
- DevOpsDays 2023 Keynote, 架構師也要 DevOps - 談服務模型的持續交付
- DevOpsDays 2022 Keynote, DevOps 潮流下的 API First 開發策略
- DevOpsDays 2021, 大型團隊落實 CI/CD 的挑戰
- DevOpsDays 2019, 從零開始的 DevOps
- DevOpsDays 2018, 微服務基礎建設 - Service Discovery

AI 轉型

領域模型, API 的重要性
(BizDevOps)

基礎建設 / 維運議題
(DevOps)

我們在那些地方運用 AI ?



1. 用 AI 開發產品 & 改善流程與效率 (Prompt)

2. 開發 AI 產品服務 (Product)

使用 AI 方式本質上的差異

內部流程優化：

- 大家都已習慣用 Prompt 解決問題
- 大家都已理解 AI 的產出需要被 Review
- 內部使用, Prompt 重複執行的次數 1000 次內
- 提升每個團隊成員的工作效率

對外產品設計：

- 客戶不一定都熟悉怎麼下 Prompt (需要封裝)
- 客戶期待產品給的結果都是可信賴的
- 產品內的 Prompt 會被重複執行 100萬次以上
- 突破關鍵問題, 用產品的形式提供客戶價值

AI 進化, 工程師的領域轉移

內部流程優化：

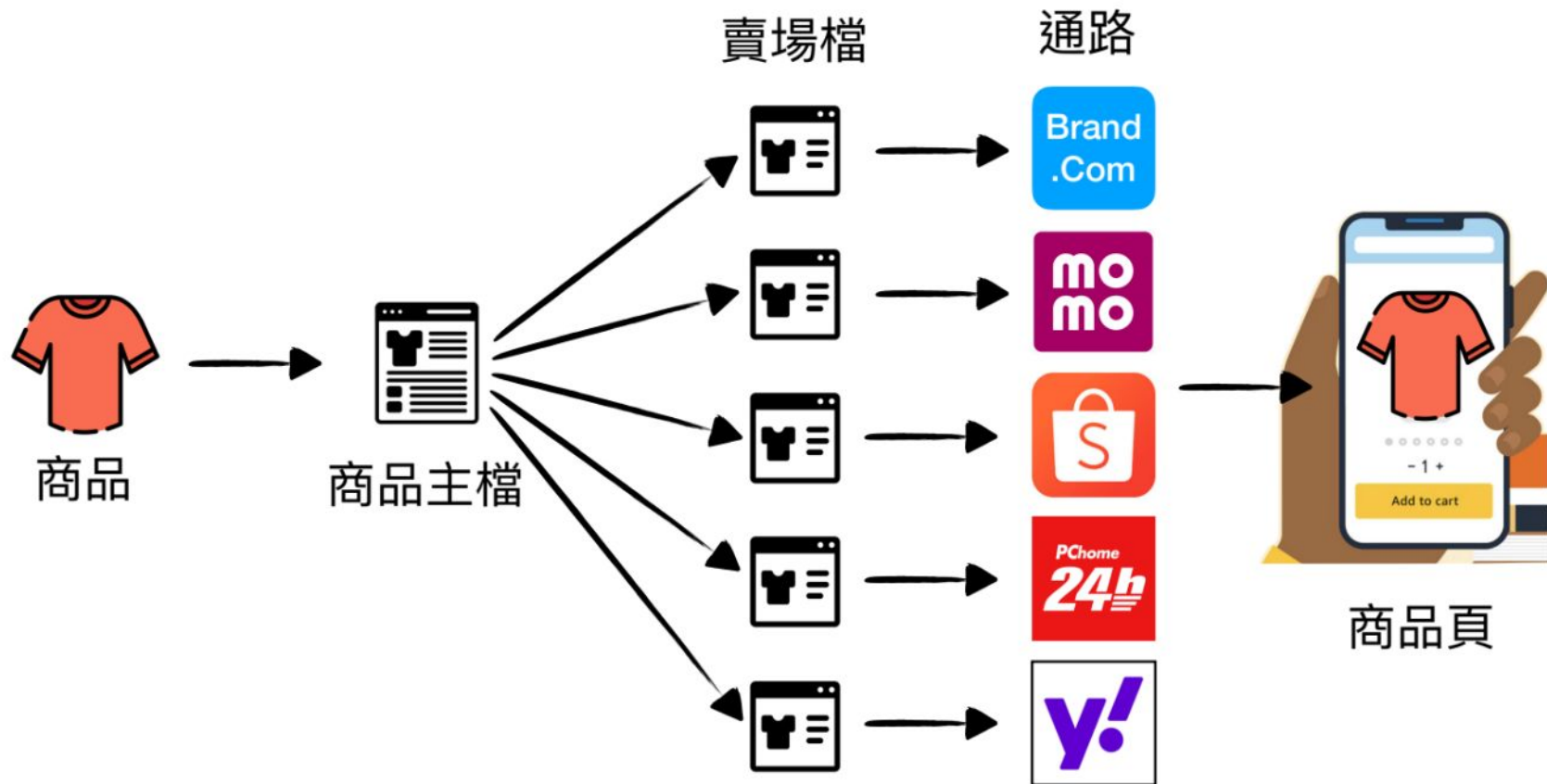
- !! 非工程人員: 開始也能有 coding 能力
- !! 第一線人員: 比工程師更能理解現場需求
- !! 有經驗的人: 一個人的生產力可以抵 10 人
- 需要能理解新工具, 以及搭配的組織流程調整
- 要拿部分 AI 賺到的時間, 強化過去的不足

對外產品設計：

- 需要掌握 AI 的特性, 設計產品規格, 系統架構
- 需要掌握跨系統 (AI 與非 AI) 的整合能力
- 需要掌握 AI 的不確定性, 與持續優化能力
- 善用自身 (內部) 使用 AI 的經驗, 融入產品設計

零售業的工作日常：商品上架

91APP



商品上架時，有很多欄位要填：

平台類目

* 商品分類 鞋/包/箱 男鞋 靴子

* 業績屬性內容代碼 2000200001

商品規格

商品規格

選完分類後，需

* 尺寸

* 材質

* 對象與族群

* 款式（鞋頭）

選擇服飾鞋包

可選規格：

☐ 簡約商務 ☐ 優雅氣質

☐ 華麗時尚 ☐ 街頭潮流

☐ 運動休閒 ☒ 小清新

☐ 甜美 ☐ 性感

取消 儲存

選商品規格

選商品規格

選商品規格

週週有新品 100 X 30 X 5 = 15,000

新品 欄位 通路

15,000 X 10 = 150,000 秒

格子 秒 2,500 分

41 小時

5 工作天

神奇的小按鈕

賣場管理

庫存

可賣量

訂單

出貨作業

退貨作業

系統設定

上架狀態

時間區間

查詢

清除查詢條件

匯出查詢結果

自動生成賣場規格

上架

匯出Y購上架檔案

876543

賣場序號

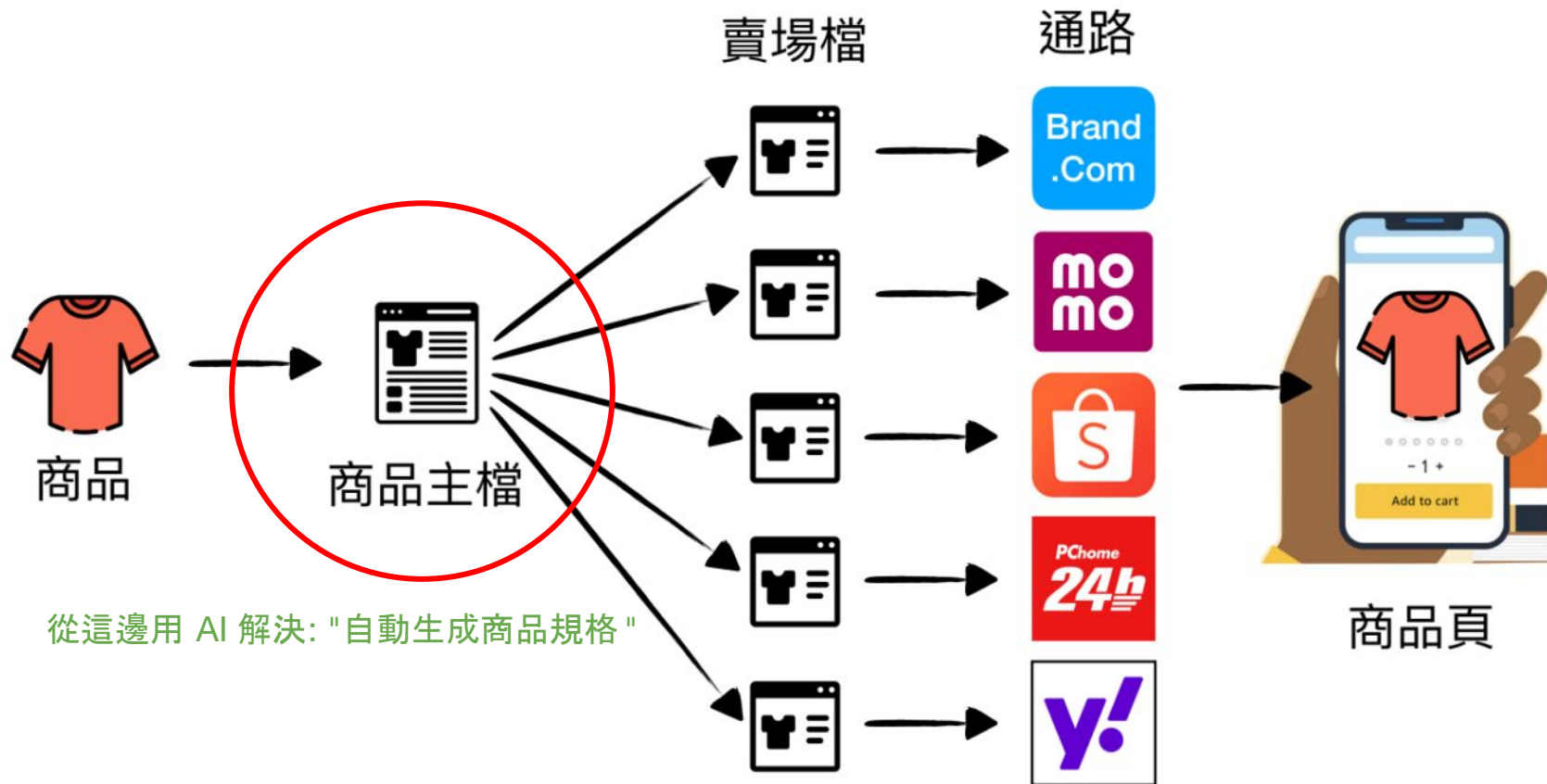
通路店

通路賣場ID

可輸入16

可輸入16

	動作	上架狀態	
<input checked="" type="checkbox"/>	<div>編輯</div> <div>刪除</div>	草稿	建議售價 售價 成本
<input checked="" type="checkbox"/>	<div>編輯</div> <div>刪除</div>	草稿	建議售價 售價 成本
<input type="checkbox"/>	<div>編輯</div>		建議售價



Q: 過去的開發流程還適用 Gen AI APP 嗎?

// 軟體的 "功能" (code) 不再是交付的唯一考量, 生成的 "資料" 品質 (prompt) 也應該要持續優化

// Code & Prompt 都可以有自己的 BizDevOps 循環

問: AI 對客戶的價值是什麼? 功能規格還是最重要的嗎?

想想看這些 "需求" 會怎麼進入開發?

1. 開功能需求 & 規格:
身為商店管理員, 在上架商品時按下 "一鍵生成賣場規格" 就能下載已包含規格的匯入檔案 ...。
2. 展開設計:
UI / Process / API / DB Schema ... 等設計規格
3. 進入執行:
拆分 Task, 評估 Size, 進入開發 ..., 測試驗收。

請問: **這功能的 DOD (Definition Of Done) 是什麼?**

// AI 最強的是處理語言的不確定性, 但是同時也難以驗證 (伴隨著幻覺) 資料生成的品質。

// 團隊成員不擅長面對 "不確定性" 的問題, 因此也沒有定義 "正確性" 相關的驗收標準 ...

// PO 沒定義期待, RD 開發沒有著墨, QA 測試不會關注正確性, 所以客戶看到的功能就 ...

這功能第一版的 PRD 這麼寫 ...

以產品功能為主的敘述，定義操作流程，匯出格式，畫面規格。

2. 專案目標

客戶可以透過「自動生成賣場規格」，針對指定的 momo 和 Yahoo 賣場，提供賣場資訊給 AI，讓 AI 根據賣場資訊生成商品規格，並產生 Excel 讓使用者檢查完，檢查完畢即可透過批次功能進入。

INPUT: 賣場資料

自動生成賣場規格 上架 匯出Y購上架檔案

賣場	動作	上架狀態
<input checked="" type="checkbox"/>  貨號: 876543765415749385	<input type="button" value="編輯"/> <input type="button" value="刪除"/>	草稿

	F	G	H	I
	通路賣場分類	規格名稱	規格限制	規格內容
動力】#服裝內著>女裝>外套>鋪棉外套		通用尺寸	複選	S,M,L
動力】#服裝內著>女裝>外套>鋪棉外套		紡織物	複選	GORE-TEX,有機棉,生化棉,記憶
動力】#服裝內著>女裝>外套>鋪棉外套		性別	複選	女性,男性,中性
動力】#服裝內著>女裝>外套>鋪棉外套		族群	複選	嬰幼兒,兒童,少年,青年,壯年,中年
動力】#服裝內著>女裝>外套>鋪棉外套		服飾鞋包	單選	運動休閒
動力】#服裝內著>女裝>外套>鋪棉外套		袖長	單選	短袖
動力】#服裝內著>女裝>外套>鋪棉外套		上身/下身	單選	合身
動力】#服裝內著>女裝>外套>鋪棉外套		顏色	複選	黑色,白色

3. 主要功能

OUTPUT: 生成的規格選項

- 自動生成賣場規格：可以勾選指定賣場，由 AI 生成該賣場的規格，生成後，使用者可以到批次作業中心下載 AI 生成完畢的 Excel 檔案。
- 批次匯入賣場規格：使用者可將 Excel 進行匯入。

開發團隊善用 AI, 很快就能完成功能開發, 但是...

Input - 賣場資料

浪漫蕾絲口袋緞面A字裙(深藍;M-XL) + 緞面材質, 滑順有光澤側邊蕾絲口袋, 精緻優雅彈性後腰 + 隱形拉鍊, 方便穿脫 + 服裝內著>女裝>下著>裙子 + 顏色:深藍,尺寸:XL,顏色:深藍,尺寸:L,顏色:深藍,尺寸:M,緞面材質, 滑順有光澤側邊蕾絲口袋, 精緻優雅彈性後腰 + 隱形拉鍊,

反正 AI 都看得懂, 我們直接使用原生格式 (誤)

Output - 自動生成的規格選項

根據提供的資料, 填入商品規格如下:

紡織物|聚酯纖維,彈性纖維;
性別|女性;
裙長|中長裙;
顏色|藍色;

為了方便, 直接讓 AI 生成原本的匯入格式 (誤)

System Prompt

只靠一句話, 結果會推論的準嗎?

根據提供賣場資料與圖片內容, 分析並填入商品規格, 並有以下需求:

若該商品規格有單選、複選、輸入框的限制, 請依照限制進行填寫:

單選: 只能填入一項。

複選: 可以填入不只一項。

輸入框: 可輸入最多50 中文字。

若該商品規格是單選和複選, 只能回答相符合表格範例提供的選項, 不可回答不在規格內容填寫範例內的選項。

寫了一大段格式的敘述,
輸出格式會 100% 正確嗎?

若賣場現有資料無法判斷商品規格要填寫的值, 請將該商品規格留空, 並繼續填寫。如果格式明確, 用 code 處理更精確更有效率。

回答格式遵守下面舉例 款式|休閒褲,縮口褲;顏色|白色系,黑色系款式為格式名稱, 休閒褲,縮口褲為分析回答, 若沒答案結尾也要加上'若為多項則使用',''分隔, 不需回答分析原因

Prompt 都專注在處理 RD 在意的 "格式", 而不是處理客戶在意的 "原則" (誤)

為何正確率這麼重要？

請問，客戶會希望哪一種結果？上面的 Prompt 會跑出哪種結果？

1. 每個選項都有答案，但是正確率約 70%

// 因為正確率不夠高，客戶每一個都要逐一確認。

// UI 如果設計的夠好，其實 review 跟 edit 的時間差別不大。全部都要看過一次等於沒省到時間

2. 只有 50% 的商品有答案，但是正確率有 90% 以上

另外 30% 的商品有提示，告訴你商品資訊缺了什麼

// 如果有填的部分正確率夠高，高到可以不需要 review，或是可以讓較沒經驗的人確認，可以先省 50%

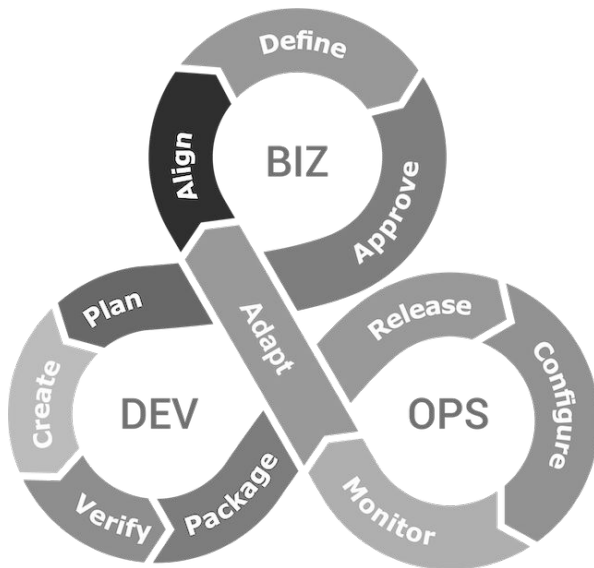
// 如果 30% 的部分已經有提示訊息，客戶本來就該補那些資訊了，有提示可以省一半，再省 15%

// 理想情況下，至少可以省下客戶 50+ % 以上的時間

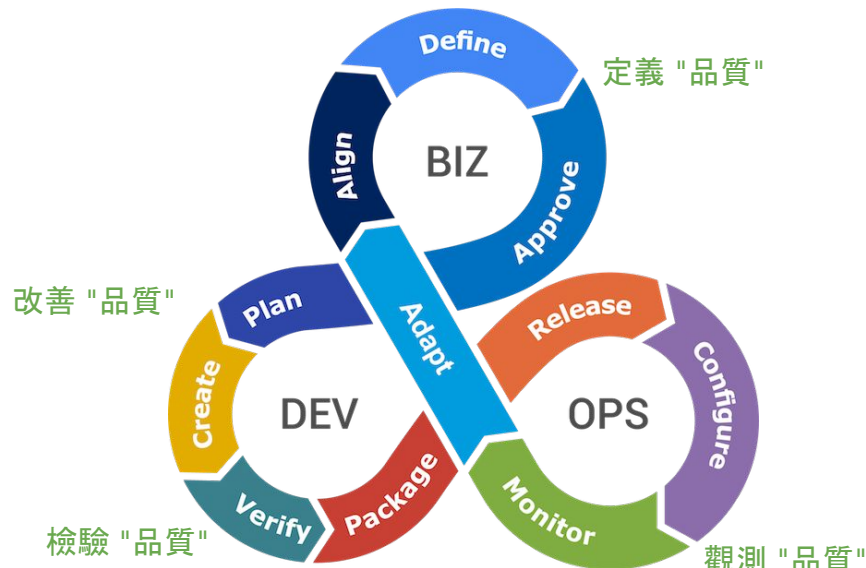
哪一個對客戶比較有幫助？

// 客戶要的是能幫助他省時間。因此我們的目標：正確率夠高，覆蓋率才有意義

Gen AI 開始普及, 所有的 APP 都要開始面對 "功能" 與 "正確" 的挑戰.. 91APP



依據 [PRD], 快速迭代
持續優化 Code 的功能性需求



依據 [參考資料], 快速迭代
持續優化 Prompt 的正確性需求

1. Evaluation - 評估量化指標
2. Prompt Improvement - 與領域專家建立改善的循環
3. Engineering Improvement - 加速改善循環, 從工程角度優化
4. GenAI Skills for Developers - 必備的開發技能訓練
5. Summary

1, Evaluation

// 以我們的案例, 不到 90% 以上的正確率, 對客戶而言跟亂猜沒兩樣。

// 有些狀況甚至沒有 "標準答案", 我們該如何定義 AI "合理" 的正確率?

// 寫 code 都知道要寫 unit test, 寫 prompt 該怎麼測試?

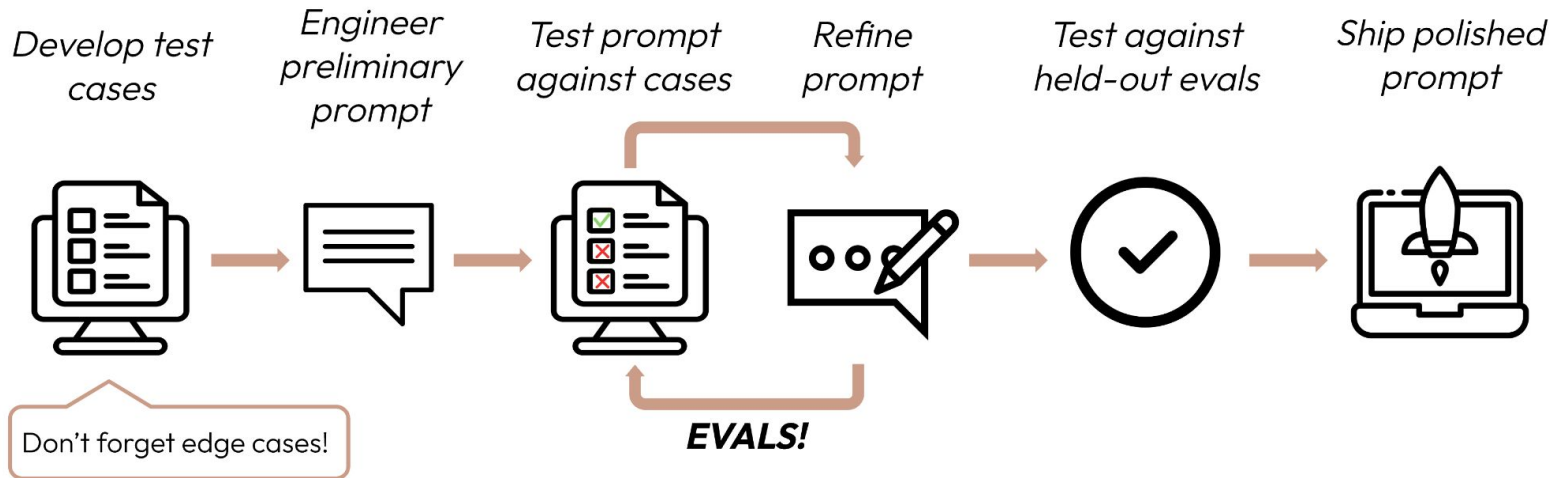
// 除非你的運氣很好, 不然不要拚運氣, 事前先做好評估

"務必" 量化正確率指標，先能掌握你的正確率後，足夠可靠再上場

91APP



如何做到 (#1, 先參考 Best Practices) ?





評估驅動開發

生成式 AI 軟體不確定性的解決方法

ihower@Hello World Dev Conference 2024/9/11

<https://ihower.tw/blog/archives/12444>

如何做到 (#2, 確認實際該執行的項目) ?

將抽象化的作業流程, 展開成實際專案內的執行項目:

1. 確認指標定義, 設定目標值, 準備參考資料集

// 專案目標是客戶效率改善。

// 效率的定義是: 正確率達標的前提下 (高於 90%), AI 生成的覆蓋率越高越好。

// 覆蓋率: $(\text{AI 有回答的總數}) / (\text{題目總數}) \times 100\%$

// 正確率: $(\text{AI 答對的總數}) / (\text{AI 有回答的總數}) \times 100\%$

2. 啟動 EVALS 循環, 驗證指標結果

// 初期資料雜訊過多: 想辦法用 AI 生成 + 人工 Review

// 已有實際資料: 過濾雜訊之後拿來運用

3. 加速迭代的循環速度

// 加速從改 code / prompt, 到能得到指標數據的速度

// 加速領域專家能判讀結果並且提供改善方向的速度

指標定義: 我們選擇了 "正確率" 與 "覆蓋率" 兩個指標來量化成效

規格選項	預期答案	AI的答對
紅色	O	O
橘色	O	O
黃色	O	O
綠色	X	X
藍色	X	X

商品 A

規格選項	預期答案	AI的答錯
紅色	O	O
橘色	O	O
黃色	O	O
綠色	X	O
藍色	X	X

商品 B

規格選項	預期答案	AI的答棄
紅色	O	
橘色	O	
黃色	O	
綠色	X	
藍色	X	

商品 C

覆蓋率 (%): 所有規格的評估結果中, AI 有回答的占比

正確率 (%): 在所有 AI 能回答的規格中, 回答正確的占比

選定 [參考資料集], 達成指標的目標值

何謂 "達標" ?

在選定的 "**參考資料集**" 上, 調整 Prompt, 直到選定的指標達到目標 值為止。

除了需要定義品質指標 (以及目標值) 之外, 也要準備參考資料集。

我們的目標: 參考資料集, 應該要 **跑出 90% 以上的正確率**, 並且產品要有對應功能協助客 戶提高覆蓋率。

參考資料集怎麼來?

使用 LLM 生成 "合理" 的假資料

- 新功能開發, 尚未累積足 夠的真實資料前
- 真實資料雜訊過多, 需要 "理想" 的資料先解決主要流程時
- 生成參考資料, 盡可能使用你能用的所有模型中, 品質最好的那個。
- *務必* 經過人工檢視, 確認資料的正確性

直接使用 "真實世界" 的資料

- 功能正式開放給客 戶使用前, *務必* 使用真實資料評估驗證
- 需要人工校正, 剔除非預期狀況的資料 (除非你希望 AI 能夠處理它們)
- 準備兩組, 分別用來優化 & 驗證用。

參考資料集: (還沒資料時) 可以考慮用 LLM 生成 + 人工確認

賣場資料：

桃粉系漸層水洗抓破大學 T (粉；S-L) + 100%純棉材質，親膚透氣貼身舒適，精緻刷破，讓穿搭更具視覺衝擊，夢幻桃粉漸層水洗甜美中帶點個性 + 服裝內著>女裝>上衣>大學T + 顏色：粉，尺寸：M，顏色：粉，尺寸：L，顏色：粉，尺寸：S，100%純棉材質，親膚透氣貼身舒適精緻刷破，讓穿搭更具視覺衝擊夢幻桃粉漸層水洗甜美中帶點車細節，穿出輕鬆自在上衣 | 3243

| 3243122407 上衣 | 3243321326

| 3243398619 褲裝 | 3244136306

參考答案 (用 AI 生成)：

```
"name": "紡織物",
"suggest": [
  { "opt": "棉布", "score": 90, "comment": "商品標示 100%純棉材質" },
  { "opt": "GORE-TEX", "score": 0 },
  { "opt": "有機棉", "score": 0 },
  { "opt": "棉麻", "score": 0 },
]

"name": "袖長",
"suggest": [
  { "opt": "長袖", "score": 90, "comment": "大學T通常為長袖" }
```

不要只靠 AI 生成 "正確答案"。

為了方便調整與人工確認，每個選項逐一生成 "信心指數" + "原因說明"：

1. 提高合理性
2. 提高調整彈性
3. 提高人工確認的效率

```
{ "opt": "桃色", "score": 70, "comment": "商品呈現桃粉系漸層" },
{ "opt": "黑色", "score": 0 },
{ "opt": "白色", "score": 0 },
...

```


2, Prompt Improvement

// 從評估結果, 決定改善方向

// 最重要的事: 建立技術專家 (架構師) 與領域專家 (大PO) 的協作

跑出指標分數了，那麼接下來我該做甚麼？

1. 先從數據的分布，猜測可能的狀況

// 數據的分布狀況合理嗎？

// 大範圍：調整 Prompt 寫法，調整各種參數 threshold 數值，例如信心指數

2. 翻出每一筆資料的處理紀錄，診斷問題

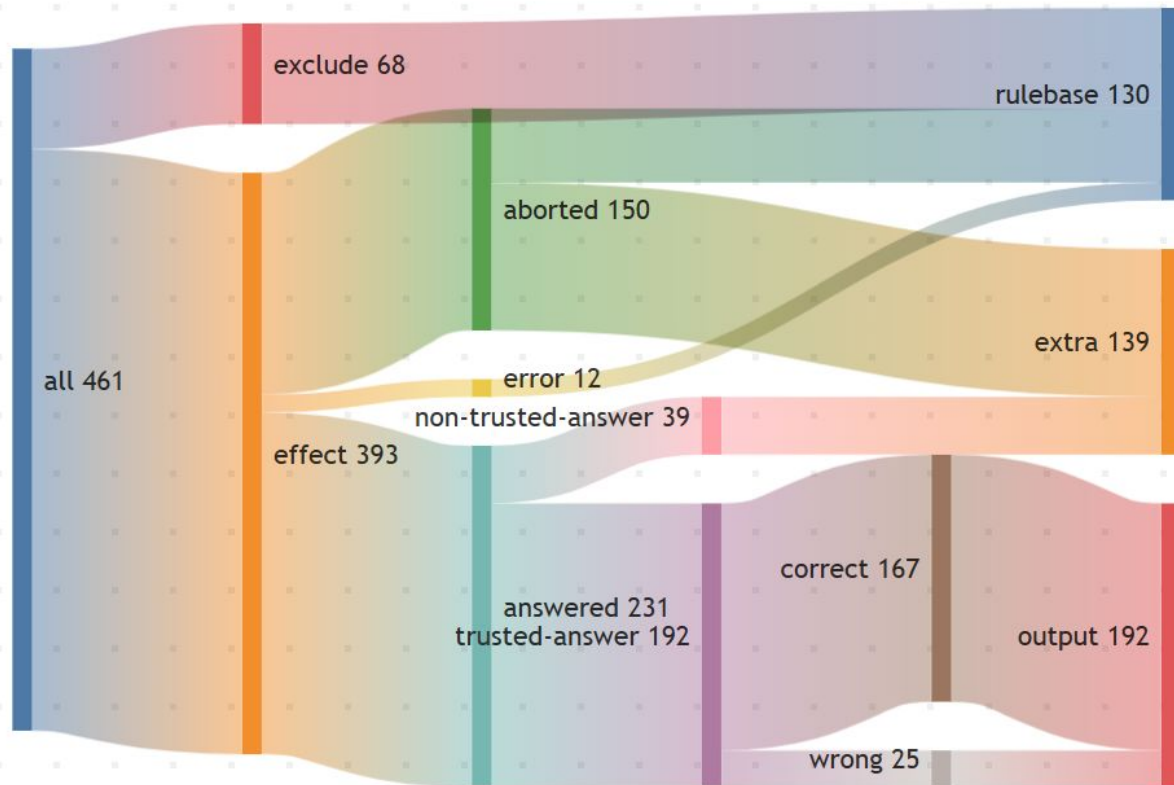
// 逐筆觀察，歸納處理原則，用原則修正 Prompt

// 排除特例，用 Rulebase 處理特例（別用 Prompt 處理特例）

3. 建立技術專家 & 領域專家的協作

// 初期：由我（架構師）與 大PO 協作，定下處理原則

// 後期：由 Tech Lead 與 Product Owner 協作，持續優化專案



用規則處理 (Rule):

- // 有淺規則的
- // 各種不適合 AI 處理的
- // 用 Rule 預先排除 (越大越好)

提示客戶補充資訊 (Hint):

- // 各種原因無法回答的區塊
- // 需要附上註記
- // (註記越清楚越好)

給客戶建議答案 (Autofill):

- // AI 能回答的區塊
- // 錯誤的數字越低越好
- // (盡可能將 wrong 區塊移轉)

領域專家參與的重要性 - 案例: "緞面材質", 該給什麼?

賣場資料 :

浪漫蕾絲口袋緞面 A 字裙 (深藍 ; M-XL) + 緞面材質, 滑順有光澤, 側邊蕾絲口袋, 精緻優雅, 彈性後腰 + 隱形拉鍊, 方便穿脫 + 服裝內著 > 女裝 > 下著 > 裙子 + 顏色 : 深藍, 尺寸 : XL, 顏色 : 深藍, 尺寸 : L, 顏色 : 深藍, 尺寸 : M, 緞面材質, 滑順有光澤。側邊蕾絲口袋, 精緻優雅, 彈性後腰 + 隱形拉鍊, 方便穿脫, A 字裙擺, 修飾身形, 針織 | 1243465001

規格名稱 : 紡織物

規格限制 : 複選

規格內容填寫範例 :

GORE-TEX, 有機棉, 棉布, 棉麻, 不織布, 丹寧布, 牛津布, 帆布, 防水布, 網布, 蠶絲, 紗, 絨布, 莫代爾, 萊卡, 嫻縐, 尼龍, 聚酯纖維, 彈性纖維, 羊毛, 皮革, 無, 其它

// 哪個才是正確答案?

// 我們要的是靠 AI 用前後文合理的推論, 還是實際上廠商的資料為主?

// (可能根本沒有足夠資料可以推論)

// 我們需要的是 "領域專家" 判定問題, 搭配 "AI 專家" 來調整 Prompt ..

#← Also sent to the channel



Andrew Wu [UPD] 吳剛志 Apr 9th at 11:58

1. 包包/鞋子，正確答案是 "專櫃"，AI 答案是 "運動品牌"，我自己看商品敘述也會選 "運動品牌"，這些選項是兩個不同面向，一個是產品屬性，一個是經營通路。是否要決定一種方向來讓 AI 回答
2. 鞋子(日規)，正確答案包含 21cm, 21.5cm, AI 回答沒有這兩項。直接看商品敘述，AI 回答是對的。請確認是否需要修正

#← Also sent to the channel



VIP Apr 9th at 13:48

先說前提，目前核心是確認AI效果，還有整理標準資料集，所以：

1. 人為的答案是都可以，但還是要做個決定，我覺得單選有多個答案是不填，回到填錯成本比沒填高的原則
2. 正確答案是 "專櫃"，AI 答案是 "運動品牌" => 我觀察有一些低品質的屬性，可能要討論一下，用rule base來處理這種沒什麼道理的屬性
3. 顏色: 標準答案是 "黑色"，AI 回答 "灰色"。商品名稱明確提到 [redacted] 女款灰色DWR防潑水連身裙"，請確認上架資料的正確性 => 我們現況的目的是做出足以拿來長期評估prompt成果的資料集，我覺得應該是人為事後校正的比較對，就是灰色設為答案，前面21cm也應該比照

從領域專家的回饋, 找出改善的範圍與實施方式 (AI, 工程, 資料集)

1. 正確率 (主要都是優化 AI 的處理與判定):

最有效的方式, 是降低答錯的區塊, 不是提高答對的區塊

// ⇒ 剔除不確定的 (prompt 要求, 輸出信心指數)

// ⇒ 輸出理由 (強迫 AI 思考, 方便人工 review, 可用於給使用者回饋)

// ⇒ 用正確的格式輸入輸出 (markdown, json 是好選擇)

2. 覆蓋率 (主要都是優化 產品的功能與流程設計):

// ⇒ 用規則 "事前" 過濾不需要 AI 處理的部分 (主要是來自客戶的 "潛規則")

// ⇒ 用規則 "事後" 修正 AI 回答不周全的部分

// ⇒ 使用產品流程設計, 引導客戶 (給理由彙整, 讓客戶補充賣場檔) 修正資料

3. 修正資料集的品質 (主要都是優化參考資料集):

// ⇒ 手動排除不符合常理的參考資料

// ⇒ 避免誤導 Prompt 的判定原則, 讓量化指標公平客觀, 有代表性

3, Engineering Improvement

// 最重要的事: 加速改善循環, 用對技術來改善循環的瓶頸

// 評估: 每次調整 prompt 後多久可以知道改善結果?

// 我 (架構師) 陪著團隊跑了第 1~10 次循環, 建立執行環境與流程

// 跟領域專家 (大PO) 跑了第 11 ~ 13 次, 建立問題判定與改善方向決策流程

// 團隊自己能跑完剩下的循環 (~20)

從工程角度, 也能做出改善..

1. 改善 AI 本身的成效, 輸入輸出的結構調整優化 (+5%)

// 選擇正確的輸入格式 (markdown), 提高 AI 理解能力。

// 選擇正確的輸出格式 (json + schema), 用 schema 替代 prompt 來說明格式會更精準。

// Json Mode 是先進 LLM 的重要功能, 各大廠重點優化項目, 絕對比你自已寫 prompt 來的強

2. 改善 工程師 的執行效率 (4hr → 10m), 修正 & 跑分

// 選擇正確的工具, 善用 No Code + LLM 加速驗證

// AI 的 code 大部分都是改 LLM 參數 + 組成 prompt 內容

// 期待: 能快速調整 prompt + LLM 參數

// 期待: 能快點看到指標結果與評估報告

3. 改善 PO / 領域專家 的執行效率 (8hr → 30m), 檢視報告 & 決策改善方式

// 輸出正確的報告, 加速領域專家的溝通與判讀問題

// 總表 (csv), 指標數值的計算 (excel 公式)

// 評分紀錄 (markdown) (正確 | AI答案) (註記), 原始資料 (商品資訊, 規格定義) 連結

// 統計數據分布 (mermaid sankey)

調整適合 AI / CODE 的輸入輸出結構

91APP

Input - 賣場資料

浪漫蕾絲口袋緞面A字裙(深藍;M-XL) + 緞面材質, 滑順有光澤側邊蕾絲口袋, 精緻優雅彈性後腰+隱形拉鍊, 方便穿脫 + 服裝內著>女裝>下著>裙子 + 顏色:深藍,尺寸:XL,顏色:深藍,尺寸:L,顏色:深藍,尺寸:M, 緞面材質, 滑順有光澤側邊蕾絲口袋, 精緻優雅彈性後腰+隱形拉鍊, 方便穿脫 A字裙擺, 修飾身形針織 | 1243465001

Input - 賣場資料

ProductID
1243465001

SalePageName
浪漫蕾絲口袋緞面A字裙(深藍;M-XL)

ProductDescription
緞面材質, 滑順有光澤側邊蕾絲口袋, 精緻優雅彈性後腰
隱形拉鍊, 方便穿脫
A字裙擺, 修飾身形

輸入格式改用 Markdown, AI 更能理解文字的意義與 內容

Output - 自動生成的規格選項

根據提供的資料, 填入商品規格如下:

紡織物|聚酯纖維,彈性纖維;
性別|女性;
裙長|中長裙;
顏色|藍色;

Output - 自動生成的規格選項

```
{  
  "json": {  
    "specs": [  
      {  
        "opt": "紡織物",  
        "values": ["聚酯纖維", "彈性纖維"],  
        "sure": true,  
        "reason": "商品描述中提到緞面材質及彈性後腰, 符合規格選項。"  
      }  
    ]  
  }  
}
```

輸出改成 Json + Schema, 格式更精準, 且更容易讓程式做後續處理, 正確率 +5%

調整 system prompt, 聚焦回答的原則, 而非格式的處理

System Prompt

根據提供賣場資料與圖片內容, 分析並填入商品規格, 並有以下需求:

若該商品規格有單選、複選、輸入框的限制, 請依照限制進行填寫:

單選: 只能填入一項。

複選: 可以填入不只一項。

輸入框: 可輸入最多50 中文字。

若該商品規格是單選和複選, 只能回答規格內容填寫範例提供的選項, 不可回答不在規格內容填寫範例內的選項。

若賣場現有資料無法判斷商品規格要填寫的值, 請將該商品規格留空, 並繼續填寫其他商品規格。

回答格式遵守下面舉例 款式|休閒褲,縮口褲;顏色|白色系,黑色系款式為格式名稱, 休閒褲,縮口褲為分析回答, 若沒答案結尾也要加上'若為多項則使用','分隔, 不需回答分析原因

system prompt

我需要你從商品敘述, 從規格範例中, 推論出規格選項。請按照我提供的賣場資訊, 以及商品規格要求說明, 幫我對每個規格名稱推論適合的規格選項

salepage info

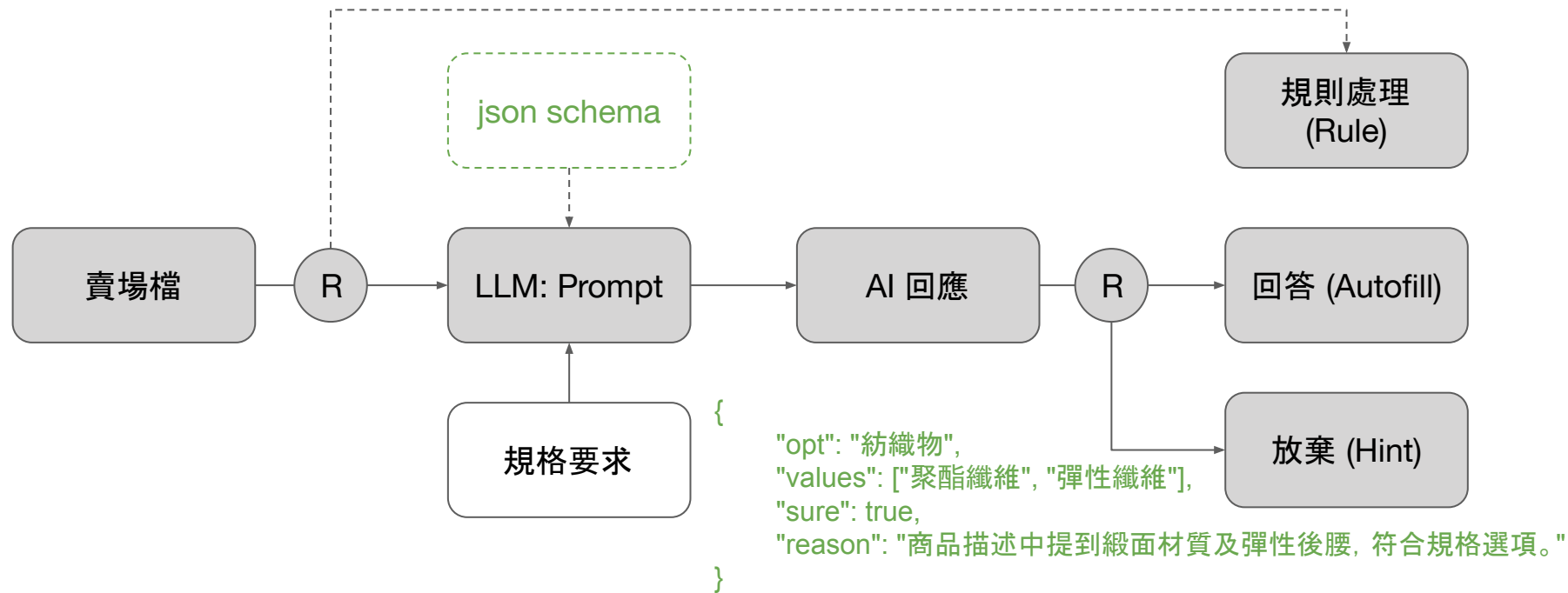
product specs

answer rules

1. ...
2. ...
3. ...
4. ... 請優先將完全相符的規格選項納入預期回答結果。
5. ... 若不完全相符, 就放棄作答, 回報原因。
6. ... 請推論規格選項的意義, 選出與商品描述同義含廣義的規格選項。
7. ...
8. 若沒有任何規格選項符合要求 或是你判定資訊不足, 請給我空集合。

明確的說明推論規格的原則, 將 PO 回饋的處理原則變成 Prompt

核心: 正確設計 Prompt 的回應格式, 配合 Code 精準的解決問題



Diffy 加速 AI 應用的開發與驗證流程, NoCode 更適合作為 Playground. 91APP

The image displays the Diffy AI playground interface, which is a NoCode environment for developing and testing AI applications. The interface is divided into several sections:

- Workflow:** A visual flowchart showing the process from **START** (with a required input `sp_info`) to **LLM** (using `gpt-4o-mini` in `CHAT` mode) and **CODE** (with a `result` object), leading to **END**.
- LLM Configuration:** A panel for configuring the `gpt-4o-mini` model. It includes a **MODEL** dropdown, a **CONTEXT** field (set to `Set variable`), and a **SYSTEM** prompt (100 characters, Jinja template). The **PARAMETERS** section allows adjusting various settings:
 - Temperature:** 0.2
 - Top P:** 0.75
 - Presence Penalty:** 0.5
 - Frequency Penalty:** 0.5
 - Max Tokens:** 512
 - Response Format:** `json_schema`
- Test Run #15:** A panel showing the output of the test run. It includes tabs for **INPUT**, **RESULT**, **DETAIL**, and **TRACING**. The **OUTPUT** section displays the JSON result:

```
1 {
2   "result": {
3     "性別": {
4       "sure": true,
5       "reason": "商品描述中提到是男性運動短褲，明確指出性別為男性。",
6       "values": [
7         "男性"
8       ]
9     },
10    "袖長": {
11      "sure": false,
12      "value": "無袖",
13      "reason": "商品描述未提及袖長，因為是短褲，所以無法確定。"
14    },
15    "顏色": {
16      "sure": false,
17      "reason": "商品描述中沒有提供顏色選項的資訊，因此無法推斷顏色。",
18      "values": []
19    }
20  }
21 }
```

快速驗證 LLM 的參數與 Prompt, 每次調整後可以立即驗證成果 (4hr → 10min)。

每次評估都能快速產生報告 (1/3) - 總表 .csv

91APP

	A	B	D	F	H	I	J	K	L	M	N
1	sale_page_id	spec_name	status	final_scc	sure	reason					
2	1093504	鞋子(美規)	Warning	0	TRUE	賣場信息中没有对美规的说明，因此没有依据来选择相应的美规尺寸。					
3	1093504	鞋底	Warning	0	TRUE	商品描述中只有提到'橡膠大底'，没有具体说明是厚底还是平底等选项，因此无法选择					
4	1093504	鞋面	Available	0	TRUE	商品描述明确提到'皮革鞋面'，与'真皮'对应，因此确定选择'真皮'。					
5	1093504	包包/鞋子	Available	100	TRUE	商品名稱和品牌名稱中的'adidas'以及'運動休閒鞋'都明顯表達這是一個運動品牌產品，					
6	1093504	性別	Available	66.666667	FALSE	商品名稱和分类名称都有提及男鞋，但因為商品選項中也有多個尺寸適合女性使用，因					
7	1093504	鞋子	Warning	0	TRUE	商品描述中缺乏關於耐磨、彈力等特徵的信息，因此無法從中推斷任何對應的鞋子規格					
8	1093504	鞋子(日規)	Available	100	TRUE	賣場商品選項列出了多個日規尺寸，包括22cm到30.5cm，因此這些可以確定地選擇。					
9	1093504	鞋子(歐規)	Warning	0	TRUE	賣場信息中没有对欧规的任何说明，因此无法对应选择任何欧规的尺寸。					
10	1190063	鞋襪	Available	100	TRUE	商品名稱和描述中都明確提到中筒襪，因此這是確定的選項。					
11	1190063	顏色	Warning	0	TRUE	賣場資訊中未從任何地方指出或展示襪子的顏色，無法從中推論出規格選項。					
12	1190063	服飾	Error	0	FALSE	賣場資訊中提到此商品適合中性，也未限定性別，但有提到男/女，因此推論可能適合女					
13	1190063	紡織物	Available	100	TRUE	賣場資訊明確描述了襪子材質為65%棉、32%再生聚酯纖維和2%彈性纖維，這些材料在					
14	1190063	襪子	Warning	0	TRUE	賣場資訊未提供任何尺寸數據來對應於規格說明的選項範疇，故無法推論出規格選項。					

加速溝通，產生合適的報表讓 PO 能快速判定結果，決定該如何優化 (8hr → 30min)

每次評估都能快速產生報告 (2/3) - 評分紀錄 .markdown

1093504-4o-Report.md

Preview 1093504-4o-Report.md

1169040-4o-Report.md

116903

Spec: 包包/鞋子

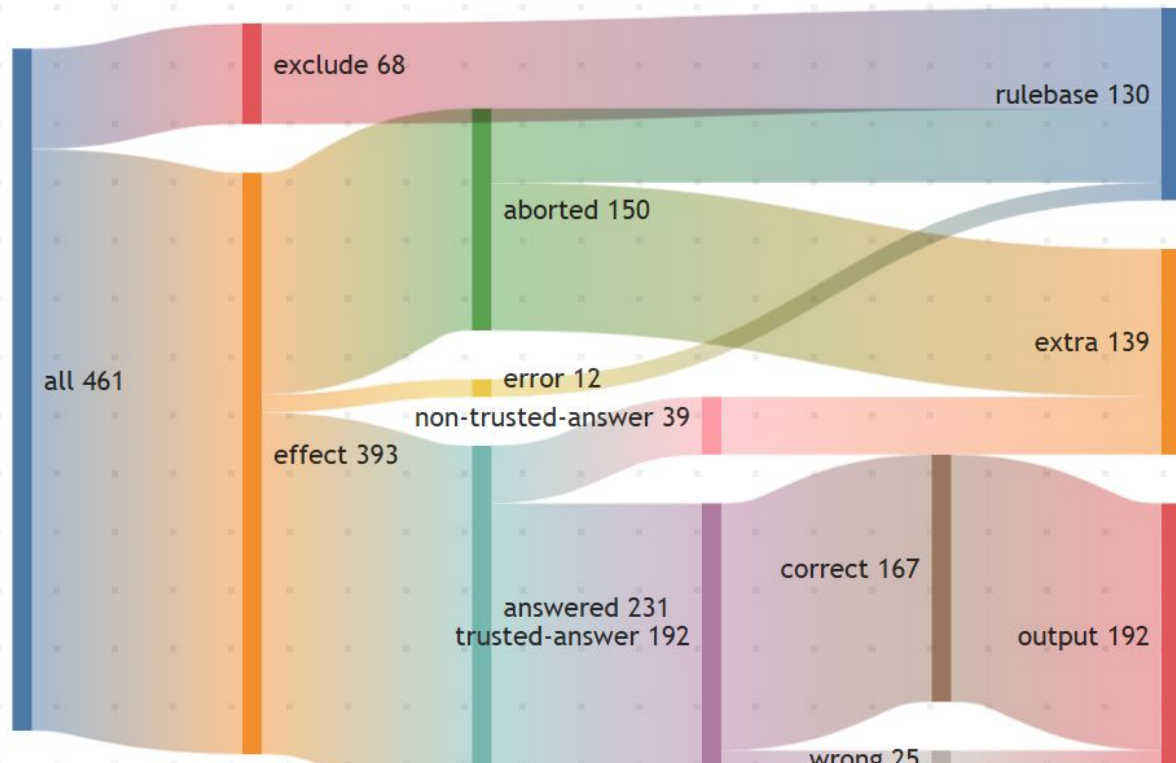
賣場資訊: [1093504-info.md](#)

賣場規格: [1093504-spec.md](#)

參考資料: AI 合成資料 | 真實資料

規格	標準答案	測試答案	得分	信心指數(標準答案的)
運動品牌	O	O	100	100
專櫃	X	X	0	0
流行	X	X	0	0
休閒	X	X	0	0
歐系精品	X	X	0	0
美日精品	X	X	0	0

加速溝通，產生合適的報表讓 PO 能快速判定結果，決定該如何優化（8hr → 30min）



用規則處理 (Rule):

- // 有淺規則的
- // 各種不適合 AI 處理的
- // 用 Rule 預先排除 (越大越好)

提示客戶補充資訊 (Hint):

- // 各種原因無法回答的區塊
- // 需要附上註記
- // (註記越清楚越好)

給客戶建議答案 (Autofill):

- // AI 能回答的區塊
- // 錯誤的數字越低越好
- // (盡可能將 wrong 區塊移轉)

加速溝通，產生合適的報表讓 PO 能快速判定結果，決定該如何優化 (8hr → 30min)

4, GenAI Skill for Developers

// 收到上面的設計跟規劃, 身為 Developer, 你是否有能力實作出來?

// 要 Vibe Coding, 你至少要會開規格, 並且有能力驗證程式碼

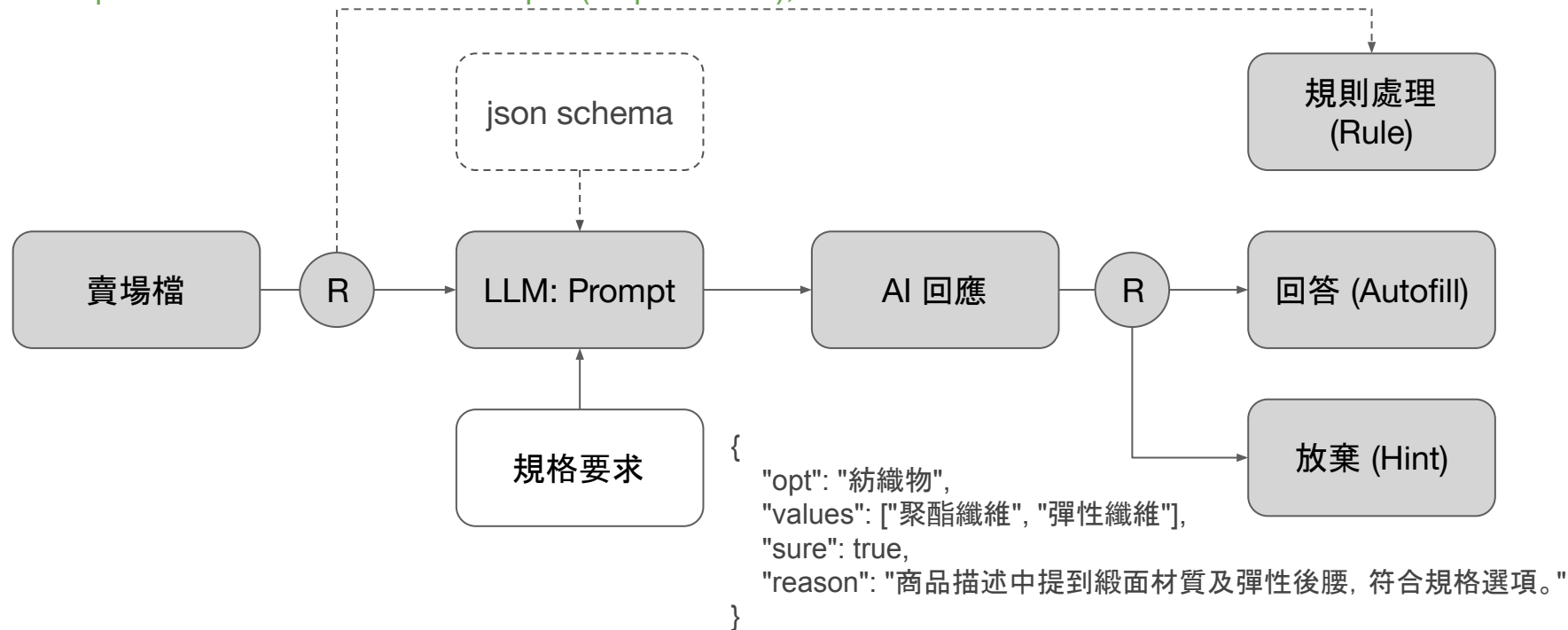
// 掌握三個關鍵操作: Chat Completion, Json Mode, Function Calling

核心: 正確設計 Prompt 的回應格式, 配合 Code 精準的解決問題

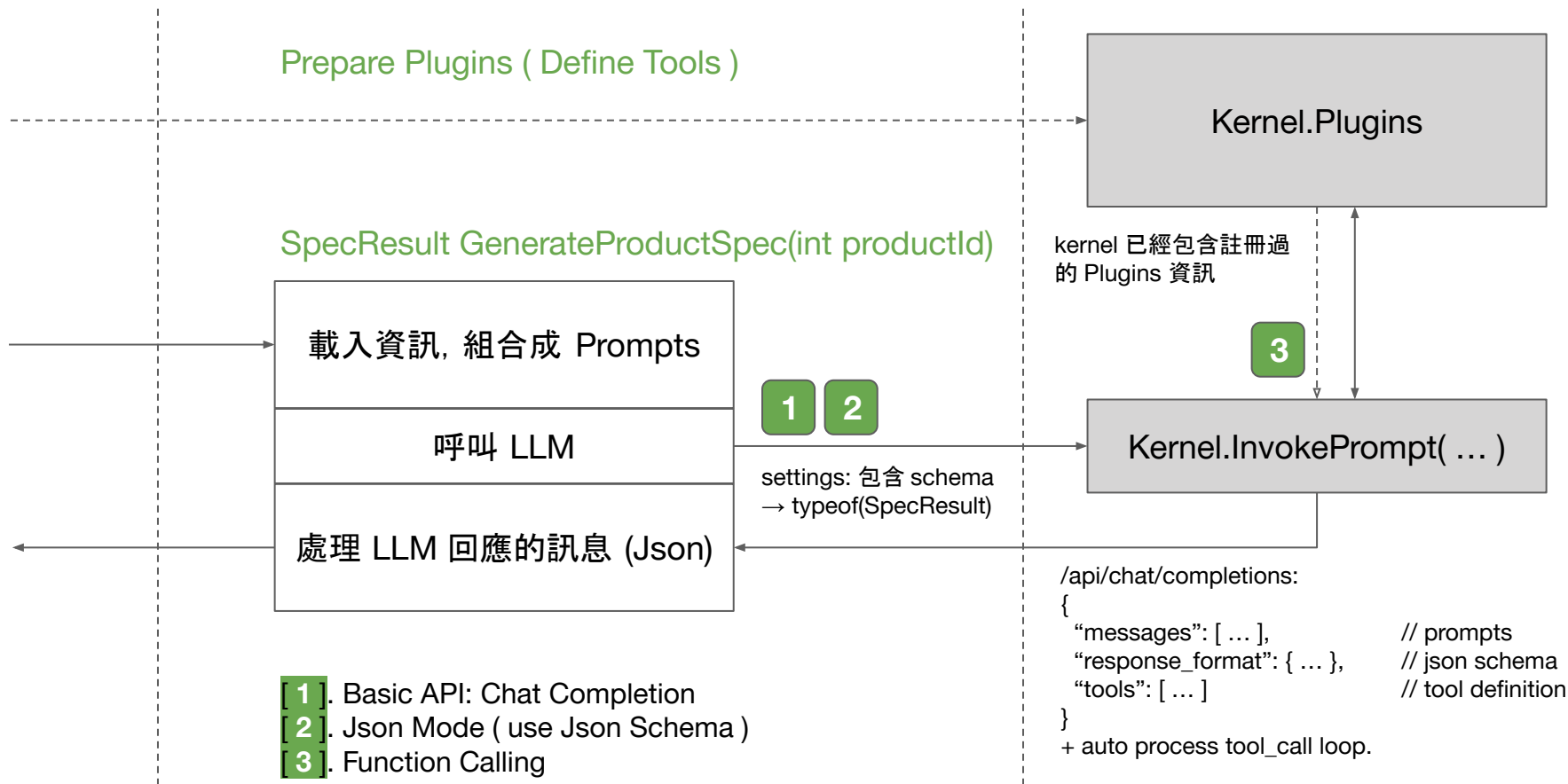
// 封裝與抽象化:

// Prompt 已經是 AI 時代的核心邏輯了, 把它當作系統內的關鍵元件, 他會需要明確的 input / output ...

// SpecResult GenerateProductSpec(int productId);



拆解 AI Function: SpecResult GenerateProductSpec(int productId);



參考: DevOpsDays Taipei 2024, 從 API First 到 AI First

LLM 的 API 長這樣子:

```
1 curl https://api.openai.com/v1/chat/completions \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "model": "gpt-4o-mini",
6     "messages": [{"role": "user", "content": "Say this is a test!"}],
7     "temperature": 0.7
8   }'
```

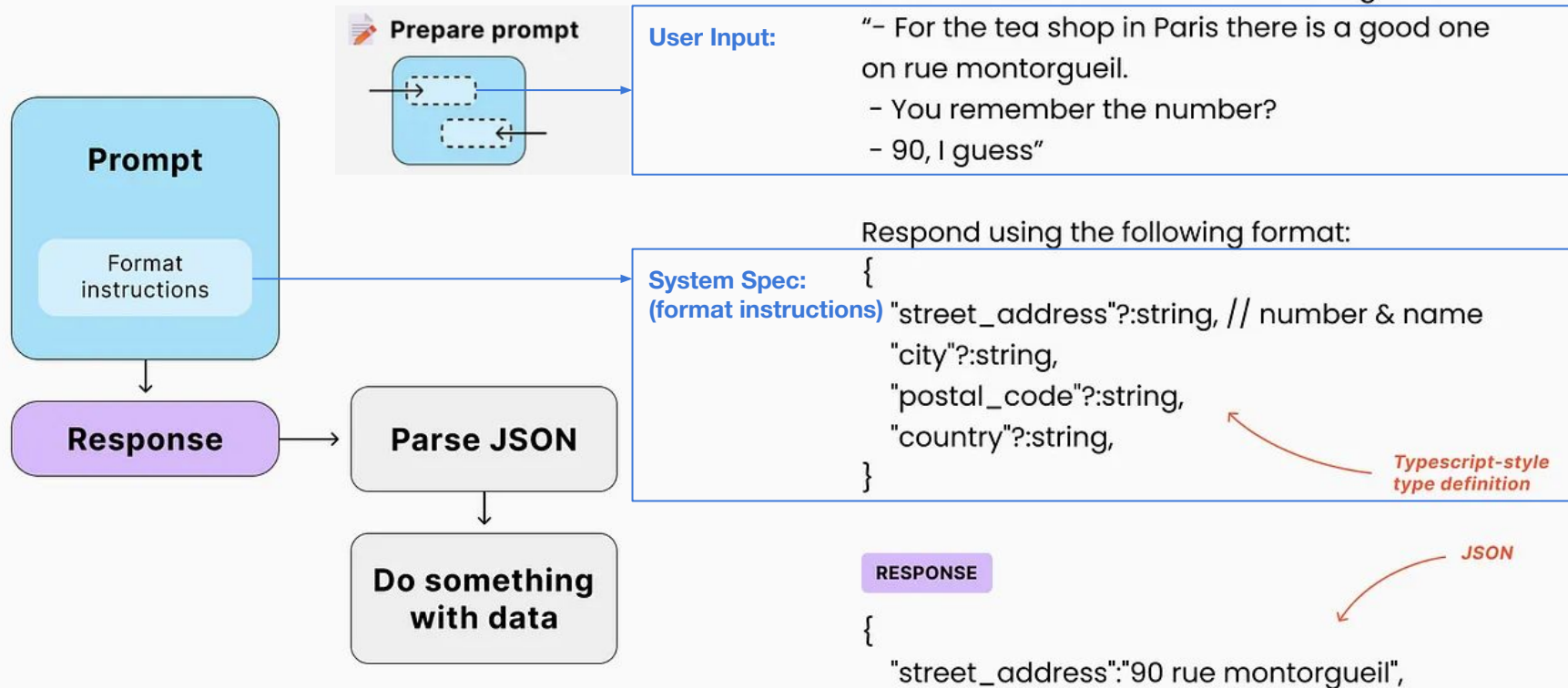
若要繼續對話, 你需要連同前後文一起傳遞:

```
2   "id": "chatcmpl-abc123",
3   "object": "chat.completion",
4   "created": 1677858242,
5   "model": "gpt-4o-mini",
6   "usage": {
7     "prompt_tokens": 13,
8     "completion_tokens": 7,
9     "total_tokens": 20,
10    "completion_tokens_details": {
11      "reasoning_tokens": 0,
12      "accepted_prediction_tokens": 0,
13      "rejected_prediction_tokens": 0
14    }
15  },
16  "choices": [
17    {
18      "message": {
19        "role": "assistant",
20        "content": "\n\nThis is a test!"
21      },
22      "logprobs": null,
23      "finish_reason": "stop",
24      "index": 0
25    }
26  ]
27 }
```

DEMO: Example01_SimpleChat (x3)

<https://platform.openai.com/docs/api-reference/making-requests?lang=curl>

JSON completion



// JSON mode 是串聯 "自然語言" 與 "程式語言" 的重要機制

程式語言

```
27     modelId: "gpt-4o-mini",  
28     apiKey: OPENAI_APIKEY,  
    httpClient: HttpClient.GetHttpClient(true));
```

```
    var kernel = builder.Build();  
    var settings = new OpenAIPromptExecutionSettings()  
    {  
        ResponseFormat = typeof(Address);  
    };  
    
```

```
32  
33  
34  
35  
36  
37  
38  
39  
40  
    var result = await kernel.InvokePromptAsync(  
        """,  
        <message role="system">  
            Extract the address from the following text.  
        </message>  
        <message role="user">  
            - For the tea shop in Paris there is a good one on rue montorgueil.  
            - You remember the number?  
            - 90, I guess.  
        </message>  
        """,  
        new(settings));
```

```
44  
45  
46  
47  
48  
49  
50  
51  
52  
    var address = JsonSerializer.Deserialize<Address>(result.ToString());  
    Console.WriteLine($"Extract the Address from conversation:");  
    Console.WriteLine($"- Street: {address?.Street}");  
    Console.WriteLine($"- City: {address?.City}");  
    Console.WriteLine($"- Postal Code: {address?.PostalCode}");  
    Console.WriteLine($"- Country: {address?.Country}");
```

```
public class Address  
{  
    [JsonPropertyName("street_address")]  
    1 reference | Andrew Wu, 67 days ago | 1 author, 1 change  
    public string? Street { get; set; }  
  
    [JsonPropertyName("city")]  
    1 reference | Andrew Wu, 67 days ago | 1 author, 1 change  
    public string? City { get; set; }  
  
    [JsonPropertyName("country")]  
    1 reference | Andrew Wu, 67 days ago | 1 author, 1 change  
    public string? Country { get; set; }  
  
    [JsonPropertyName("postal_code")]  
    1 reference | Andrew Wu, 67 days ago | 1 author, 1 change  
    public string? PostalCode { get; set; }  
}
```

不需要在 Prompt 去敘述輸出格式
直接定義 ResponseFormat (用 C# Entity 來定義)

自然語言

程式語言

```

58  Send request | Debug
59  POST https://api.openai.com/v1/chat/completions
60  Content-Type: application/json
61  Authorization: Bearer {{OpenAI_APIKEY}}
62
63  {
64    "model": "gpt-4o-mini",
65    "messages": [
66      {
67        "role": "system",
68        "content": "Extract the address from the following text"
69      },
70      {
71        "role": "user",
72        "content": "'- For the tea shop in Paris there is a good one on rue montorgueil.\n- You remember the number?\n- 90, I guess.'"
73      }
74    ],
75    "temperature": 0.2,
76    "response_format":
77    {
78      "type": "json_schema",
79      "json_schema":
80      {
81        "name": "address_format",
82        "schema":
83        {
84          "type": "object",
85          "properties":
86          {
87            "street_address":
88            {
89              "type": "string",
90              "description": "Number and name c
91            },
92            "city":
93            {
94              "type": "string",
95              "description": "Name of the city.

```

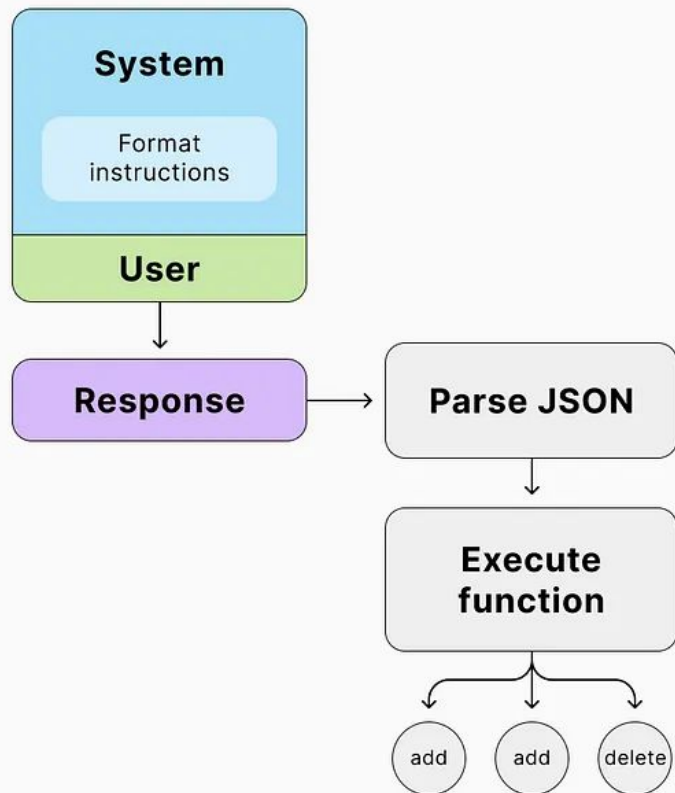
不需要在 Prompt 去敘述輸出格式
直接定義 ResponseFormat (用 Json Schema 來定義)

```

{
  "id": "chatcmpl-BcE901VuFN6wKWJ2Ra65I10jXpNH",
  "object": "chat.completion",
  "created": 1748450534,
  "model": "gpt-4o-mini-2024-07-18",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "{\n  \"street_address\": \"90 rue montorgueil\",\n  \"city\": \"Paris\",\n  \"postal_code\": \"\",\n  \"country\": \"France\"\n}",
        "refusal": null,
        "annotations": []
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 131,
    "completion_tokens": 21,
    "total_tokens": 152,
    "prompt_tokens_details": {

```

Function calling



SYSTEM

Based on the following conversation, manage the shopping list. Write your response in JSON using the following format:

```
[  
  { "action": "add", "item": string, "quantity": string },  
  { "action": "delete", "item": string }  
]
```

USER

Mmm, remember to buy some butter and a pair of zucchinis. But I already bought bread.
櫛瓜、西葫蘆

RESPONSE

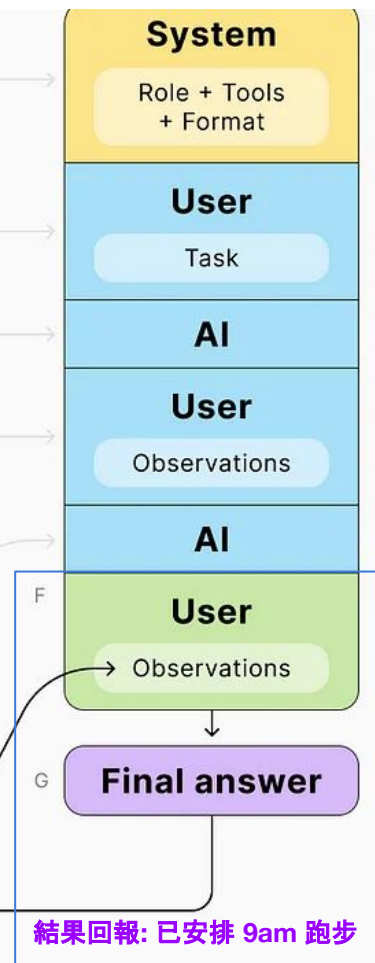
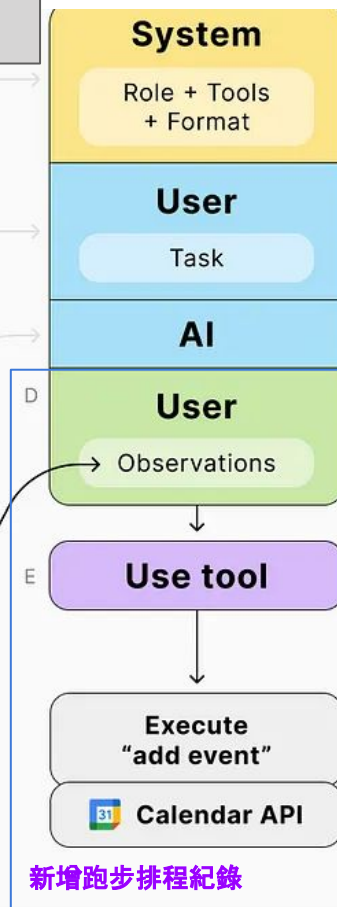
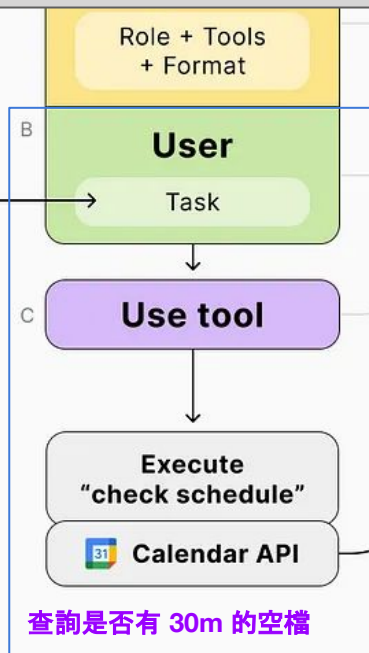
```
[  
  { "action": "add", "item": "butter", "quantity": "1" },  
  { "action": "add", "item": "zucchinis", "quantity": "2" },  
  { "action": "delete", "item": "bread" }  
]
```




"Find a 30 min slot
for a run tomorrow
morning"



"Morning run scheduled
for tomorrow at 9am!"



.NET RAG 神器 - Microsoft Kernel Memory 與 Semantic Kernel 整合應用

📅 2025/06/22 📁 系列文章: 架構師觀點 🔖 架構師觀點 🔖 技術隨筆



讚

分享

成為朋友中第一個說這個讚的人

今年三月, 跟保哥開了一場直播, 從 LLM 的基礎 (OpenAI ChatCompletion API 開始), 談基本的 API 操作。使用方式從 HttpClient, 到 OpenAI .NET SDK, 再到 Microsoft Semantic Kernel, 示範了 Chat, Json Mode, 到 Function Calling 的操作, 最後示範了用 Microsoft Kernel Memory 這套服務來實作 RAG ...



Facebook Pages



5, Summary

// 功能規劃的心得

// 開發與改善循環的心得

// 軟體工程師的下一步

產品化，重點在做出對客戶的價值

對客戶而言，這是個輔助功能，不是 AI 炫技的地方

不是所有 AI Agent 都是 "對話" 型態的應用。做對方式，也能對客戶有大助益。

客戶真正在意的期待，要有對應的指標來量化：

效率 = 減少客戶需要手動輸入的資料數量 + 減少客戶需要檢查的資料數量

- // 要讓客戶放心，正確率就要夠高 (所以必須把關 正確率 分數, 至少高於 90%)
- // 要知道客戶能省多少時間，我們要知道 AI 能回答多少問題 (所以必須預估 覆蓋率)
- // 有些問題可以有明確規則，可以讓客戶自己安排，不需要為難 AI (所以要有 Rulebase 設計)
- // 有些情況客戶真的漏掉必要的商品資訊，藉由規格來提醒客戶 (所以要 UX + AI Feedback 的設計)
- // 老闆的期待：90% 以上的正確率，90% 以上的覆蓋率，替客戶省下 90% 的時間

1. **(Scope) 界定處理範圍, 把 AI 用在刀口上。**

- // 你的 Prompt 會被執行 1,000,000+ 次
- // 用程式碼處理明確問題, 會比 AI 處理來的好

2. **(Evaluate) 掌控正確性。**

- // 將目標換成能量化的指標, 建立評估的機制
- // 掌握 "選擇題" 與 "問答題" 的評估能力

3. **(Optimize Improvement Cycle) 加速循環。**

- // 評估 → 改善的循環速度, 直接影響你優化的能力
- // 建立評估循環比開發功能還重要, 必要時這邊也需要投入開發人力

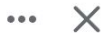
4. **(Prompt Engineering) 學習正確使用 LLM。**

- // 工程師應掌握 LLM API 的正確使用技巧
- // Json Mode, Function Calling, Prompt Engineering 是必要技巧



哈利說 · 追蹤

6月22日下午2:59 · 地球



91APP

Sam Altman:「模型能力已經走到很前面了，但大家還沒有真的把這些能力Build 進產品中。現在的機會十分巨大。」

你如果真的瞭解現在前沿模型的能力以及效率提升的速度，你就會發現其實一般大眾平時使用的絕大多數產品，全都是過時的。而每一個過時的產品，都是一個新的 AI 創業的機會。



#AI焦慮 - 軟體工程師的下一步？

1. 軟體工程師會被取代嗎？

AI coding 會越來越厲害，需要 "手寫" 程式碼的機會越來越少
有能力 "生產" code 的門檻越來越低，非工程人員也開始能用程式碼滿足自己的需求
⇒ "現在世代的軟體"，開發會越來越容易，門檻會越來越低。

2. 懂得開發 AI 應用的工程師，未來需求會越來越大

改朝換代，都需要大量人力重建系統。是否具備 "新世代軟體開發能力"，是競爭力的關鍵。
如果你認為 Cursor 對你影響很大，想看看：
⇒ 什麼樣的人寫得出 Cursor / Windsurf 這樣的程式？
⇒ 其他行業的 "Cursor" 是什麼？
⇒ 你需要俱備什麼樣的能力才能開發出來？

3. 你應該開始準備的基礎能力：

基礎工程技巧 ⇒ 有系統的優化正確性，與領域專家協作的的能力，提升改善的循環速度
控制 AI 的能力 ⇒ AI 的操控能力、跨服務整合 (json mode / tool use / mcp)、設計與規劃能力

91APP

OMO EMPOWER RETAIL BRAND

感謝各位聆聽