



變革

AI時代的軟體開發

Software Development in the AI Era: The Impact of LLMs on DevOps

About Me...

董大偉 David@isRock.com.tw

- 光岩資訊 資深技術顧問
- 微軟認證講師(Microsoft Certified Trainer)
- 微軟技術社群區域總監(Microsoft Regional Director)
- 微軟最有價值專家(Microsoft AI+DevOps MVP)
- LINE API Expert (LAE, LINE官方認證API專家)
- Professional Scrum Master (PSM I)
- 恆逸教育訓練中心特約講師、集英信誠資深顧問講師
- Regional Scrum Gathering Taipei 2021 大會講師
- Agile Summit 2023, 2022, 2019大會講師
- DevOpsDays Taipei 2024, 2023, 2022, 2021, 2019, 2017 大會講師
- 歷屆台灣微軟TechED, TechDays, DevDay 大會講師, China TechED大會講師
- ViewSonic、遠傳電信、信義房屋、中華電信學院、TSMC、世界先進、國防部、國衛院、渣打銀行、中國信託、家扶基金會...Agile/Azure DevOps教育訓練顧問講師
- Visual Studio上市發表會巡迴講師





變革緣起

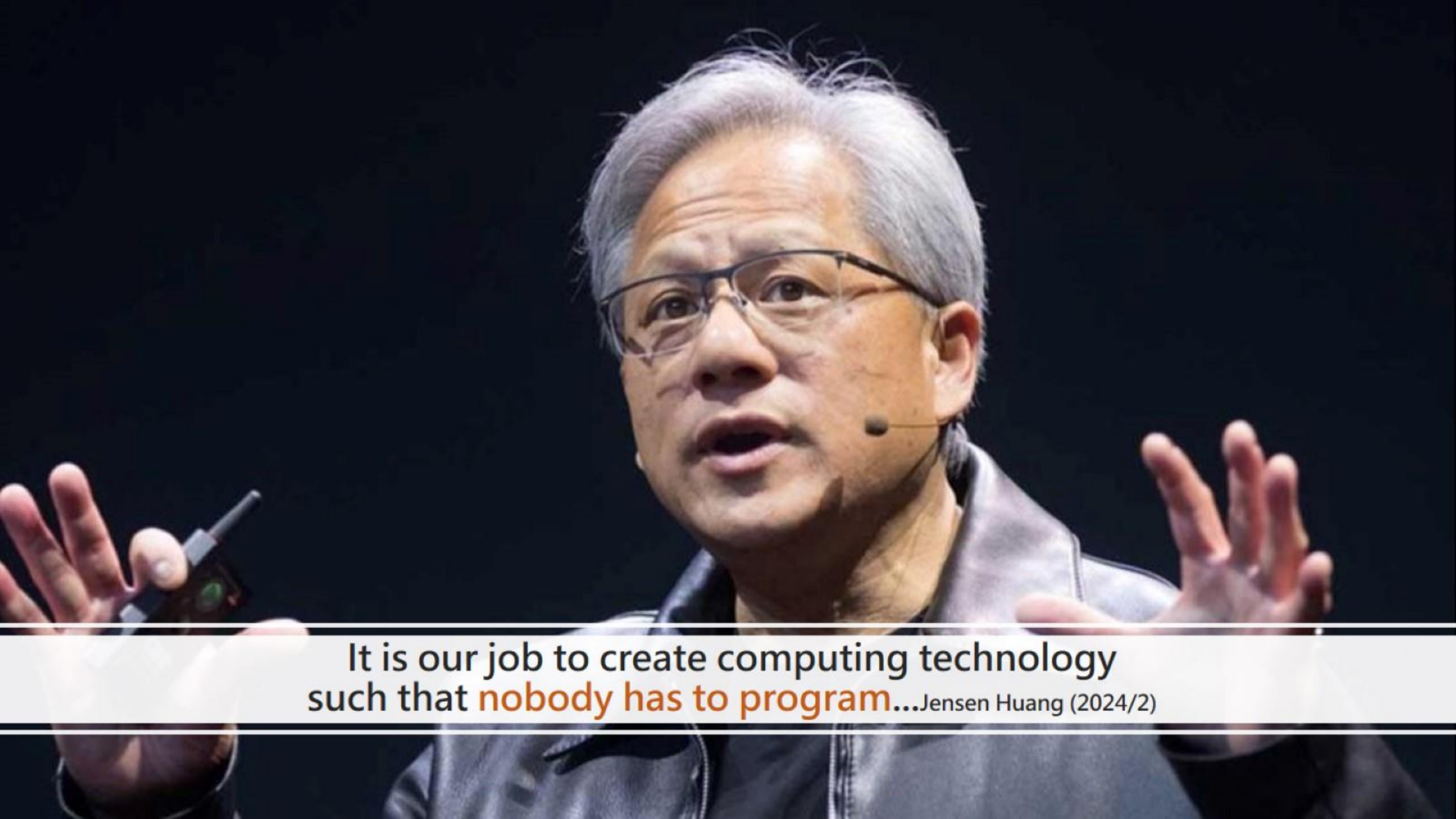
軟體工程到了 2025 年底 將會變得非常的不同...

一直以來，軟體開發充滿著各種挑戰：需求瞬息萬變、技術日新月異、協作錯綜複雜。面對高度變動的市場與需求，軟體交付需要迅速且頻繁，又不能犧牲品質和安全性，這一直個難題，直到...

01

PART ONE





It is our job to create computing technology
such that **nobody has to program**...Jensen Huang (2024/2)




Coding will go away.

We will just talk to AI
to describe what we
want and AI will create
it in code.

... Sam Altman (2024/3)





in 3-6 months AI will be
writing 90% of software code.
And that in 12 months, AI
may be writing all the code

... Dario Amodei (2025/3)





軟體開發的挑戰

需求瞬息萬變、技術日新月異、
協作錯綜複雜、時程壓力龐大...



LLM即將(已經?)帶來的改變

LLM出現之後，軟體開發乃至於整個軟體工程開始有了不一樣的質變



觀察到的現象

業界專家們看到了未來的可能性



← Back to Resources

AI Won't Solve Your Developer Productivity Problems for You

October 18, 2024 • 9 Minute Read



Uplevel Data Labs (2024)

Uplevel Data Labs，透過分析近 800 名開發人員樣本的數據後顯示，使用 AI 輔助開發並未如預期提升生產力，反而在某些情境下增加了錯誤率...

Share this Article:   

Measuring developer productivity is messy and hard primarily for two reasons:

- Goodhart's Law explains that once a metric becomes a target, it ceases to become a good measure. People operate by incentives, and if the incentive is to raise the metric, then the incentive will be to game the metric.
- "All models are wrong, but some are useful," explains statistician George Box. There is no one framework for productivity that can perfectly explain human behavior. There will always be variances, gaps, and complexity that a framework must ignore to provide the simplicity of a framework.

Table of Contents

The Results Are In: GenAI Isn't Improving Developer Productivity (Yet)

The Case-by-Case Search for Use Cases

Embracing a Scientific Mindset

Developer Productivity Remains an Unsolved Problem





RQ1 - Where

人工智慧在軟體開發生命週期(SDLC)中，何處(Where)帶來了改變？哪些地方該用 LLM？哪些地方不該用？



RQ2 - How

如何做(How)才能有效的將 LLM 帶入軟體開發流程中，實際幫上開發人員的忙？



RQ3 - Challenge

LLM 所帶來的變革，對於開發人員和開發團隊，將帶來哪些挑戰？如何因應？



變革の探索

LLM 在何處(Where)帶來了改變

一直以來，軟體開發充滿著各種挑戰：需求瞬息萬變、技術日新月異、協作錯綜複雜。面對高度變動的市場與需求，軟體交付需要迅速且頻繁，又不能犧牲品質和安全性，這一直個難題，直到...

02

PART TWO





LLM帶來的變革



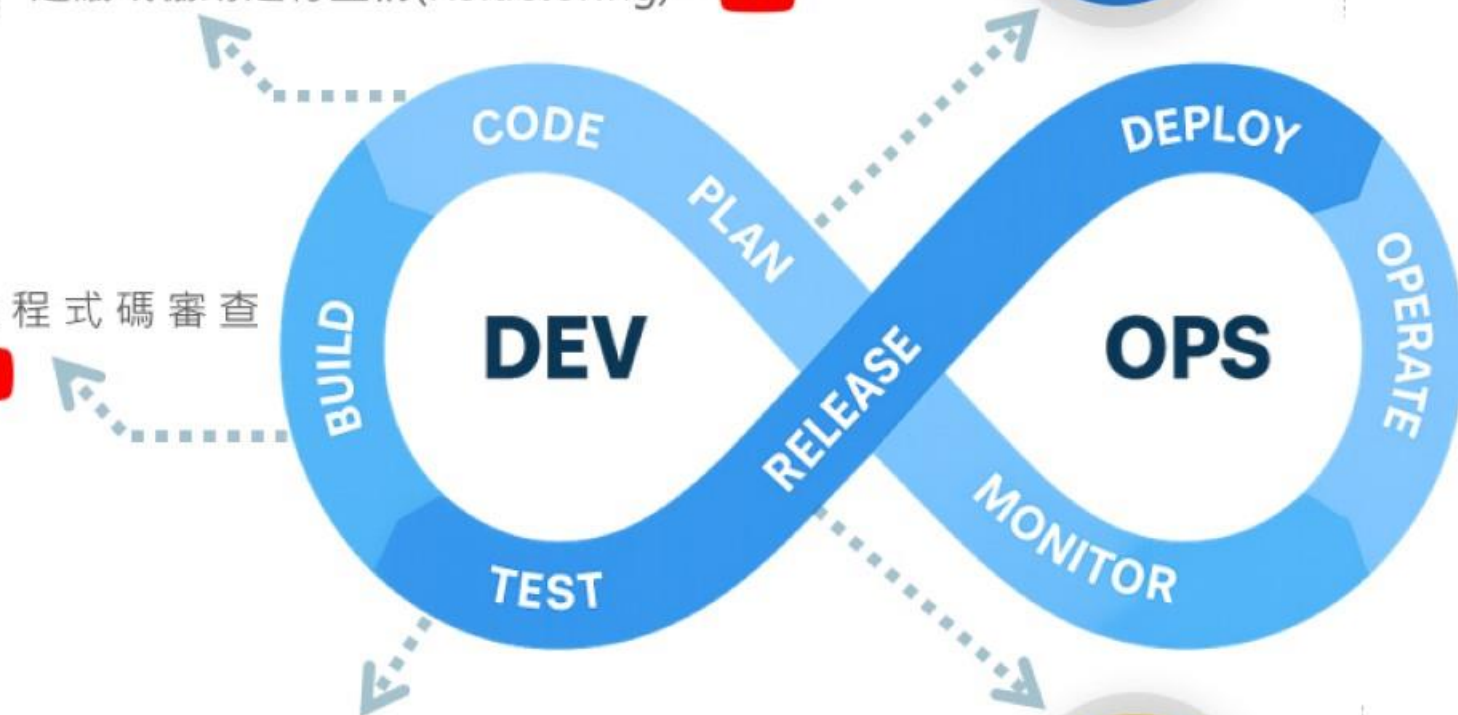
依照需求生成程式碼、自動撰寫單元測試 (Unit Test)、自動完成提交訊息 (Commit Message)、提供程式碼最佳化建議或協助進行重構 (Refactoring)。



在需求蒐集階段，自動產生使用者故事 (User Story) 與驗收準則 (Acceptance Criteria)。



協助進行自動化程式碼審查 (Code Review)



自動產生人工測試所依賴的測試案例 (Test Case)，提高測試的一致性和測試覆蓋率。

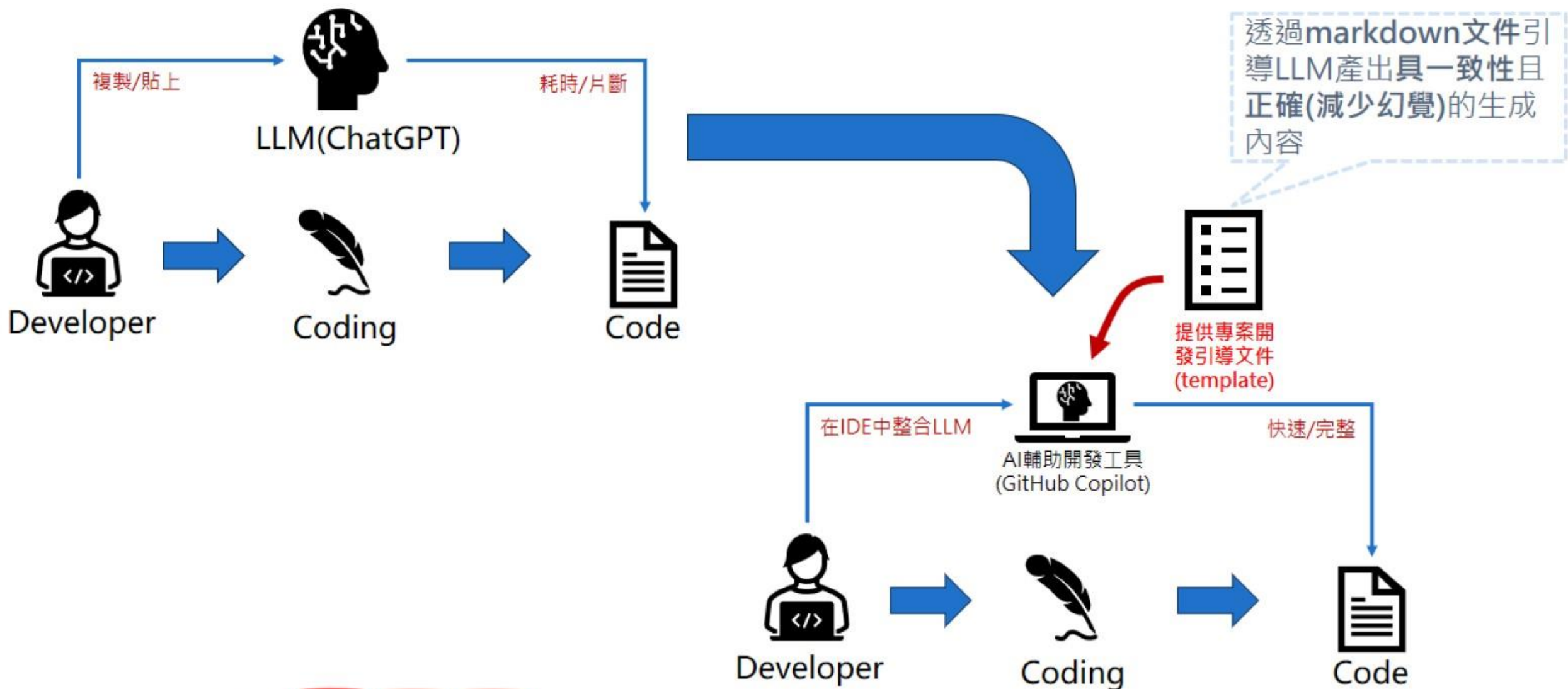


產生或更新系統維護文件，以及系統架構說明文件。





LLM帶來的變革



CODE





說明

👉 大範圍的程式碼生成仍有限制，使用**Code Completion**在正確性、一致性上，遠高於大範圍的**Code Generation**。

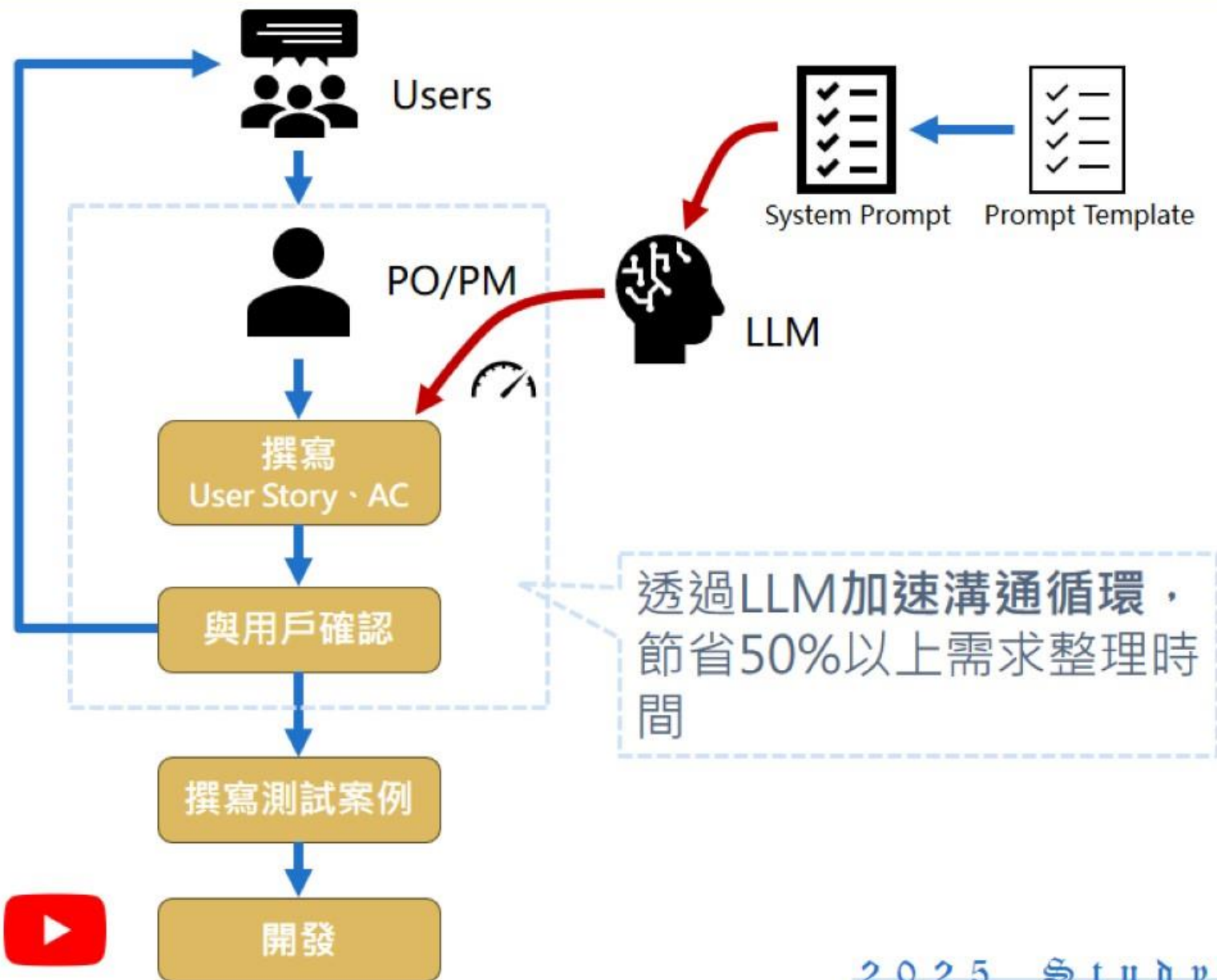
👉 可使用**引導文件**幫助LLM在每次生成時，維持一致性。

關鍵

- 1 使用IDE整合工具，90%程式碼由AI開發應該不是問題
- 2 最大的麻煩是源自於Context Window大小限制而產生的幻覺
- 3 造成程式碼生成不一致或錯誤
- 4 使用持續更新的**CodingGuide**文件引導LLM進程式碼生成

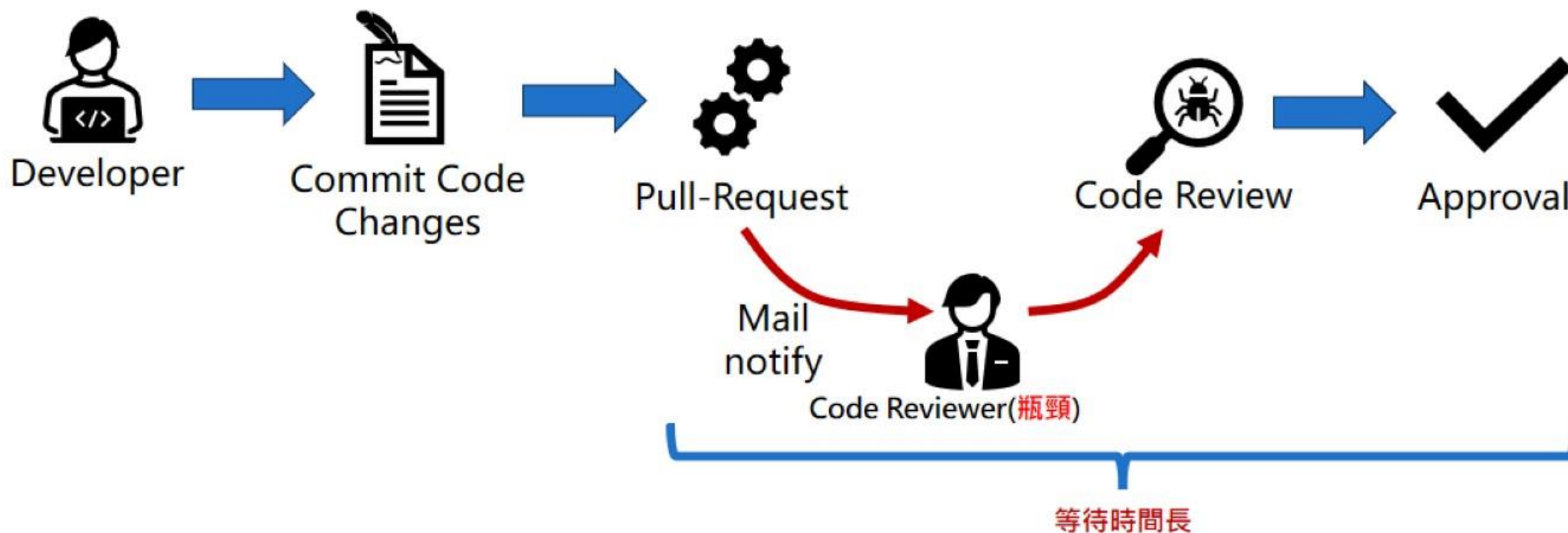


LLM帶來的變革



PLAN







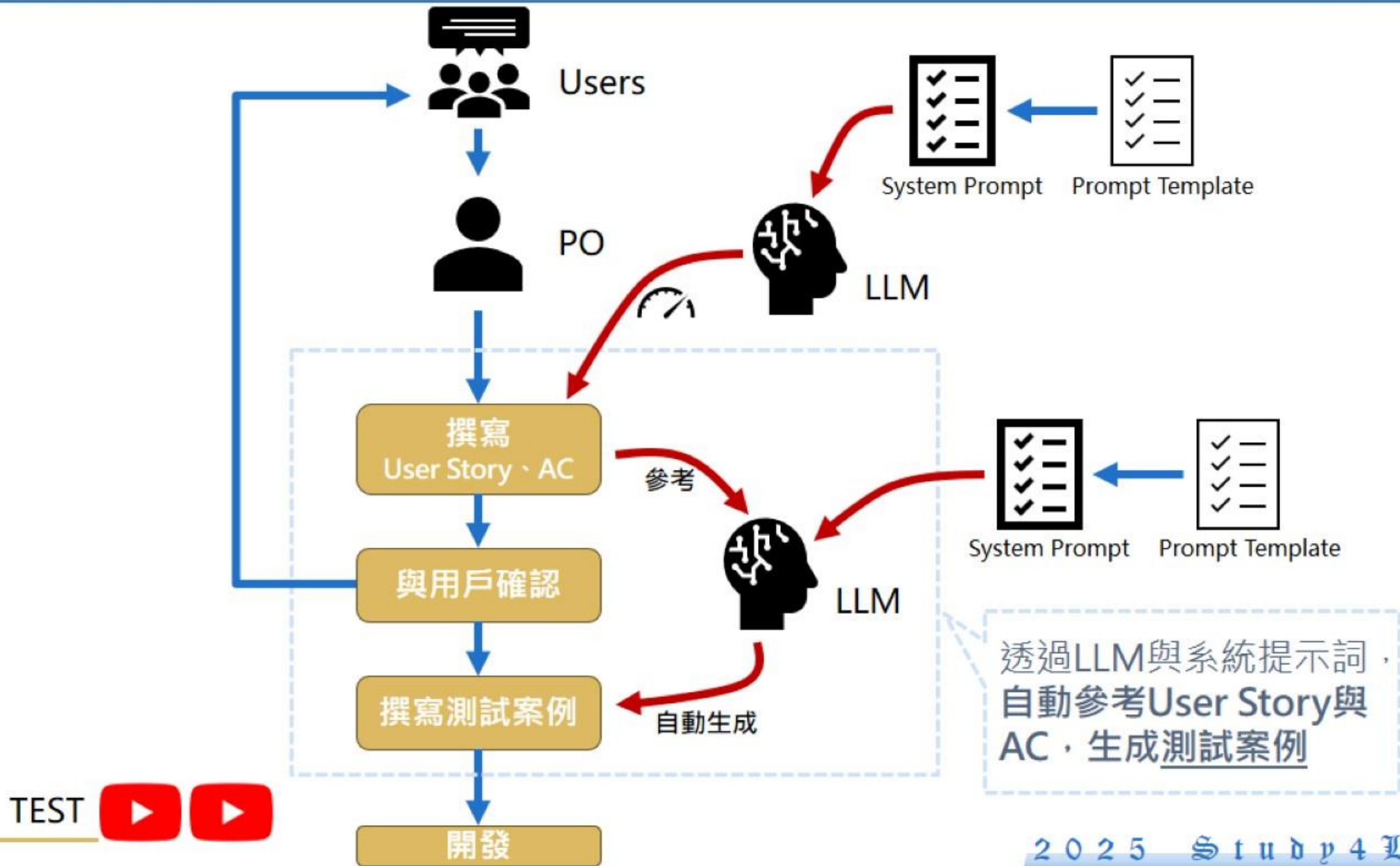
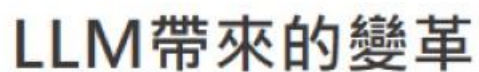
自動觸發

LLM Automatic
Code Review

掃描完成

在人工進程式碼審核前，由 AI 先行 Code Review，大幅節省時間(90%)、降低人力成本，並且提升品質





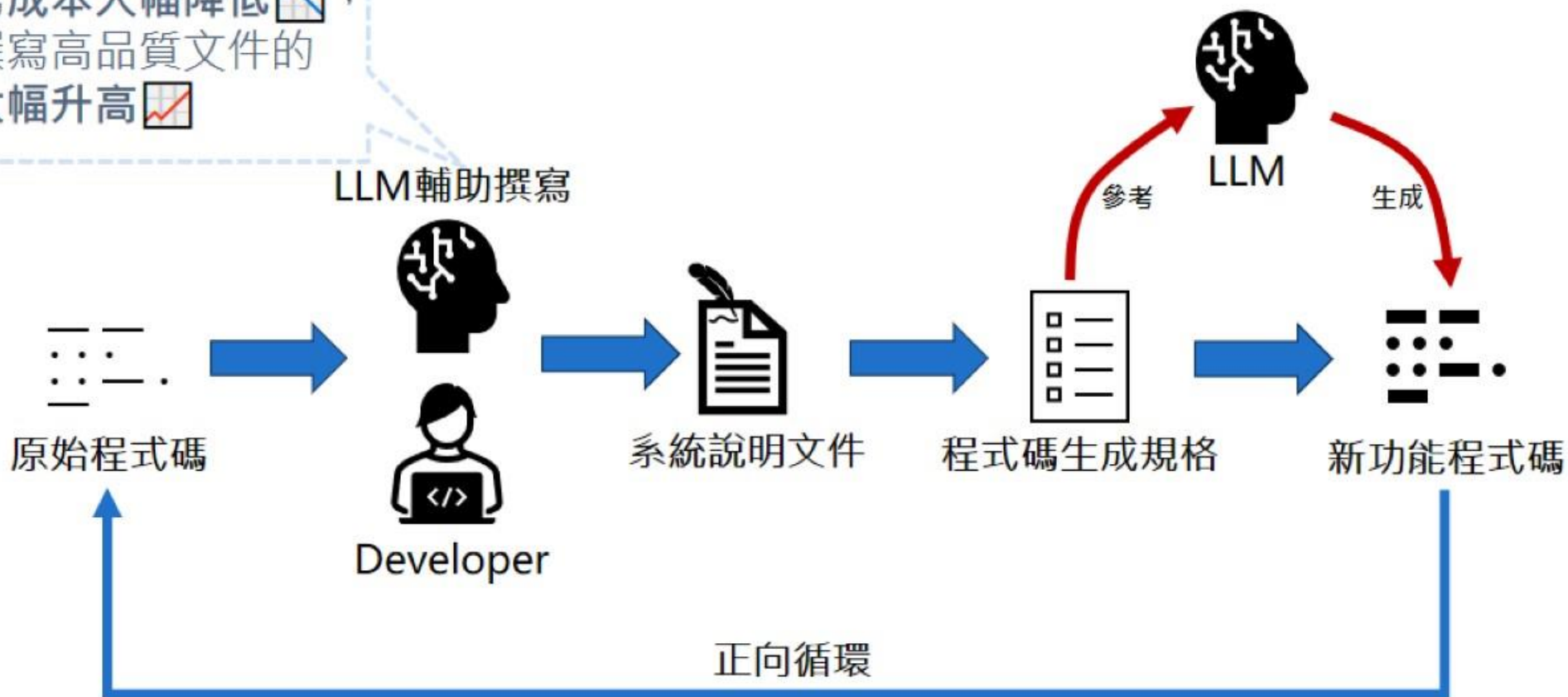


LLM帶來的變革



文件撰寫**成本大幅降低** ，
工程師撰寫高品質文件的
意願則**大幅升高** 

LLM輔助撰寫

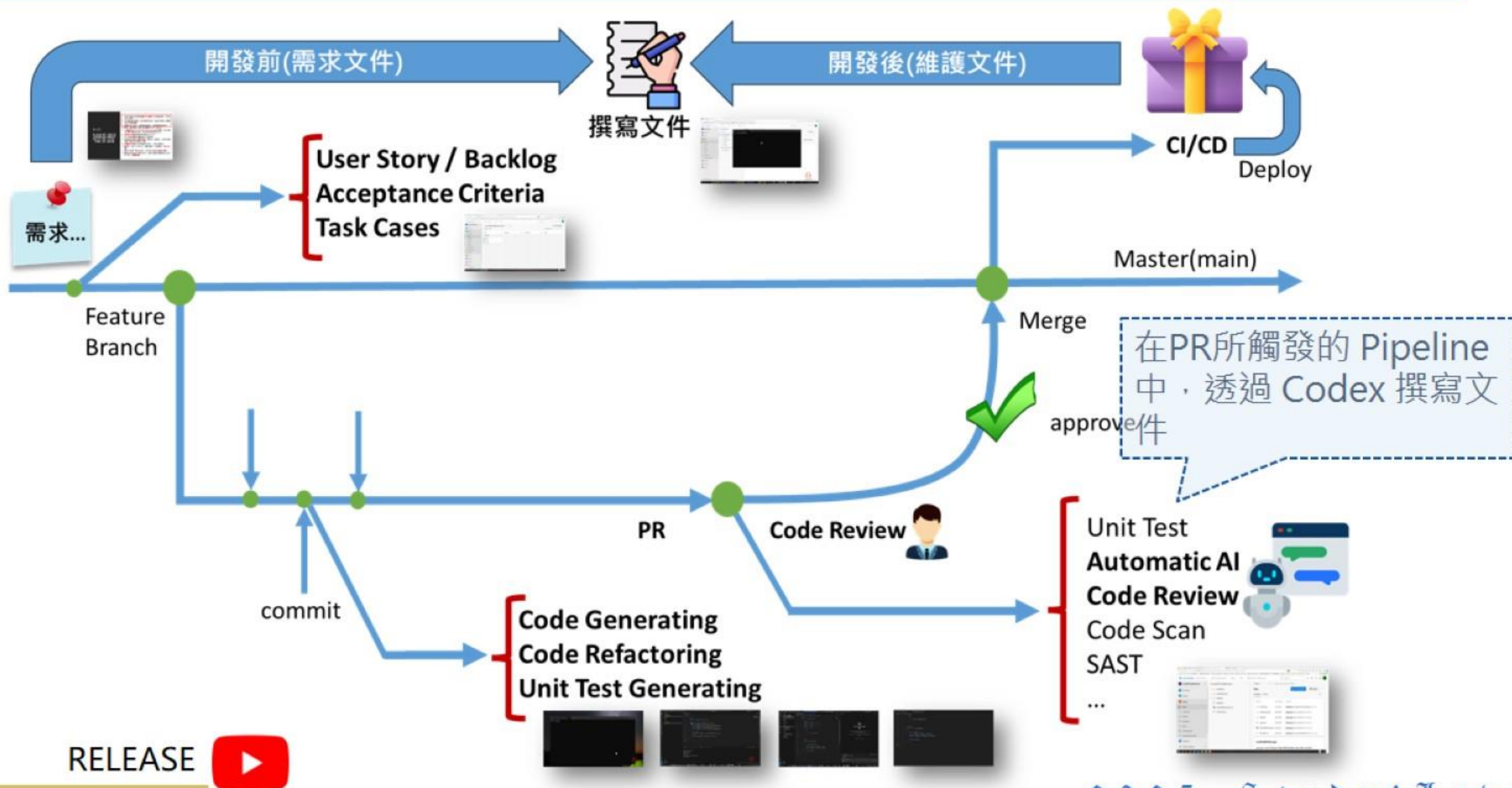


RELEASE





LLM帶來的變革





說明

使用 OpenAI Codex CLI / Gemini CLI，可以在命令列模式中，呼叫 LLM 模型的能力，進行任何外部工作，更新檔案、修改程式碼 Bugs、撰寫文件...etc.

如果我們可以將其併入 CI Pipeline 當中的話...

關鍵

- 1 OpenAI Codex CLI
- 2 可運行在具有 git 版控的環境
- 3 可自主在CLI中執行各種命令
- 4 若將其整合到Pipeline當中，是否能在夜間完成文件？



變革の挑戦

LLM 帶來了哪些挑戰(Challenge)

一直以來，軟體開發充滿著各種挑戰：需求瞬息萬變、技術日新月異、協作錯綜複雜。面對高度變動的市場與需求，軟體交付需要迅速且頻繁，又不能犧牲品質和安全性，這一直個難題，直到...

03

PART THREE



開發人員角色的轉變

未來**開發人員**所花費最多時間的工作，已不再是親手撰寫每一行**程式碼**，而是將重心轉移到如何讓AI能正確理解並生成符合需求的程式碼。



LLM 的除錯能力

大語言模型擅長生成語法正確的自然語言與程式碼，但**缺乏邏輯驗證能力**，若應用於需要高正確性的除錯任務，容易暴露其限制。



培養先行者策略

透過**內部社團**小範圍試行與分享，有助知識傳承、降低抗拒，並以實務經驗協助同仁快速上手，促進LLM在團隊中的推動。



文件的重要性提升

AI 輔助開發時代下，「**文件價值**與**文件撰寫成本**間的比例關係」正發生根本性的變化。



面對變革の挑戰

CONCLUSION

一直以來，軟體開發充滿著各種挑戰：需求瞬息萬變、技術日新月異、協作錯綜複雜。面對高度變動的市場與需求，軟體交付需要迅速且頻繁，又不能犧牲品質和安全性，這一直個難題，直到...

04

PART FOUR





結論



培養與他人、與機器間的 溝通能力

這一波浪潮中，被裁掉的是誰？
過去的中階主管，擅長的是拆解任務、帶領
技術專業人員，但未來需要的還是這樣嗎？



十倍速時代，必須以妥善 的流程整合人、機協作

開發速度變快了，但 bugs 也變多了怎麼辦？
倚賴人？依賴AI？
建構妥善的人機協作流程
(By AI+Human Process)



不再只是使用工具，而是 與工具對話

UI/UX曾經是重點，但如果透過對話就可以解決問題，
那又何必進行複雜的操作？



用在哪裡、怎麼用比你想像的 來得重要

讓機器(AI)做它擅長的事，然後讓人做他擅
長的事，重點是人機合作



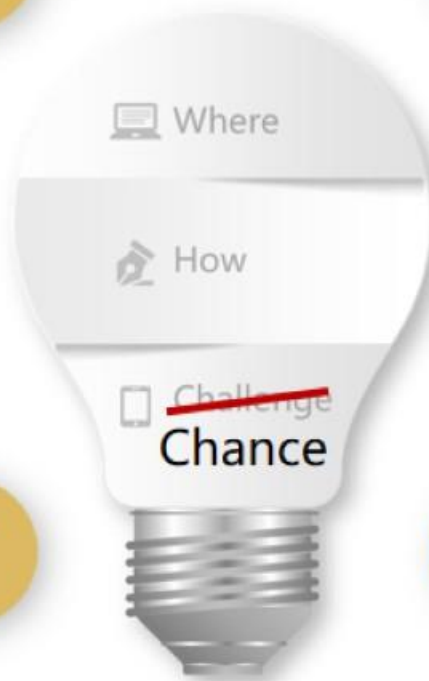
當格式不再是問題，內容 就開始成為關鍵

每個人都能寫程式、每個人都能寫文章、每個
人都能畫兩張圖、但有價值的内容依舊稀缺...



軟體工程變了

寫程式這件事，正慢慢從「寫程式的人」手
中被交接出去。目的沒變，但軟體工程變了，
「價值的交付」沒變，但實現的手法變了，





面對危機



關鍵

- 1 專注在你現有的優勢上
- 2 盤點你手上的options
- 3 設法突顯優勢，避開劣勢
- 4 想辦法讓自己活下去比較重要



十大生成式AI使用案例

2025年十大生成式AI用例點出了從技術性應用轉向情感性應用的轉變，尤其是在心理治療、個人生產力與個人發展等領域的成長。

主題

● 個人與專業支援

● 內容創造與編輯

● 學習與教育

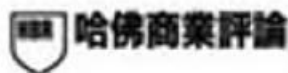
● 技術協助與故障排除

● 創意與娛樂

● 研究、分析與決策

* 未能進入2025年的百大名單

資料來源：Filtered.com



<https://www.hbrtaiwan.com/article/24037/how-people-are-really-using-gen-ai-in-2025>



那我們就先到這裡...

LINE官方帳號...

@studyhost

<https://wwjd.tw/409k687>



輸入『本日活動簡報』



THANKS

A I 時 代 的 軟 體 開 發

Software Development in the AI Era: The Impact of LLMs on DevOps

光岩資訊

董大偉(David)

日期: 2025/6/28