

DORA

2024

Google Cloud

加速發展： 開發運作現狀

Gold 贊助者



10

DORA 研究十年沿革

目錄

內容提要	3	結論	83
軟體推送成效	9	特別銘謝	85
人工智能：採用情形與觀感	17	作者	87
探討 AI 的下游影響	27	受訪者和公司統計結構	91
平台工程	47	研究方法	99
開發人員體驗	57	模型	113
引領變革	69	推薦讀物	117
DORA 研究十年沿革	77		

內容提要

十多年來，DORA 致力於調查表現出色、技術導向的團隊與組織具備哪些能力，又採取哪些做法與措施。在第十版的 DORA 報告，我們收集了全球超過 39,000 位專業人士的意見，他們所在的組織規模各異，所屬產業也不同。他們對這項研究貢獻良多，萬分感謝！

DORA 每年會對技術與相關職務的專業人士進行全球問卷調查，藉此收集資料。我們希望透過問卷調查，瞭解整個組織及其員工的工作方式與成果。

DORA 以嚴謹的統計評估方法解讀這些因素之間的關係，以及各因素對團隊與組織成就的貢獻程度。

我們今年擴充了問卷調查，加入專業人士深度訪談，希望取得更精闢的洞察資料、結合多種觀點，並為研究結果提供更多背景脈絡。詳請請參閱「[研究方法](#)」一章。

今年調查的重要成就和結果如下：

減少工作倦怠

「工作倦怠」是指長期承受壓力或壓力過大，形成情緒、身體和心理不堪負荷的狀態，通常會有憤世嫉俗、疏離淡漠和沒有成就感等感受。

心流狀態

「心流狀態」衡量員工在執行開發工作時，通常能多專心。

工作滿意度

「工作滿意度」會衡量個別員工對工作的整體感受。

組織表現

這項指標會衡量組織在下列面向的表現：獲利能力、市占率、顧客總數、營運效率、顧客滿意度、產品與服務的品質，以及達成目標的能力。

產品效能

這項指標會衡量產品的可用性、功能、價值、供應情形、效能（例如延遲時間）和安全性。

工作效率

「工作效率」衡量員工認為自己創造價值、完成工作的效率與成果。

團隊表現

這項指標衡量團隊的互助合作與創新能力、工作效率及適應力。

重要研究結果

AI 帶來廣泛影響

AI 為軟體開發領域帶來典範轉移，率先採用 AI 的組織已取得亮眼成果，不過仍須謹慎小心。

採用 AI 可以在下列面向產生效益：

- 心流狀態
- 工作效率
- 工作滿意度
- 程式碼品質
- 內部說明文件編寫
- 審查程序
- 團隊表現
- 組織表現

不過，採用 AI 也有可能產生負面影響。我們觀察到軟體推送成效不增反減，AI 對產品效能的影響仍不明確。另外，某些受訪者表示，隨著 AI 採用率提高，他們投注於高價值工作的時間反而降低。後續章節會進一步探討這項有趣的研究結果。

團隊應持續實驗探究，深入瞭解提高 AI 採用率的影響。

AI 日漸受到信任，採用率也隨之提高

採用生成式人工智慧(簡稱「生成式 AI」)之後，開發人員認為工作變得更有效率，信任生成式 AI 的開發人員也使用得更頻繁。不過，這個部分仍有進步空間，39.2% 的受訪者表示不太信任或完全不信任 AI。

重視使用者才能增進表現

首重使用者體驗的組織可以打造出更優質的產品，開發人員的工作效率與滿意度更高，出現倦怠的機率較低。

轉型領導力至關重要

轉型領導力可以提高員工的工作效率、工作滿意度、團隊表現、產品效能和組織表現，工作倦怠的情況也會減少。

確定優先要務，才能增進工作效率和身心健康

即便組織有出色的領導階層、優質的內部說明文件，還有以使用者為中心的軟體開發做法，如果優先要務經常更動，工作效率就會顯著降低，倦怠情形也會明顯增加。

平台工程有望大幅提高工作效率

平台工程會對工作效率和組織表現帶來正面影響，但在軟體推送成效方面則須多加留意。

採用雲端可以提高基礎架構彈性

有彈性的基礎架構可以增進組織表現，但若未充分運用雲端的彈性，遷移至雲端可能會比繼續使用資料中心更不利。在做法、程序和技術方面都必須轉型，才能順利遷移。

優異的軟體推送成效並非遙不可及

成效最高的團隊在四項軟體推送指標(變更前置時間、部署頻率、變更失敗率和失敗部署作業復原時間)全數勝出，成效最低的團隊則在四項指標都表現不佳。按表現劃分的各族群中，都有來自各行各業的團隊。

應用 DORA 洞察資料

如要運用 DORA 增進團隊和組織表現，您需要評估目前的情況、找出要投注資源並加以改善的部分，還要建立回饋循環來掌握進展。秉持精益求精的思維與做法，就有機會獲得最大效益。此外，也要投入資源來鞏固必要的組織能力，持續提升表現。

您可以參考 DORA 研究結果，用來規劃實驗及擬定假設，但請務必測試並衡量變更帶來的影響，瞭解怎麼做最適合您的團隊和組織，同時也可以驗證我們的研究結果。您的實驗結果不一定與我們相同，但歡迎分享進展，讓所有人一起從您的經驗中學習。

透過實驗持續改善

1. 確定想增進的領域或成果
2. 衡量基準或當前狀態
3. 根據想達成的目標擬定假設，提出可能有效做法
4. 取得共識並開始實施提升計畫
5. 執行計畫
6. 衡量進展
7. 重複相同的程序。反覆試驗、逐步改善，才能取得成果

DORA COMMUNITY



提高成效無法僅憑一己之力！

我們可以從彼此的經驗中學習。歡迎瀏覽 DORA 社群網站 (<https://dora.community>)，在這個優質論壇分享、查看各式提升計畫。

軟體推送成效

技術導向團隊需要制定方法來衡量成效，以便評估目前的情況、排定改善項目的優先順序，以及確認進展。要想有效衡量軟體推送程序的成果，需特別留意 DORA 已多次檢核的四項關鍵指標。



四項關鍵指標

衡量軟體變更的處理量和穩定度時，業界採用 DORA 的四項關鍵指標。所謂的「變更」包含各種變動，例如設定和程式碼的異動。



變更前置時間：

將程式碼修訂版本或異動成功部署至正式環境所需的時間。



部署頻率：

將應用程式變更部署至正式環境的頻率。



變更失敗率：

部署項目在正式環境產生問題¹，而需要熱修補 (hotfix) 或復原的百分比。



失敗部署作業復原時間：

部署失敗時所需的復原時間。

我們發現這些指標通常會有連動性，成效最高的團隊在四項指標的表現都很優異，成效最低的團隊則在四項指標都表現不佳。

持續改良軟體推送成效衡量方法

分析四大關鍵指標時，我們發現「變更失敗率」長久以來都是離群值²，雖然與另外三項指標高度相關，但從統計檢定和統計方法的角度來看，這四項指標並不能合併為單一因素。關於變更失敗率的問題經過重新設計後，關聯度有所上升，但我們認為可能還有其他影響因素。

長久以來，我們都假設變更失敗率指標可以反映團隊必須重新處理的工作量。推送失敗時，團隊必須修正變更，這麼做就有可能產生另一項變更。

為測試這個理論，我們今年新增了一個有關應用程式重新處理率的題目：「針對您處理的主要應用程式或服務，過去六個月中，為解決會影響使用者的應用程式錯誤，您大約執行過幾次未預期的部署作業？」

資料分析結果驗證了我們的假設，重新處理率確實與變更失敗率有關。只要合併使用這兩項指標，就能衡量軟體推送穩定度，而且結果十分可靠。

分析軟體推送效能等級時，我們也觀察到相同的現象。今年的研究中，軟體推送作業處理量與穩定度出現差異的團隊數量超過一半，這讓我們改以兩項不同的因素來評估軟體推送成效：

概念	
軟體推送成效	
因素	
軟體推送作業處理量	軟體推送作業穩定度
採用的指標	
<ul style="list-style-type: none">• 變更前置時間• 部署頻率• 失敗部署作業復原時間	<ul style="list-style-type: none">• 變更失敗率• 重新處理率

以這份報告中的分析資料來說，我們多次使用軟體推送成效的概念和兩項因素，描述軟體推送成效時則採用全部五項指標。

描述軟體推送作業處理量時，我們採用變更前置時間、部署頻率和失敗部署作業復原時間指標。所謂軟體推送作業處理量，衡量的是執行更新的速度，無論是一般性或因應失敗所做的變更，所有種類的更新都包含在內。

描述軟體推送作業穩定度時，我們採用變更失敗率和重新處理率指標。軟體推送作業穩定度會衡量部署作業意外引發額外工作，必須立即處理的機率。



成效等級

針對受訪者處理的主要應用程式或服務，我們每年都會調查軟體推送成效，並對受訪者的答案進行集群分析。這種統計方法可將相似的答案集中在同一群組中，使得群內差異小、群間差異大。

為了與往年的集群分析做法保持一致，我們只針對原本的四項軟體推送指標進行集群分析。

分析軟體推送成效之後，受訪者的回答形成四個集群。我們並未事先設定這些等級，而是讓問卷調查回答自然形成不同等級。這樣一來，我們就能透過數據匯報，查看各年所有受訪者的軟體推送成效。

今年的資料形成四個截然不同的集群，如下所示。

成效等級	變更前置時間	部署頻率	變更失敗率	失敗部署作業復原時間	受訪者百分比*
優異	不到一天	視需求 (每日部署多次)	5%	不到一小時	19% (18 至 20%)
高	介於一天至一週之間	介於每天一次至每週一次之間	20%	不到一天	22% (21 至 23%)
中	介於一週至一個月之間	介於每週一次至每個月一次之間	10%	不到一天	35% (33 至 36%)
低	介於一至六個月之間	介於每個月一次至每六個月一次之間	40%	介於一週至一個月之間	25% (23 至 26%)

*不確定性區間為 89%

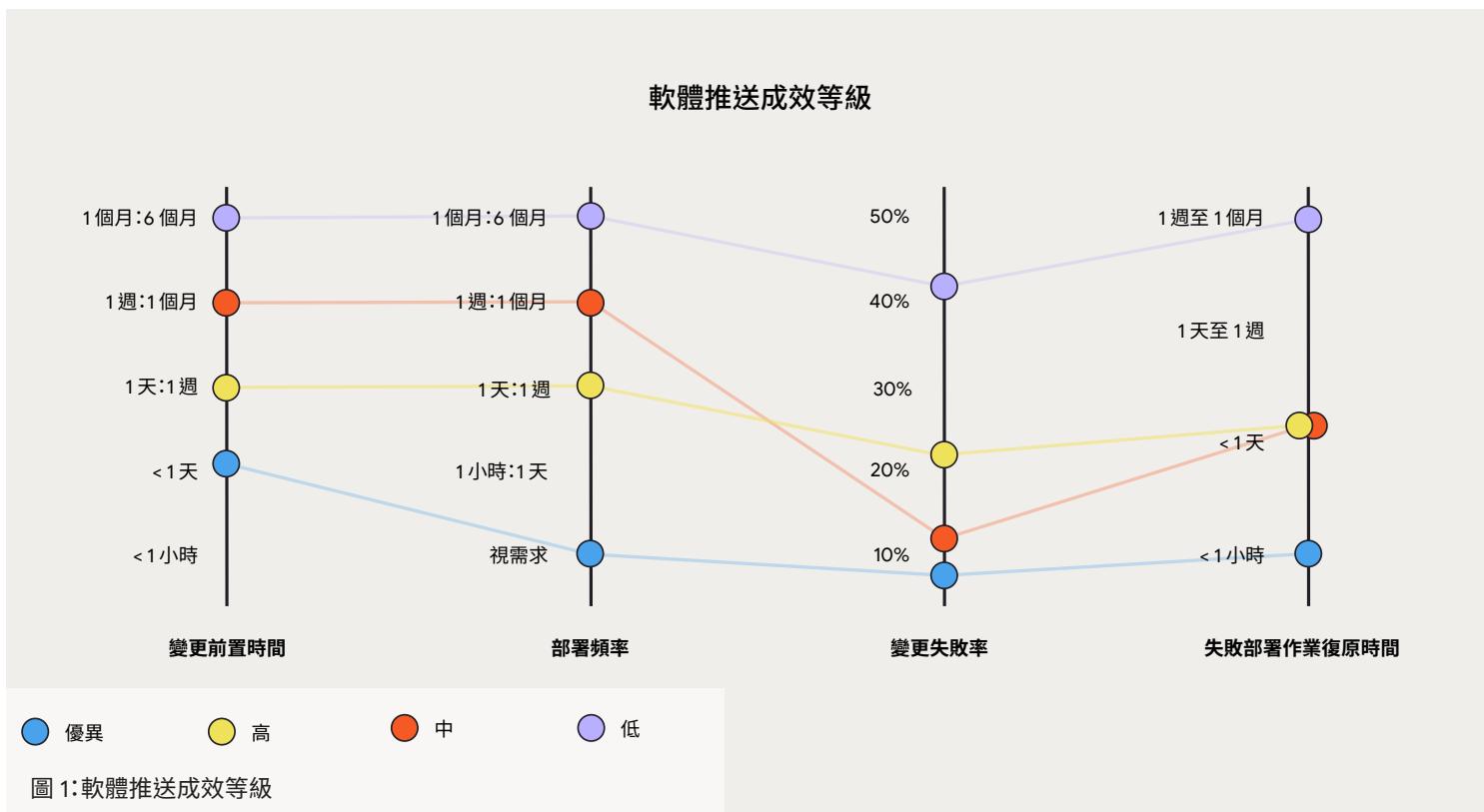
處理量和穩定度哪個重要？

全部四個集群中，處理量和穩定度都有關聯性，即便在中成效集群（橘色）也是如此。不過相較於高成效集群（黃色），前者的處理量較低，而穩定度較高。這代表集群成效會受到處理量和穩定度之外的因素影響。舉例來說，如果中成效集群的變更頻率提高，軟體推送成效就有可能增加。

那麼「較高的部署頻率」和「較低的失敗率」，哪個比較好？

這個問題或許沒有放之四海而皆準的答案，需要考量多項因素，包括要處理的應用程式或服務、應用程式團隊想達成的目標，其中最重要的是應用程式使用者的期望。

我們以往將作業速度較快的團隊稱為「高成效團隊」，作業速度較慢而較穩定的團隊則稱為「中成效團隊」。不過，這麼做突顯了採用這些成效等級的其中一項潛在缺陷：比起達到特定成效等級，團隊更應注重進步。最出色的團隊應該是「進步」良多者，不一定是「成效」優異者。



相較於成效低的團隊，成效優異的團隊具備下列優勢：

127 倍

前置時間加快
的程度

182 倍

每年多部署的
次數

8 倍

變更失敗率降低
的幅度

2293 倍

失敗部署作業復
原時間縮短的
幅度

如何運用成效集群

您可以透過成效集群，瞭解今年問卷調查受訪者在軟體推送成效方面的基準表現。任何組織都有機會實現優異成效，我們希望藉由這些集群鼓勵您朝這個目標邁進。

比起達到特定成效等級，我們認為團隊更應該注重提高整體成效。最出色的團隊應該是「進步」良多者，不一定是「成效」優異者。

成效等級與產業別的關聯性不顯著

我們的研究幾乎不曾³顯示產業別會影響軟體推送成效，各行各業都有高成效的團隊。這不代表各產業沒有特有的難題，只是在軟體推送成效方面，沒有特定產業出現成效低落或突出的情況。

運用軟體推送成效指標

各項應用程式或服務的背景脈絡都不同。由於情況複雜，開發人員很難預測每一次變更對系統整體成效的影響。另外，組織幾乎不可能一次只實作一項變更。考量到這樣的複雜性，您該如何根據軟體推送成效指標，進行改良工作呢？

首先，請找出要衡量及改良的主要應用程式或服務，接著建議您召集負責該應用程式或服務的跨職能團隊，衡量目前的軟體推送成效並取得共識。您可以參考 DORA 快速檢驗 (<https://dora.dev/quickcheck>) 來溝通及制定這個衡量基準。團隊必須瞭解成效受阻的原因。

如要有效找出成效受阻的原因，可以與團隊一起製作價值流程圖⁴。

接著，請擬定改良計畫並達成共識。計畫可側重 DORA 研究列出的任一能力⁵加強，也可以依據應用程式或組織特有的需求調整。

計畫確立之後，即可開始實行！請分配專屬資源與人力來執行這項改良工作，並留意過程中學到的經驗。

實作變更一段時間後，請重新評估四項關鍵指標，瞭解成效是否出現變化，並記錄您學到的經驗。

團隊可以重複這項程序，培養精益求精的行動力。

別忘了，改變並非一蹴可幾。唯有不斷改善，才能營造學習風氣，並建立快速的流程和意見回饋機制⁶。

-
1. 如果部署項目推送至正式環境後產生問題，並有可能影響使用者，我們才會判定為變更失敗。反過來說，如果變更尚未進入正式環境就中止，則代表該組織在部署程序中有能力偵測出錯誤。
 2. Forsgren、Nicole、Jez Humble 與 Gene Kim, 2018 年。《Accelerate: The Science Behind DevOps: Building and Scaling High Performing Technology Organizations》(ACCELERATE：精益軟體與 DevOps 背後的科學)。IT Revolution Press, 第 37 至 38 頁
 3. 我們曾發現零售業的軟體推送成效大幅高於其他產業，詳情請參閱《加速發展：2019 年開發運作現狀》報告第 32 頁，連結如下：<https://dora.dev/research/2019/dora-report/2019-dora-accelerate-state-of-devops-report.pdf#page=32>
 4. <https://dora.dev/guides/value-stream-management/>
 5. <https://dora.dev/capabilities>
 6. <https://dora.dev/research>

人工智慧： 採用情形 與觀感



重點摘要

在我們研究的所有產業中，多數組織都開始優先將 AI 深度整合至自家應用程式和服務。由於這樣的趨勢，大部分的專業開發人員也開始運用 AI 處理核心工作，並表示效率有所提升。專業開發人員普遍認為，要在現今的市場保有競爭力，使用 AI 是勢在必行，這似乎也說明了組織和個人在進行專業開發時，為何紛紛採用 AI 技術。

簡介

以今年來說，AI 對開發工作生態的驚人影響力絕對不容忽視。從各大新聞媒體爭相報導即可見一斑，無論他們認為 AI 的效益是好¹、壞²還是讓人啼笑皆非³。在《加速發展：2023 年開發運作現狀》報告討論 AI 時，我們只將其視為會影響成效的其中一項技術功能⁴，今年則會更完整探討這個主題。

AI 在專業開發工作的使用量迅速提升，從小眾工具變得無人不用。在這個關鍵的產業轉折點，我們希望把握眼前的重大機會，透過《加速發展：2024 年開發運作現狀》報告，評估 AI 的採用情形、用途和專業開發人員對這項技術的觀感。

研究結果

採用人工智慧

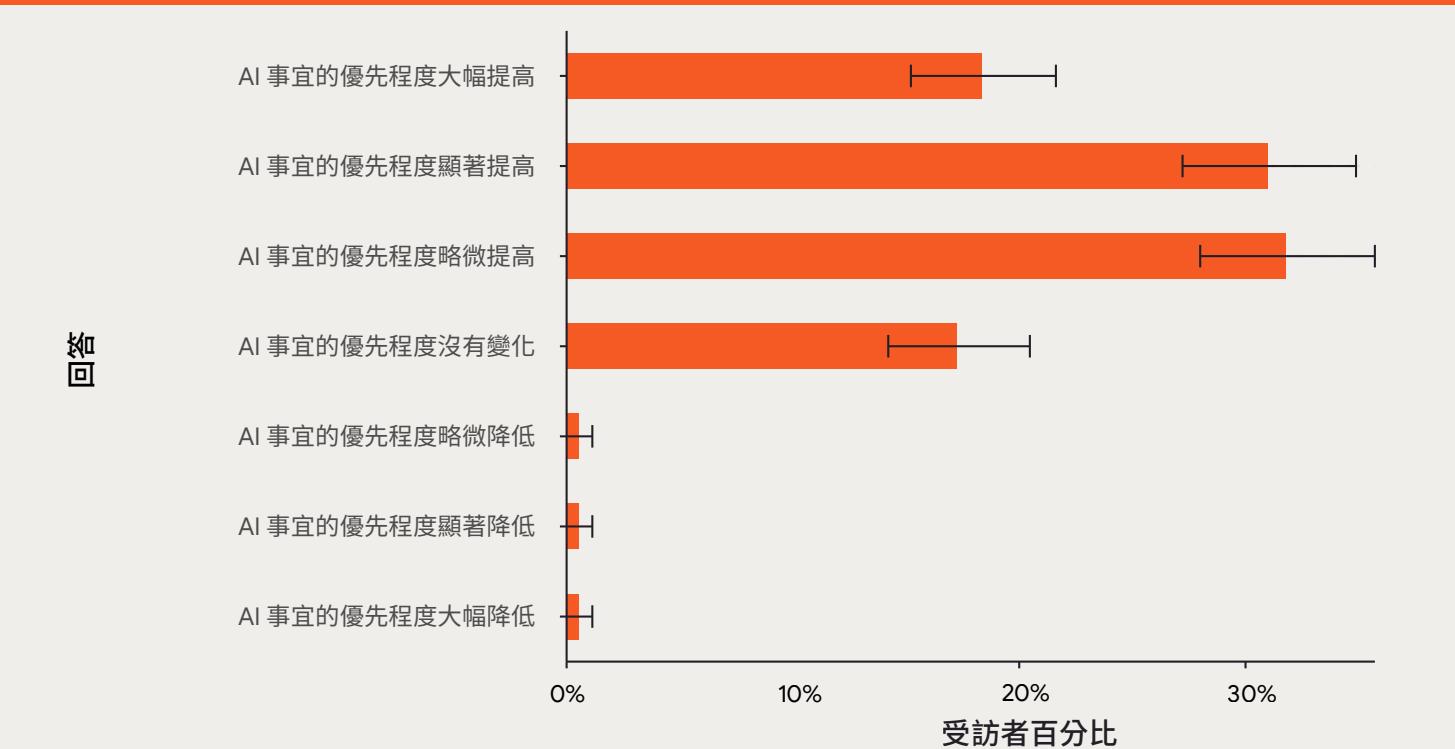
AI 採用情形的研究結果顯示，越來越多人認為 AI 不再只是新興技術，而且已經大行其道，未來各界也很有可能持續使用。

組織的人工智慧技術採用情形

絕大多數的受訪者 (81%) 表示，所屬組織已改變重點業務，開始優先將 AI 進一步整合至自家應用程式和服務。49.2% 的人甚至認為有「大幅」轉變，或力度「顯著」。

值得注意的是，3% 的受訪者表示所屬組織正在減少 AI 相關規劃（這個百分比仍在問卷調查的誤差範圍內），而有 78% 的人相信所屬組織在變更優先要務之後，會開誠布公地說明預計如何運用 AI。詳細資料請見圖 2。

AI 相關的組織優先要務變化



誤差線代表 89% 的不確定性區間

圖 2：針對將 AI 整合至自家應用程式和服務，受訪者認為所屬組織的優先要務變化傾向（提高或降低）。

無論產業別，受訪對象仰賴 AI 處理日常工作的程度，從統計學來看皆可視為相同等級，代表所有產業都正在快速採用 AI 技術。這個結果讓我們有點訝異，因為個別產業的監管限制程度可能大不相同，過往的創新速度也大相逕庭，這兩項因素都有可能影響技術採用速度。

不過我們確實發現，相較於在小型組織服務的受訪者，來自大型組織的受訪者較不仰賴 AI 處理日常工作。這與過往的文獻一致，即大型公司的組織架構較為複雜，協調成本較高，因此適應技術變遷的速度較慢⁵。

個別員工的人工智慧技術採用情形

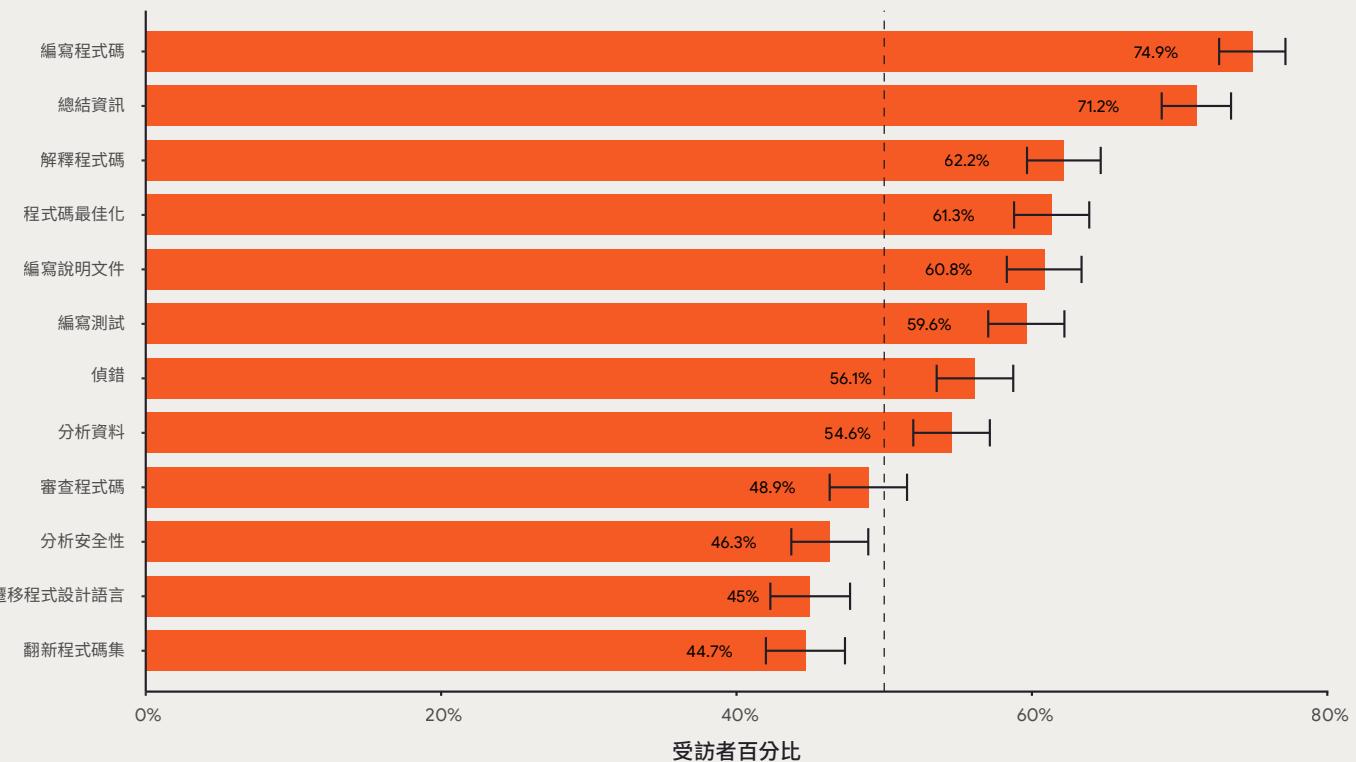
75.9% 的個人受訪者，至少會在一或多項日常工作中的某些部分使用 AI 技術。在職務涵蓋下列工作的所有受訪者當中，多數人採用 AI 技術來處理這些工作：

1. 編寫程式碼
2. 總結資訊
3. 解釋不熟悉的程式碼
4. 程式碼最佳化
5. 編寫程式碼說明文件
6. 編寫測試
7. 偵測程式碼錯誤
8. 分析資料

以軟體開發工作來說，問卷調查回答涵蓋的所有工作當中，最常用到 AI 技術（至少會在某些部分使用）的是編寫程式碼及總結資訊，在需要處理這兩項工作的受訪者當中分別占 74.9% 和 71.2%。詳細資料請見圖 3。



各工作的 AI 仰賴程度



誤差線代表 89% 的確信區間

圖 3：受訪者在執行十二項常見的開發工作時，使用 AI 技術（至少在某些部分使用）的百分比

日常工作中，受訪者最常與 AI 互動的介面為聊天機器人 (78.2%)，其次是外部網頁介面 (73.9%)，以及 IDE 內嵌的 AI 工具 (72.9%)。相較之下，受訪者較少在內部網頁介面 (58.1%) 和自動 CI/CD pipeline (50.2%) 使用 AI。

不過，我們確定受訪者與 CI/CD pipeline 和內部平台互動的頻率，可能會影響他們是否知道自己在這些技術使用 AI。因此，圖中的數字可能會因人為因素而偏低。

我們發現，相較於其他所有職務的受訪者，數據資料學家和機器學習專家較有可能仰賴 AI。相反地，硬體工程師仰賴 AI 的可能性則比其他職務都低，可能是因為他們的職務不包含上述這些經常使用 AI 的工作。



採用人工智慧的原因

訪談對象經常基於兩個原因而決定採用 AI，一是因應競爭壓力，二是為了跟上組織和開發人員適用的業界標準，這些標準越來越重視對 AI 的精通程度。

對於某些訪談對象所屬的組織來說，使用 AI 是「一大賣點」(訪談對象 3)⁶，可以與競爭對手做出區隔。光是競爭對手開始在工作流程採用 AI 的消息，就足以讓公司捨棄典型的「龐大官僚體制」思維，因為害怕落於人後，而願意加快腳步採用新技術，認為採用 AI 是當務之急 (訪談對象 11)。

許多個人受訪者會採用 AI，是因為覺得精通 AI「似乎已變成軟體開發工程師的新門檻」(訪談對象 9)。一些訪談對象建議同行盡快在開發工作流程採用 AI，並表示「這個領域瞬息萬變，難以跟上腳步...。如果不使用，很快就會被拋下」(訪談對象 4)。

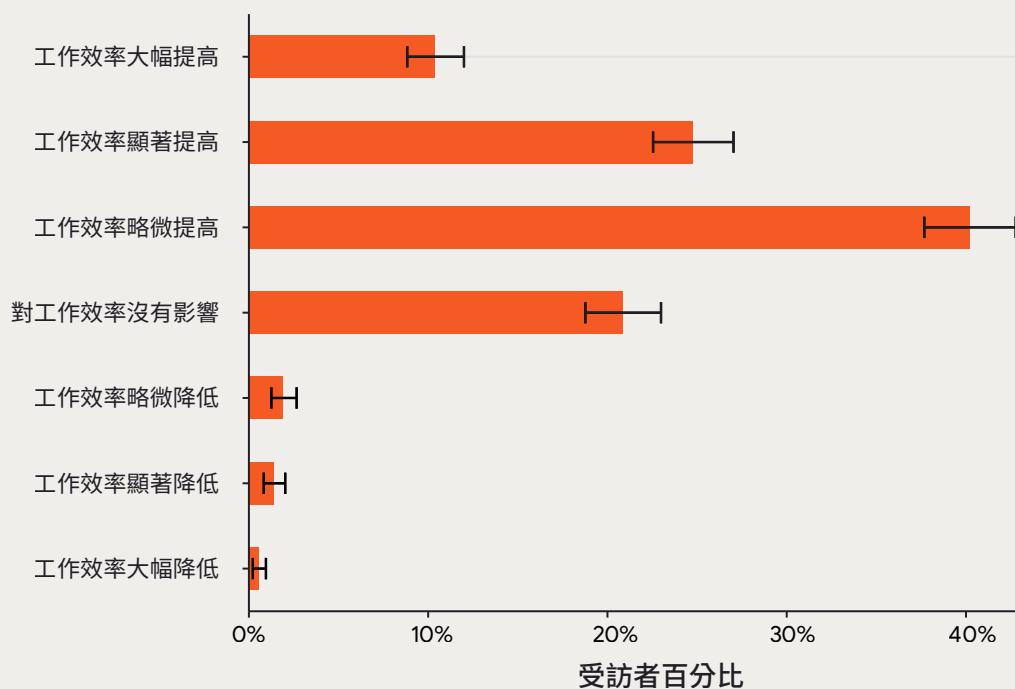
對人工智慧的看法

人工智慧讓成效提高

對於採用 AI 的眾多組織和開發人員，在開發工作採用這項技術的好處似乎相當吸引人。在 2024 年初執行問卷調查的三個月以前，75% 的受訪者認為，採用 AI 可以提高工作效率。

值得注意的是，25% 的受訪者表示觀察到顯著的工作效率提升，10% 則認為提升幅度極大，兩者總計超過三分之一；不到 10% 的人認為 AI 會降低工作效率（包含略微降低）。詳細資料請見圖 4。

對 AI 改變工作效率的看法



誤差線代表 89% 的不確定性區間

圖 4：受訪者認為 AI 對工作效率的影響。

綜觀受訪者的所有職務，工作效率提升幅度最大的是資安專業人員、系統管理員和全端開發人員。雖然行動服務開發人員、網站可靠性工程師和專案經理同樣表示 AI 可以提高工作效率，不過比起報告提及的其他所有職務，他們的工作效率提升幅度較低。

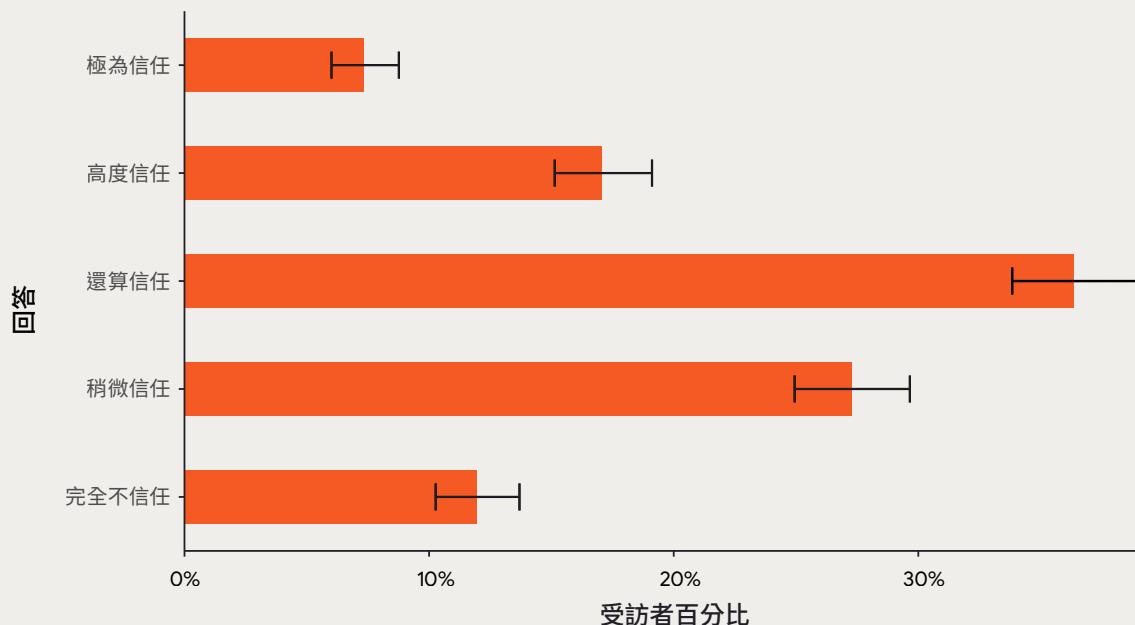
在開發工作使用 AI 是新興的做法，使用者需要一段時間才能上手，因此我們原先推測 AI 會妨礙開發人員編寫程式，但研究結果並不支持這項假說。只有 5% 的受訪者表示，AI 對自己編寫程式的能力造成某種程度的阻礙。事實上，67% 的人認為使用 AI 程式設計工具之後，編寫程式碼的能力有所提升。約 10% 的受訪者表示 AI「大幅」提升自己的程式碼編寫能力。

AI 生成的程式碼可信嗎？

AI 生成的程式碼是否值得信賴，可以在開發工作使用？訪談對象對這個問題的看法不一。雖然絕大多數的受訪者 (87.9%) 或多或少信賴 AI 生成的程式碼品質，不過信賴程度普遍偏低，其中 39.2% 的人表示不太信賴，11.9% 的人則完全不信賴。詳細資料請見圖 5。



信賴 AI 生成程式碼的程度



誤差線代表 89% 的不確定性區間

圖 5：受訪者對 AI 生成的程式碼品質有多信賴。

從問卷調查的結果可以看出，開發人員正在迅速採用 AI、仰賴這項技術，並認為 AI 對成效所有助益，因此「對 AI 普遍缺乏信賴」的事實讓我們驚訝不已。值得一提的是，許多訪談對象表示，他們願意先調整 AI 生成的程式碼，再用到工作上，或組織有規定他們必須這麼做。

針對開發人員需要評估及修改 AI 生成的程式碼，某位訪談對象甚至做出下列比喻：「這就像是 Stack Overflow 的早期階段，大家總是認為 Stack Overflow 中的使用者都經驗老到，確實知道自己在做什麼，因此就會直接複製程式碼並貼入自己的專案，然後就出事了」(訪談對象 2)。

但也可能是因為這個問題以前發生過，訪談對象 3 等某些人認為所屬的公司「並不擔心，這就像是直接複製 Copilot 或 ChatGPT 的程式碼貼上，既有的好幾道程式碼品質查驗程序應該能抓出錯誤。」

我們假設開發人員不一定完全相信 AI 生成的程式碼準確無誤，但這也不代表他們認為 AI 生成的程式碼不實用。開發人員似乎可以接受不完美的程式碼，只要大部分都正確，需要稍加修改也沒關係。AI 生成的程式碼具有充分價值，足以吸引開發人員大範圍採用，而且可用既有的品管程序查驗。

AI 的未來展望

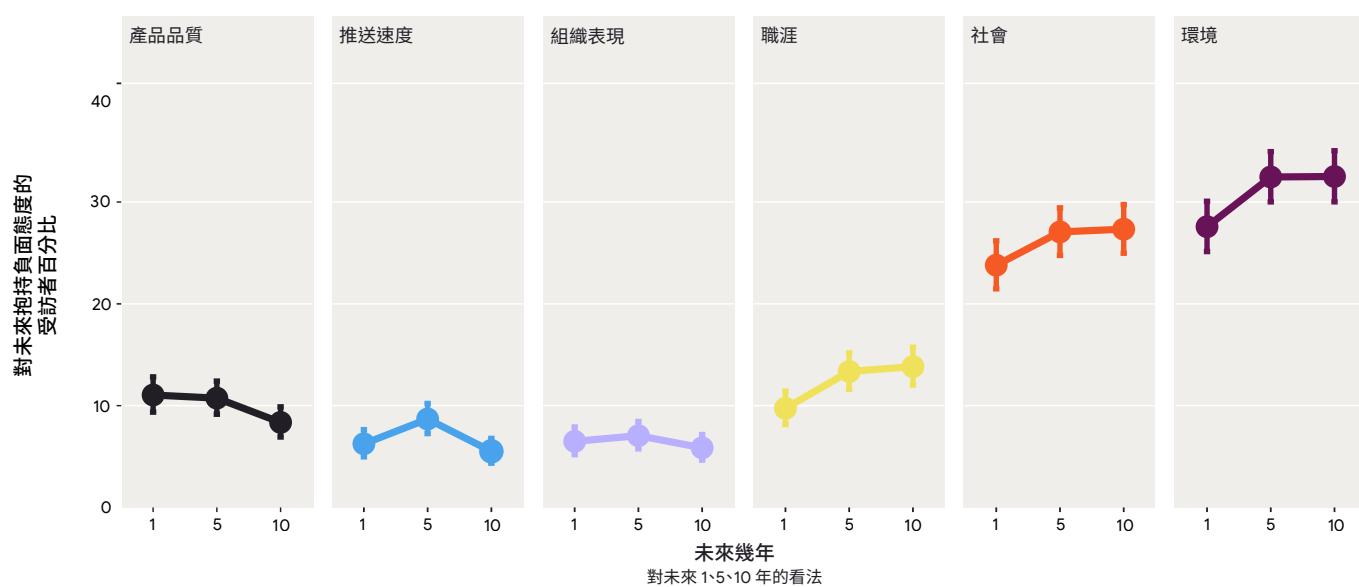
整體研究結果顯示，AI 已對專業開發人員的工作產生巨大影響，這股熱潮應該會持續擴大。雖然無法準確預測 AI 日後會對開發工作和全世界帶來哪些影響，我們仍請受訪者推測並預估 AI 在接下來 1 年、5 年和 10 年的影響。

依據受訪者近期的體驗，他們認為 AI 會對開發工作大有助益，不過針對 AI 日後的影響，預估結果就不那麼樂觀。

樂觀來說，AI 會提高專業開發人員的表現，這與我們的研究結果相符。受訪者預期在未來 1 年、5 年和 10 年，自家產品的品質會因為 AI 而持續提高。

不過，受訪者也預估 AI 會對職涯、環境和社會整體產生淨負面影響，而且大約 5 年內就會完全顯現。詳細資料請見圖 6。

預估的 AI 負面影響



誤差線代表 89% 的確信區間

圖 6：受訪者預估 AI 在未來 1 年、5 年和 10 年帶來的負面影響。

訪談對象對於 AI 未來的影響有喜有憂，這與問卷調查受訪者相似。有鑑於相關法規尚未擬定，某些人擔心日後可能面臨的法律問題，表示「萬一他們的使用行為在未來視為違法，那就麻煩大了」(訪談對象 3)。

還有些人提及長久以來的顧慮，那就是「AI 會取代人類嗎？誰知道呢？確實有可能。」(訪談對象 2)；不過，也有人認為沒什麼好害怕，並以過去的經驗為例，表示「大家以前認為 Y2K 要來了，世界要毀滅了！但講了一大堆，其實只是因為當時還不瞭解新技術。

結果誰也沒有被取代，反而出現更多工作。我相信 AI 也會是如此」(訪談對象 1)。

AI 日後對世界的影響仍然不明，但今年的問卷調查明確顯示，AI 已為軟體開發領域帶來不容忽視的典範轉移，而且專業開發人員已確實感受到變化。



1. <https://www.sciencedaily.com/releases/2024/03/240306144729.htm>
2. <https://tech.co/news/list-ai-failures-mistakes-errors>
3. <https://klyker.com/absurd-yoga-poses-generated-by-ai/>
4. <https://dora.dev/dora-report-2023>
5. Everett M. Rogers、Arvind Singhal 與 Margaret M. Quinlan, 「Diffusion of innovations」(創新的擴散)，《An integrated approach to communication theory and research》(傳播學理論與研究的整合式做法)。Routledge, 2014 年。第 432 至 444 頁，Tornatzky, L. G. 與 Fleischner, M., 1990 年，《The processes of technological innovation》(技術創新的過程)。麻薩諸塞州萊辛頓市：Lexington Books
6. 訪談對象 [數字] (例如「訪談對象 1」) 用來代稱訪談對象。

探討 AI 的 下游影響



重點摘要

這個章節會從個別開發人員到整個組織，探討採用 AI 在各層面的影響。我們發現 AI 帶來的影響相當複雜，有明顯的優點，也有意料之外的缺點。採用 AI 可以大幅提高個別員工的工作效率、心流狀態和工作滿意度，不過也會減少投注於高價值工作的時間。

同理，在程式碼品質、說明文件編寫和審查程序，AI 會帶來正面影響，不過出乎意料的是，軟體推送成效並沒有隨著這些效益提高。事實上，採用 AI 反而對這個部分不利，對產品效能的影響也微乎其微。

雖然採用 AI 會有這些挑戰，團隊和組織的表現仍然會因此提高。這個章節的結尾呼籲大家嚴格評估 AI 在軟體開發工作扮演的角色，並主動調整應用方式，創造最大效益及減少未預期的後果。

AI 時代與 DORA

預估科技巨頭在未來五年會投資約 \$1 兆美元來發展 AI¹。這與「[人工智慧：採用情形與觀感](#)」章節中的統計資料相當吻合：81% 的受訪者表示，所屬公司已將資源轉為投入 AI 發展。

AI 對環境的影響會進一步提高成本。某些預估顯示，2030 年之前，AI 會讓資料中心的能源需求上升 160%²。訓練 AI 模型所需的電力大約會超過「美國 1,000 個住家一年的用電量」³，難怪有超過 30% 的受訪者認為，AI 會對環境造成負面影響。

除了開發和環境成本，採用成本也有可能產生影響。

採用成本的形式有很多種，包括工作效率降低、必須聘請專業人才等，甚至涉及社會層面：超過三分之一的受訪者相信，AI 會在未來十年對社會造成危害。基於這些成本，人們當然會非常好奇 AI 可以帶來的投資效益。

相關的媒體、文章和研究相當多，這體現了人們對 AI 的好奇心。不過這些內容的看法和資料相當分歧，至少某種程度上是如此。



有些人認為 AI 大幅提高人類能力⁴，部分人則覺得 AI 只不過是稍有用處的家庭作業輔助工具⁵，也有些人害怕 AI 會導致人類滅亡⁶。

針對「順利完成特定工作的能力」等關係較近的成果，證據都極為正面⁷。以團隊的程式碼集等關係較遠的成果來說，結果就開始變得稍微不明確，也沒那麼正面。舉例來說，某些研究指出程式碼修改量可能會比 2021 年以前的基準高出一倍⁸。

解讀這些下游影響無疑是個挑戰，原因與影響之間的距離越遠，關係就越不清楚明確。

評估 AI 的下游影響，就好比將石頭拋入湖中後產生的影響量化。您可以輕鬆斷定最靠近石頭入水處的漣漪是由石頭產生，但離入水處越遠，石頭的影響就越不明確，這時就較難判斷波紋是不是由石頭產生。

採用 AI 就像將石頭拋入充滿其他程序和機制的洶湧海洋，難以判斷波浪是不是真的由 AI (或任何技術或做法) 引起。這或許解釋了為何業界一直難以採用一套標準化的衡量方法和數據分析框架，瞭解 AI 的影響力⁹。

DORA 的研究方法經過特別設計，可以因應這些類型的難題，讓您瞭解某項做法的正面與負面效用。過去 10 年來，我們探討了多種做法的下游影響，包括安全防護做法、轉型領導力、創造型文化、說明文件編寫做法、持續整合、持續推送軟體更新，以及重視使用者等¹⁰。

我們相信 DORA 的做法¹¹有助於瞭解 AI 的影響力，尤其是探究 AI 對多項結果的影響。



衡量 AI 採用情形

要掌握採用 AI 的影響，第一個難關就是衡量 AI 採用情形。我們認為比起計算使用頻率，衡量仰賴程度會更有意義，這樣才能瞭解 AI 在開發工作流程的扮演的角色有多重要。您可能每個月或每隔幾個月，才會審查程式碼或編寫說明文件幾次，但這些事項對您的工作極為重要。

相對來說，經常使用 AI 不代表您認為該項工作很重要或屬於您的職務重點。

因此，我們請受訪者提供他們在一般情境和特定工作對 AI 的仰賴程度。如要查看詳細的問卷調查結果和意義解讀，請回顧[前一個章節](#)。

透過因素分析，我們發現「一般情境」AI 仰賴程度問卷調查中的項目，與下列工作的 AI 仰賴程度高度重疊：

- 編寫程式碼
- 總結資訊
- 解釋程式碼
- 程式碼最佳化
- 編寫說明文件
- 編寫測試

這六個項目具有高度共通性和共變性，是衡量 AI 採用情形的基礎因素。

AI 對個別員工顯然有所助益 (也可能造成一些權衡取捨)

我們比照往年的做法，衡量與個別員工成就和身心健康有關的多個層面：

工作滿意度

這個項目可以判斷員工對自身工作的整體感受。

工作倦怠

這項因素涵蓋工作倦怠的各個層面 (例如身體、情緒和心理方面)，以及對個人生活的影響。

心流狀態

這個項目可以判斷個別員工在處理開發工作時，通常能多專注。

工作效率

這個因素可衡量個別員工是否認為自己工作效率、成果好；是否覺得能創造價值、完成工作。

處理繁瑣工作的時間

這個項目可以衡量個別員工投注於重複性手動工作的時間占比，這類工作幾乎無法提供長期價值。

處理高價值工作的時間

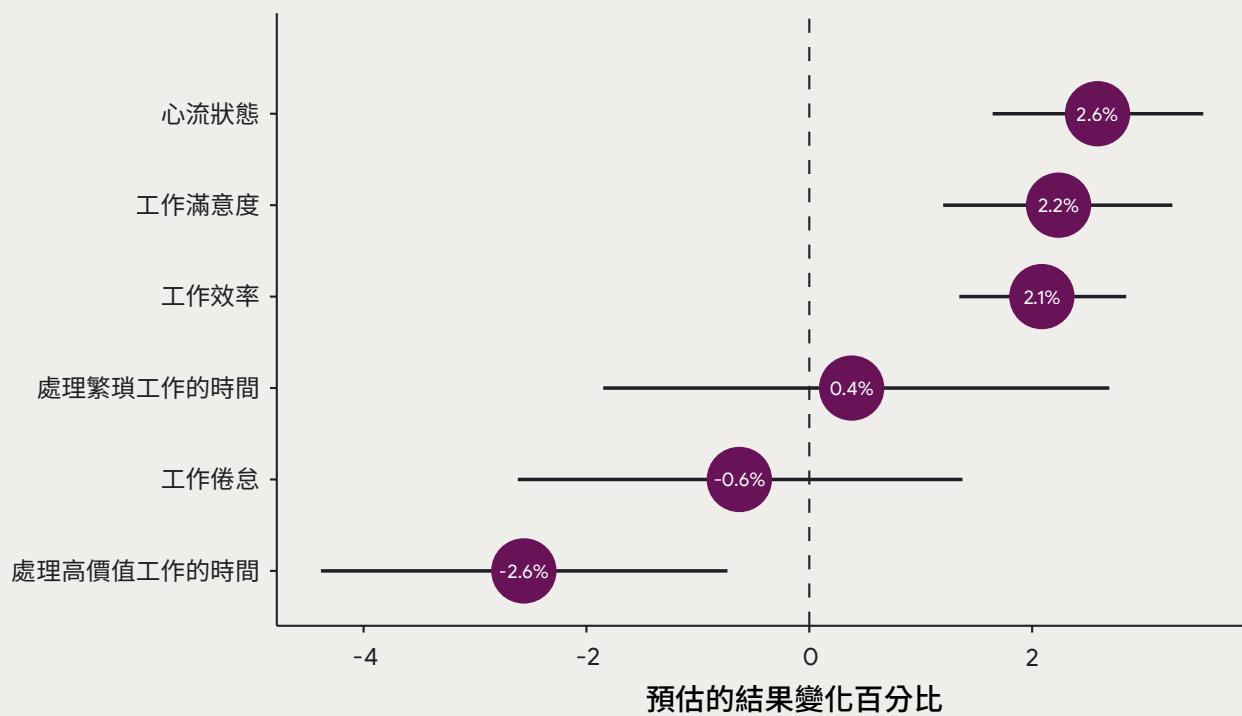
這個項目可以衡量個別員工投注於高價值工作的時間占比。

我們想瞭解受訪者的回答會不會因為採用 AI 而不同，而研究結果也顯示經常會如此。

我們針對員工的成就和身體健康影響，盡可能地預估了採用 AI 帶來的影響，並呈現在圖 7 當中。

如果個別員工的 AI 採用率提高 25%...

結果



圓點 = 預估值

誤差線 = 89% 的不確定性區間

圖 7:AI 採用情形對個別員工成就和身心健康的影響

明確助益

多數案例中，個別員工採用 AI 可以享有助益，不過當然也會有一些小問題。**可以肯定的是，AI 會為心流狀態、工作效率和工作滿意度帶來顯著的正面影響（詳見圖 7）。**

舉例來說，如果某人的 AI 採用率提高 25%，工作效率可能會提高近 2.1%（詳見圖 7）。儘管增幅不大，但這只是從個別員工的角度來看。如果數十位或數萬位開發人員都能享有這樣的助益，結果會相當驚人。

這個情況與預期相同。我們相信，有一部分要歸功於 AI 能集中整合不同的資訊來源，並提供高度個人化的回覆。如要憑一己之力做到這件事，不僅曠日費時、需要多次切換背景脈絡，而且較難進入心流狀態。

由於工作效率和心流狀態與工作滿意度有著緊密關係，採用 AI 可以提高工作滿意度也不足為奇。

潛在的權衡取捨

不過，事情並沒有想像中那麼簡單。採用 AI 的其中一項價值主張是，AI 可讓我們將更多時間投注於高價值工作。也就是說，AI 可以自動處理繁瑣的重複性手動工作，受訪者理應有更多時間能夠處理「更重要的事務」。但資料顯示，AI 採用率提高可能會帶來反效果：處理高價值工作的時間減少，投注於繁瑣工作的時間則似乎沒有變化。

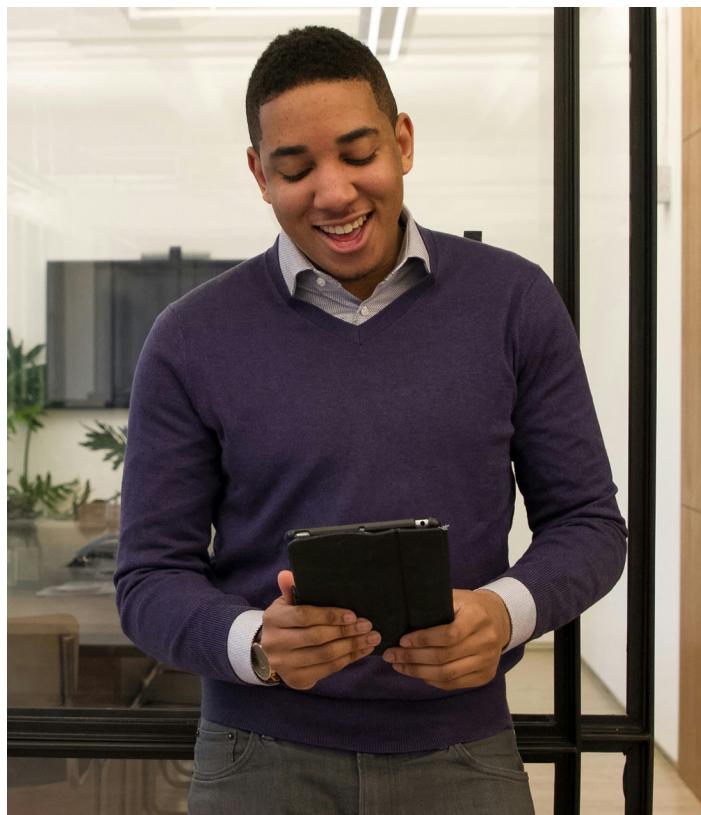
一直以來，心流狀態、工作效率等身心健康指標，與投注於高價值工作的時間都存在關聯。因此，觀察結果讓我們相當訝異，這三項身心健康指標提高，投注於高價值工作的時間反而降低。

如要針對這些模式提出恰當的解釋，我們必須反證這個看似不協調的變化。就像好的電影解析一定要談到與論點相衝突的橋段，好的書評也不能忽略說不通的章節。同理，我們不能僅憑一部分的情況就以偏概全，否則很難妥善解釋這些模式。

可以解釋這種情況的假說有無數種，不過我們想出了最能同時解釋下列三個現象的假說：1) AI 讓心流狀態、工作效率和工作滿意度提升；2) 投注於高價值工作的時間減少；3) 投注於繁瑣工作的時間不變。

我們稱之為「真假說」：AI 提升了使用者的產能和心流狀態，因此工作效率增加，讓使用者更快完成高價值工作。

省下來的時間就是所謂的「真空」，也就是說，AI 並未從受訪者的工作中偷走價值，而是加快實現價值。



那什麼是高價值工作？

為理解這些出乎意料的研究結果，我們進一步向受訪者詢問何謂「高價值」或「繁瑣」的工作。

依據過往的經驗、報告和訪談的質性資料，受訪者認為開發相關的作業（例如程式設計）為高價值工作，價值較低（甚至繁瑣）的工作則通常涉及組織內協調事宜，像是參加會議。按照這樣的分類方式，受訪者認為 AI 較適合用來輔助「高價值」工作，而非「繁瑣」工作。

分析訪談的質性資料之後，我們發現提問者詢問訪談對象是否認為自己的「有意義」時，他們經常依據自身工作對他人的影響來衡量工作的價值。

按照過去某兩年 DORA 研究「重視使用者」是否能大幅提高工作滿意度時的結果，這個論點為真。

舉例來說，問及最近轉換職務的原因時，訪談對象 10¹² 表示「這讓我能影響更多人和更多事」。訪談對象 11 同樣表示「從頭開始建構產品並交付給顧客或客戶，不僅充滿成就感，也會感到興奮，因為自己成功打造出大家會使用的產品！」

開發工作的「意義」並非直接來自編寫程式，而是取決於建立解決方案帶來的影響。這點就能說明為何受訪者投注於高價值工作的時間減少，同時工作滿意度卻提高。

AI 可協助使用者更快速輕鬆地完成高價值工作，似乎卻無法對他們不喜歡的工作有所助益。採用 AI 未讓繁瑣的作業和工作倦怠消失，開發人員還是無法藉助 AI 避開枯燥乏味的會議、官僚體系和其他許多麻煩的工作（詳見圖 8）。

好消息是 AI 並未使情況變得更糟，也沒有危害受訪者的身心健康。

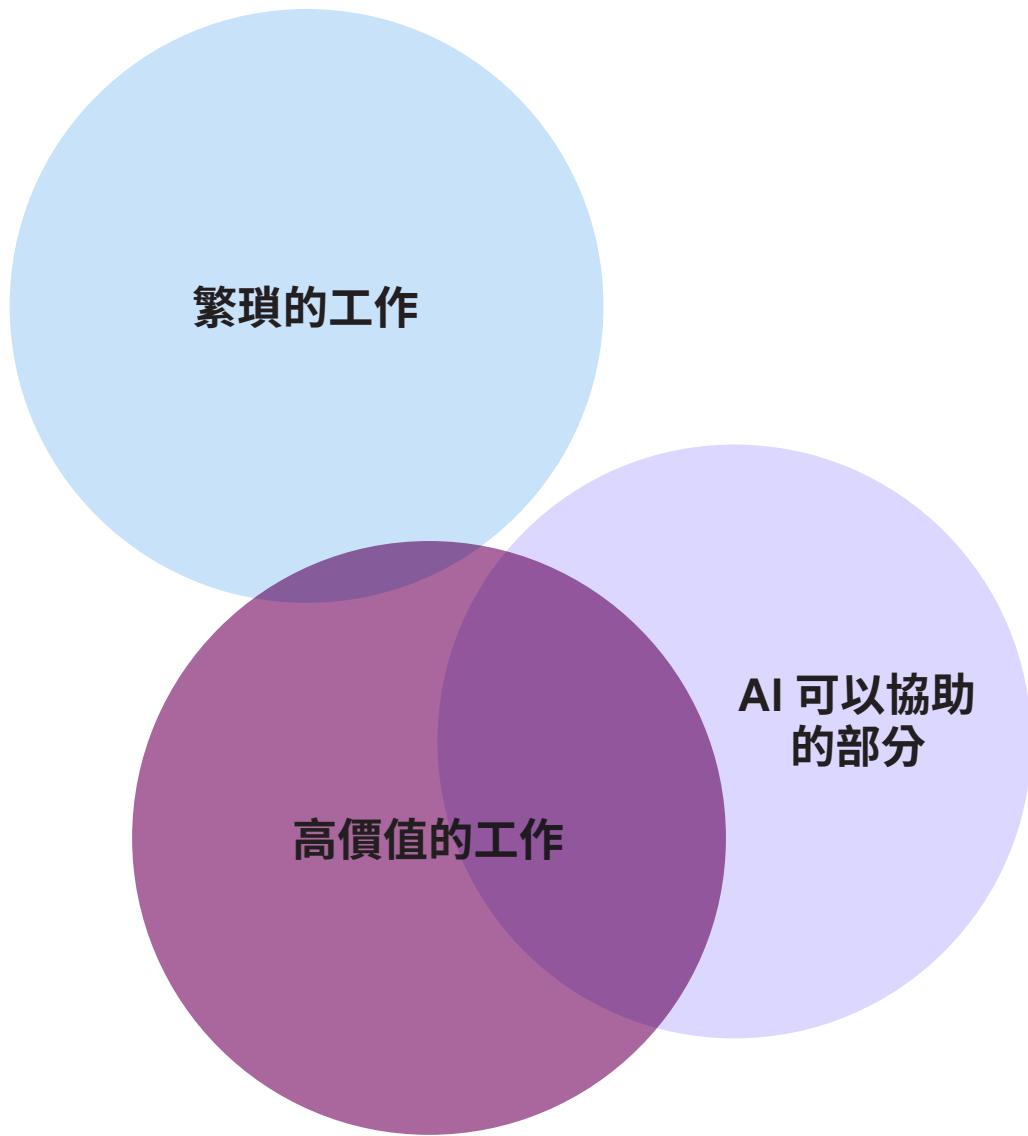


圖 8：僅以圖形呈現假說，並非真實資料：AI 對高價值工作所有助益，但無法協助繁瑣的工作。

AI 應用於開發工作流程的 前景樂觀

前一節探討了 AI 對個別員工的影響，接著將焦點轉向程序、程式碼集和團隊協調。以下列出我們衡量的結果：

程式碼複雜度	因程式碼複雜深奧而降低工作效率的程度。
技術債	在過去六個月，主要應用程式或服務內既有技術債對工作效率的負面影響。
程式碼審查速度	審查完主要應用程式或服務的程式碼，平均需要多少時間。
核准速度	從提議主要應用程式或服務的程式碼變更，到獲准將程式碼推送至正式環境通常需要的時間。
跨職能團隊 (XFN) 協調	對下列敘述的同意程度：「在過去三個月，我能有效與跨職能團隊的成員合作。」
程式碼品質	在過去六個月，對主要應用程式或服務基礎程式碼品質的滿意程度。
文件品質	對手冊、README、程式碼註解等內部說明文件的觀感，包括可靠性、尋獲度、更新情形和實用性。

如同上文，此處的目標是想瞭解採用 AI 是否會對這些面向產生影響。我們盡可能預估 AI 採用率提高 25% 之後，這些面向會有何變化，並呈現在圖 9 當中。

整體來說，下述的變化模式，為 AI 效益提供了強力論點，這一節的重大結果如下。

AI 採用率提高 25% 會帶來下列影響：

說明文件編寫品質提高 7.5%

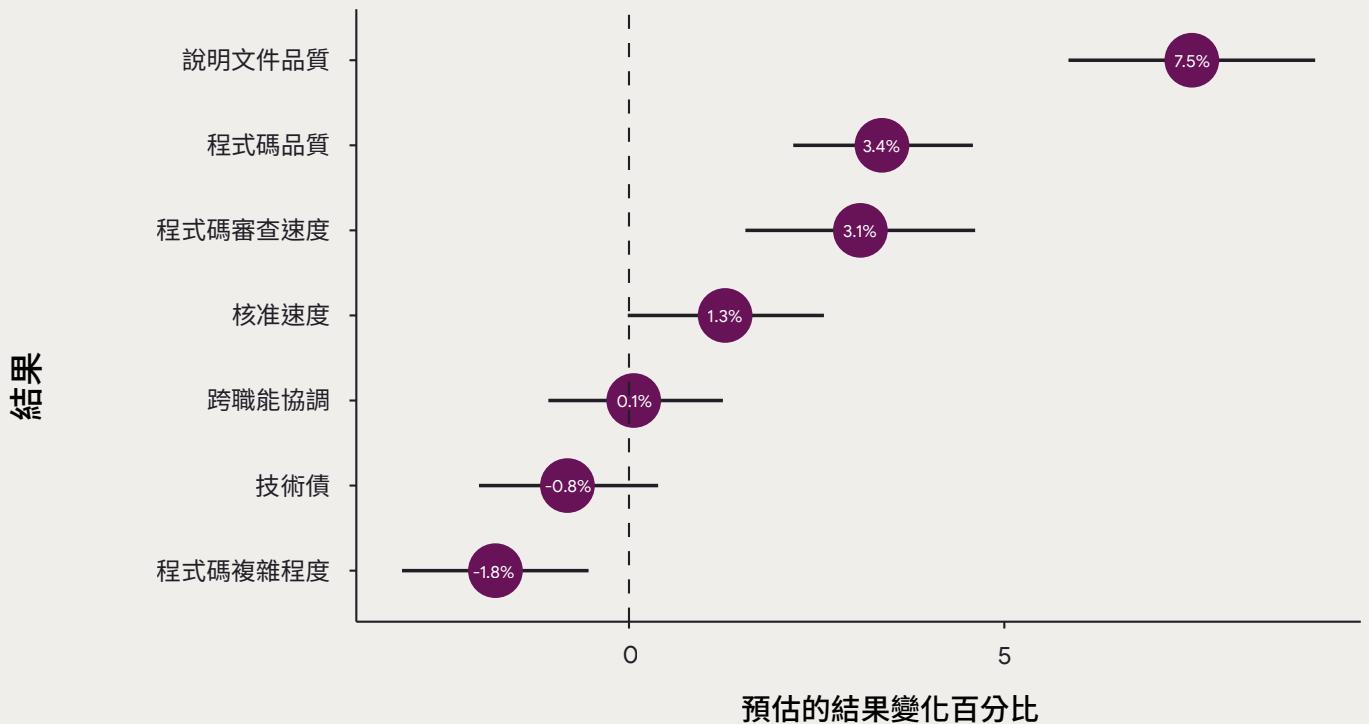
程式碼品質提高 3.4%

程式碼審查速度加快 3.1%

核准速度加快 1.3%

程式碼複雜度降低 1.8%

如果 AI 採用率提高 25%...



圓點 = 預估值

誤差線 = 89% 的不確定性區間

圖 9:AI 採用情形對組織的影響。

「[人工智慧：採用情形與觀感](#)」一節中的資料顯示，AI 最常用於協助編寫程式碼。67% 的受訪者表示 AI 有助於提高程式碼品質，這一節的研究結果進一步印證這樣的觀點。AI 似乎可以提高程式碼品質並降低複雜度（詳見圖 9）。如果再重構舊有的程式碼，AI 生成的優質程式碼就有可能提高程式碼集的整體品質。使用者可以透過 AI 生成優質的說明文件，並運用 AI 充分掌握說明文件內的資訊，進而讓程式碼集更臻完善（詳見「[人工智慧：採用情形與觀感](#)」章節）。

程式碼的品質越高，就越容易審查及核准。資料也清楚呈現一個模式：將 AI 用於程式碼審查，也能縮短查驗及核准程式碼的時間（詳見圖 9）。

當然，縮短程式碼審查與核准的時間，並不代表開發人員能更妥善周全地執行這兩項程序。速度變快有可能是因為過度仰賴 AI 來執行程序，或是過於信賴 AI 生成的程式碼。這個結論與圖 9 中的模式並不衝突，但也不能輕易蓋棺論定。

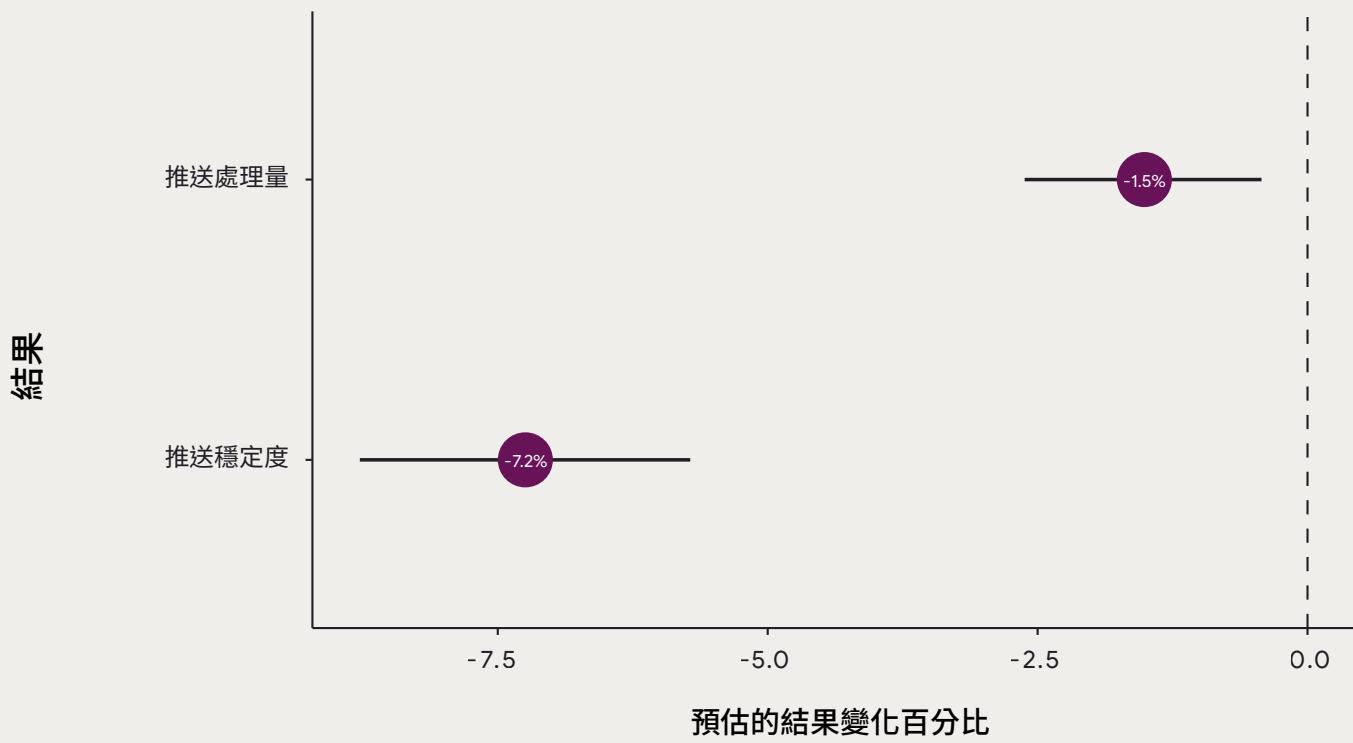
另外，我們無法確定說明文件和程式碼品質提高是因為 AI 生成了這類內容，還是 AI 強化了開發人員的能力，可以從本來認為低品質的程式碼和說明文件中取得價值。萬一這只是因為 AI 強大到可以幫助我們理解程式碼和說明文件，以至於我們對「優質程式碼和說明文件」的標準稍微降低呢？這兩種解讀方式並不互斥，兩者都有可能形成這些模式。

這些模式中，AI 可以充分發揮所用說明文件與程式碼集的價值，這點應該沒有爭議。AI 也有助於減少程式碼審查和核准程序中的瓶頸，進而降低費用。不過，我們還是難以得知 AI 究竟如何帶來這項改變，以及這些效益是否能在更下游產生助益，例如提高軟體推送成效。

AI 降低推送成效

過去幾年我們開始發現，軟體推送作業的處理量和穩定度指標，有時會互不相關。雖然過往認知的處理量和穩定度相關性仍然存在，越來越多證據顯示這兩項因素彼此獨立，有必要分開討論。

如果 AI 採用率提高 25%...



圓點 = 預估值

誤差線 = 89% 的不確定性區間

圖 10：AI 採用情形對推送作業處理量和穩定度的影響。

研究結果表明，採用 AI 會對軟體推送成效帶來負面影響，這與我們的預期相反。我們發現 AI 對推送作業處理量的影響相當小，卻有可能是負面影響：AI 採用率每提高 25%，處理量預估就會降低 1.5%。AI 對推送作業穩定度的負面影響較大：AI 採用率每提高 25%，穩定度預估就會降低 7.2%。詳細資料請見圖 10。

過往研究發現，改良軟體開發程序（包括提高說明文件品質、程式碼品質、程式碼審查速度、核准速度，以及降低程式碼複雜度），可增進軟體推送成效。因此我們很訝異，AI 可以提高這些程序指標，卻有可能降低推送作業的處理量和穩定度。

考量前幾年的研究結果，我們推測雖然 AI 提高了受訪者的程式碼生成速度和工作效率，但這項根本的典範轉移可能導致業界忽略了 DORA 其中一項最基本的原則：小批量工作的重要性。意即 AI 可讓受訪者在相同的時間內生成更多程式碼，異動清單可能會隨之變長。往年的 DORA 都指出異動越大，處理速度就越慢，並較有可能產生不穩定性。

綜合這些資料，我們認為改良開發程序不代表軟體推送成效就一定會提高。如未妥善遵守軟體開發成功基本原則（例如小批量工作和完善的測試機制），推送成效就更不可能提高。

AI 可以對許多重要的個別員工與組織因素帶來正面影響，進而產生出色的軟體推送成效，因此我們可以保持樂觀，但不宜將 AI 視為萬靈丹。



表現出色的團隊和組織都使用 AI， 但效益似乎未擴及產品

接著看看 AI 對最下游結果的影響：

組織表現

這個因素分數反映組織在下列面向的表現：整體表現、獲利能力、市占率、顧客總數、營運效率、顧客滿意度、產品/服務的品質，以及達成目標的能力。

團隊表現

這個因素分數反映團隊在下列面向的能力：合作、創新、高效率工作、相互支援和適應彈性。

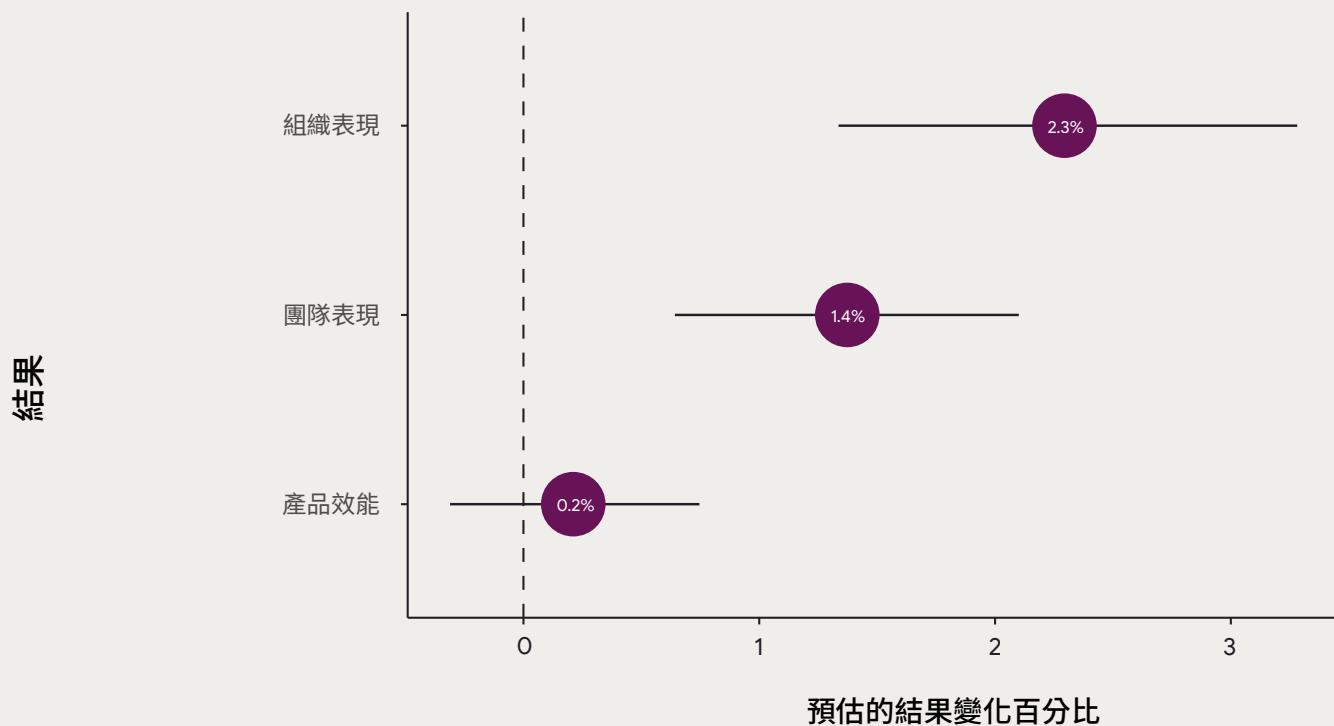
產品效能

這個因素分數反映產品的可用性、功能、價值、供應情形、效能 (例如延遲時間) 和安全性。

這些結果與個別員工採用 AI 的關係不僅難以推論，干擾因素也相當多，有時就像是在分析「今天午餐了吃什麼」對「組織今年表現」的影響。

我們會在微觀（例如個別員工）與宏觀（例如組織）視角之間切換，這個做法有其邏輯。《研究方法》章節會說明跳躍式推論。我們目前瞭解相關性即可：

如果 AI 採用率提高 25%...



圓點 = 預估值

誤差線 = 89% 的不確定性區間

圖 11：AI 採用情形對組織表現、團隊表現和產品效能的影響。

組織表現和團隊表現均因採用 AI 而提高。AI 採用率每提高 25%，組織表現預估會提高 2.3%，團隊表現則會提高 1.4%（詳見圖 11），不過產品效能似乎與 AI 採用率沒有明顯的相關性。現在來試著解讀這些影響背後的原因。

我們推測，帶來出色團隊和組織表現的原因，與影響產品效能的原因不同。

團隊和組織非常仰賴溝通、知識共享、決策和健康的文化。AI 可能會解決這些領域中的某些瓶頸，為團隊和組織帶來正面影響。

不過，產品效能或許牽涉更多因素。產品要出色，背後的原因肯定與表現優異的團隊和組織相似，不過與開發工作流程和軟體推送作業的關聯性可能更直接、相近，而這兩者在導入 AI 技術之後，可能還需要一些時間才能穩定下來。

在出色產品的背後，技術層面有其獨特的重要性，這或許是原因之一，但產品設計的藝術和人類的同理心，同樣占有一席之地。如果認為事事都是問題，都能透過運算解決，或許會難以相信這個論點，不過產品開發工作中的某些部分（例如創意或使用者體驗設計），仍然很仰賴人類的直覺與專業知識，而且或許永遠都會是如此。

可以肯定的是，組織表現、團隊表現和產品效能之間息息相關。進行皮爾森雙變數相關分析之後，我們發現產品效能與團隊表現（相關係數 = 0.56, 95% 的信賴區間介於 0.51 至 0.60 之間）和組織表現（相關係數 = 0.47, 95% 的信賴區間介於 0.41 至 0.53 之間）都呈現中度正相關。

這些結果會相互影響，證明其相依性。表現出色的團隊通常可以開發出更優質的產品，但若接手品質欠佳的產品，成效可能就不會這麼亮眼。同理，表現出色的組織可以透過資源和程序培養出表現出色的團隊，不過組織一旦遭遇困難，團隊表現就有可能下滑。因此，如果採用 AI 可以大幅提高團隊和組織表現，產品效能理當也會提高。

人們最近才開始採用 AI，某些優點和缺點可能需要一段時間才會顯現，因為 AI 的影響可能先天就較不明顯，或是我們需要學習一段時間才能有效運用。

也或許，這代表我們必須先釐清 AI 對組織和團隊的助益，才能瞭解如何充分運用 AI 的潛力，來革新及開發產品。歡迎參閱圖 12 來瞭解可能的進程。

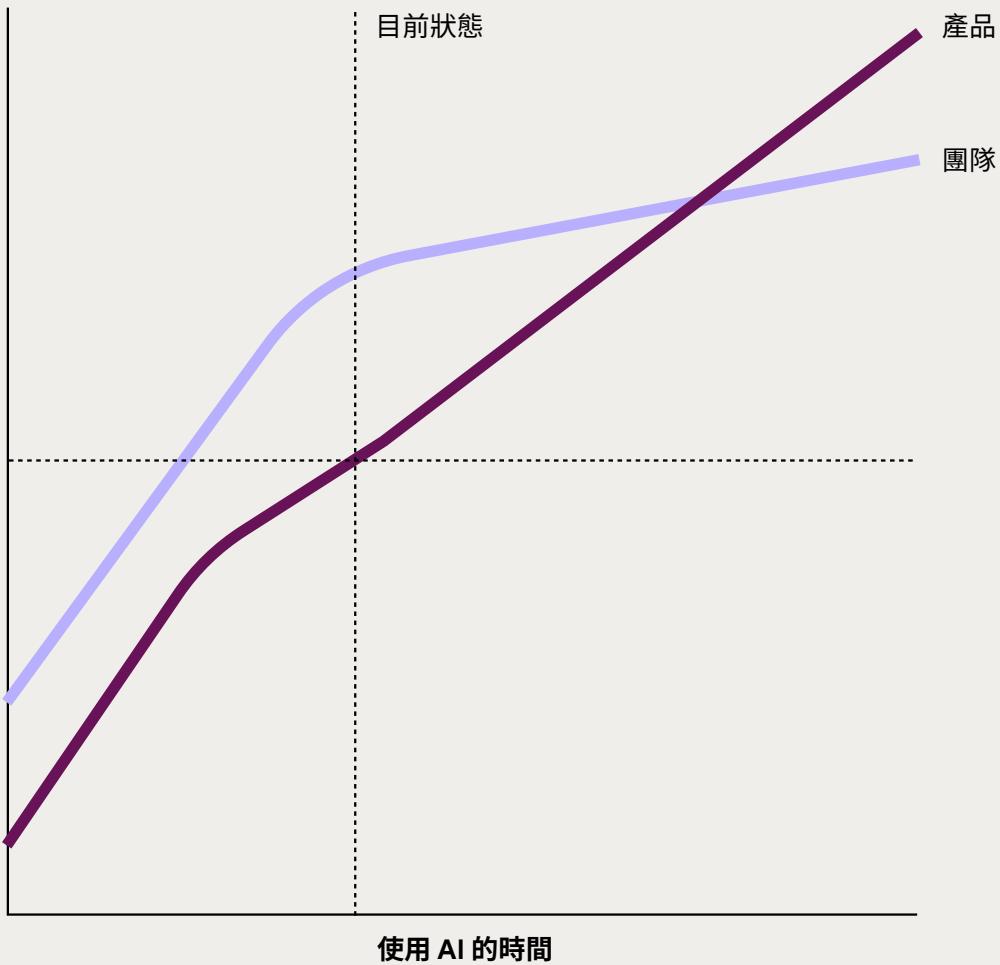


圖 12：不同學習曲線的示意圖。這個圖形僅呈現抽象概念，並非依據實際資料繪製。

接下來呢？

目前看來，AI 確實可以提高個別員工、團隊和公司的表現，因此我們想瞭解這項技術的潛能。當前浮現的趨勢證明這並不是誇大其辭，AI 確實會帶來影響。

有明確證據指出，採用 AI 有其好處，同時卻也有充分證據顯示，AI 可能會帶來許多潛在阻礙、難題和危害。

您可以參考下列幾個建議， 瞭解如何擬定 AI 採用策略：

大規模採用 AI 或許沒有想像中容易，必須擬定經過慎重考量、公開透明且可靈活調整的策略，才有機會獲得大量效益。這項策略須由主管階層、團隊、組織、研究人員和 AI 開發人員共同制定。

主管階層和組織必須設法決定要優先在哪些領域採用 AI，為員工提供最佳支援。

訂定明確的 AI 任務和政策來協助組織與團隊

秉持公開透明的原則，向員工說明 AI 任務、目標和採用計畫的資訊。如果能清楚傳達整體願景和細部政策，並解決程序相關疑慮（例如允許的程式碼使用位置和可用工具），將可減輕員工的疑慮，並將 AI 定位成實用工具，可協助所有員工專心處理價值更高、更具創意，也更能帶來成就感的工作。

營造持續學習及試驗 AI 的文化

提供專門的時間，讓個別員工和團隊探索有益的用途，並允許他們自行決定使用方式，藉此建立友善環境，鼓勵大家持續探索 AI 工具。在沙箱或低風險的環境實際試用 AI，加深對這類技術的信賴。您可以考慮專心開發完善的自動測試機制，進一步降低風險。請導入評估框架，且不能單純以 AI 採用率來衡量成效，而是要考量實質的下游影響，也就是 AI 如何提高員工的工作效率、協助團隊發揮潛能，並讓產品使用者受惠。

體認 AI 的缺點並善用其優點，取得競爭優勢

AI 可能會造成負面影響，像是減少投注於高價值工作的時間、使員工過度依賴 AI、影響軟體推送作業的穩定度和處理量等，帶來的優勢也可能反而變成其他領域的新挑戰。但若能瞭解、認知到這些缺點，您就有機會避開問題，帶領組織和團隊踏上正確的道路，充分運用 AI。除了掌握 AI 可能帶來的效益，您也需要讓員工瞭解其危害，這樣才能加快學習速度、探索更多工具，並將經驗轉化為行動和真正的競爭優勢。

顯然 AI 領域還有很多值得期待及學習的事物。如同過去十年，我們會持續透過 DORA 掌握動態，盡力提供真實、準確且實用的觀點。

1. <https://www.goldmansachs.com/insights/top-of-mind/gen-ai-too-much-spend-too-little-benefit>
2. <https://www.goldmansachs.com/insights/articles/AI-poised-to-drive-160-increase-in-power-demand>
3. <https://www.washington.edu/news/2023/07/27/how-much-energy-does-chatgpt-use/>
4. <https://www.gatesnotes.com/The-Age-of-AI-Has-Begun>
5. <https://www.businessinsider.com/ai-chatgpt-homework-cheating-machine-sam-altman-openai-2024-8>
6. <https://www.safe.ai/work/statement-on-ai-risk>
7. <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>
8. https://www.gitclear.com/coding_on_copilot_data_shows_ais_downward_pressure_on_code_quality
9. <https://www.nytimes.com/2024/04/15/technology/ai-models-measurement.html>
10. <https://dora.dev/capabilities>
11. 請記得 DORA 並非唯一途徑，但在 AI 領域算是特有的評估方法
12. 訪談對象 [數字] (例如「訪談對象 1」) 用來代稱訪談對象。

平台工程



簡介

平台工程是新興的工程學門，在業界逐漸引發興趣，也越來越熱門。這個產業的龍頭如 Spotify 和 Netflix，以及《Team Topologies》(團隊拓樸)¹ 等相關書籍，都讓大眾十分驚豔。

平台工程是社會技術學門，主要探討不同團隊之間的社交互動與交集，以及自動化處理、自助式服務和程序可重複性的技術層面。業界多年來一直鑽研平台工程背後的概念，DORA 也是其中之一。

一般來說，我們的研究著重於如何將軟體推送给外部使用者，然而平台團隊輸出的通常是主要供內部使用的 API、工具和服務組合，以便開發軟體及完成作業生命週期。

在平台工程領域，大量心力會集中於打造「黃金途徑」，讓開發人員享有更優質的體驗。有了這些高度自動化的自助式工作流程，平台使用者就能取用所需資源，來進行應用程式推送及操作。其目標為省去建構及推送軟體的複雜作業，讓開發人員專心處理程式碼。

許多工作都能透過「黃金途徑」自動化處理，包括佈建新的應用程式、佈建資料庫、管理結構定義、執行測試、建構及部署基礎架構，以及管理 DNS。

平台工程中的概念可能與「自己建構的服務自己執行」等做法相違背，像是將特定功能下移（有時稱為「向下轉移」）² 至共用系統。不過，我們認為平台工程可以擴大這些做法在組織的採用範圍，因為只要功能建構於平台，採用該平台的團隊自然能免費取得該項功能。

舉例來說，假設平台可以執行單元測試，並直接向開發團隊回報結果，而且團隊不需要建構及管理測試執行環境，那麼持續整合平台功能就能讓團隊專心編寫高品質的測試。以這個例子來說，持續整合功能可以在組織內部擴大採用，讓不同團隊可持續測試³，並將測試自動化⁴，藉此改良功能。

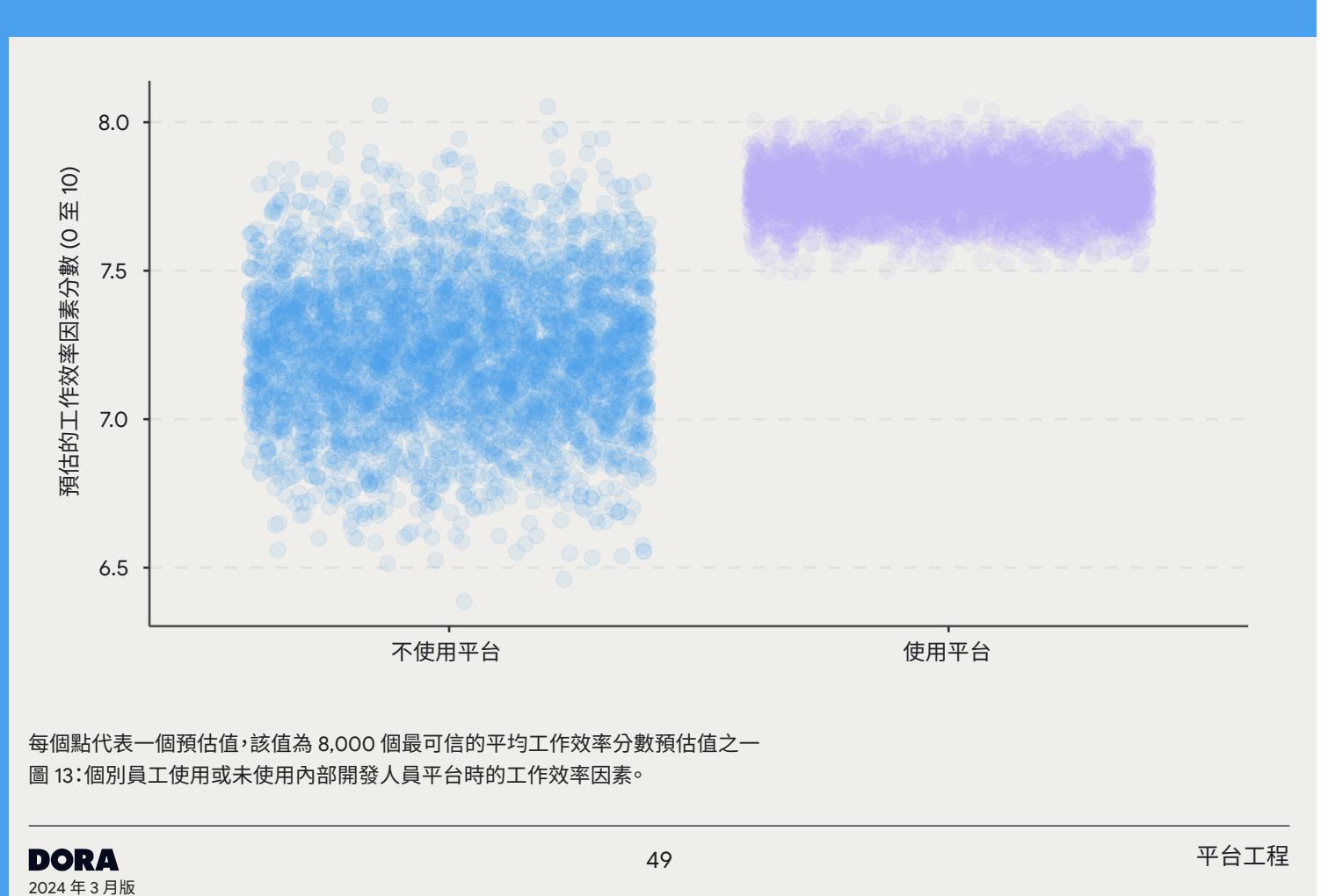


成功導入平台工程的關鍵，在於以使用者為中心（因為內部開發平台的使用者就是開發人員）、重視開發人員的獨立性，以及秉持產品思維。這點不會太讓人意外，因為在今年和往年的研究，我們都認為「以使用者為中心」是提高組織表現的關鍵因素⁵。如果不這麼做，使用平台反而會弊大於利。

今年的報告中，我們試著檢驗平台與軟體推送和運作成效的關聯性，並取得了一些正面的結果。使用內部開發人員平台之後，個別

員工的工作效率提高了 8%、團隊表現提高了 10%，組織的軟體推送成效和運作成效則提高了 6%。不過，平台也帶了一些負面影響，處理量和變更穩定度分別下降了 8% 和 14%，這讓我們相當驚訝。

下一節中，我們會深入探討數據和細微差異，並分享這次問卷調查的驚人結果。無論您的平台工程計畫是剛起步或已行之有年，應用關鍵的研究結果都能提高平台成效。



平台工程有望帶來的效益

採用內部開發人員平台有望提高效率和產能，因此受到大量軟體開發和IT產業人員的關注。在今年的問卷調查，我們對「內部開發人員平台」的定義相當廣泛⁶，並發現89%的受訪者表示會使用這類平台，互動模式則非常多樣化。

這些資料點印證了平台工程的興起，以及業界對這個領域的廣大關注。

整體來說，平台可以帶來效益。使用內部開發人員平台時，個別員工和團隊的工作效率分別提高了8%和10%。

除了工作效率之外，我們也發現採用平台可讓組織的整體表現提高6%。從整體來看，組織採用平台之後，可以快速推送軟體、滿足使用者的需求，並提高業務價值。

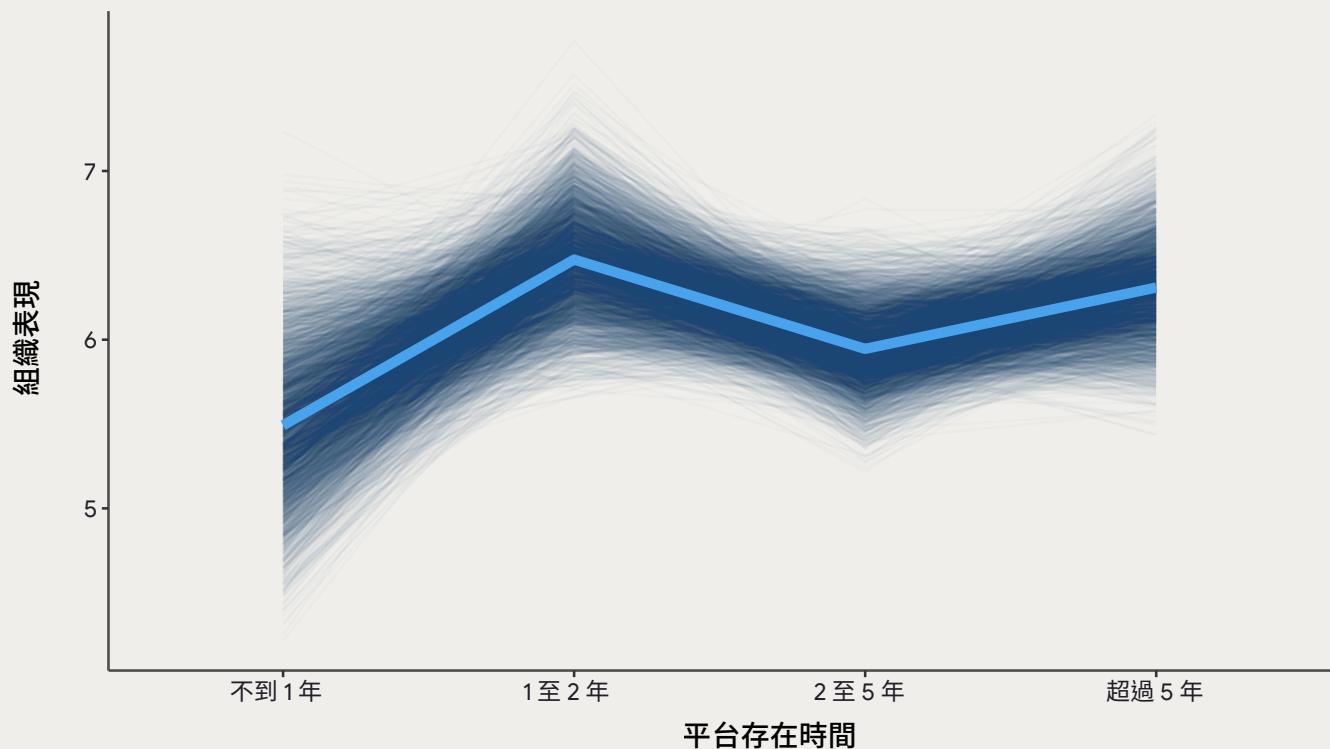


圖 14：採用內部開發人員平台的時長與組織表現的變化。

綜合考量採用平台的時長與工作效率之後，我們發現組織表現在平台工程計畫初期有所提升，然後下降，最後隨著採用平台的時間拉長、平台日漸成熟而回升。這符合轉型計畫的典型模式：初期可以取得效益，不過之後就會遭遇難題。

長遠來看，工作效率仍會提高，這顯示內部開發人員平台有望帶來整體效益，在軟體推送成效和營運程序扮演重要角色。



重要發現：開發人員獨立性的影響

使用內部開發人員平台推送軟體時，開發人員的獨立性會大幅影響個別員工和團隊的工作效率。開發人員獨立性的定義是「開發人員在整個應用程式生命週期獨立完成工作的能力，不需要團隊協助」。

我們發現，平台使用者不需要團隊協助就能完成工作時，團隊和個別員工的工作效率都提高 5%。這項發現呼應了平台工程的其中一項重要原則，也就是專注於提供自助式工作流程。

這對平台團隊來說相當重要，因為平台工程的其中一項關鍵程序就是收集使用者的意見回饋。問卷調查的回答並未指出何種形式的意見回饋最有效，不過常見的方法包括非正式對談和問題追蹤工具，其次是持續共同開發、問卷調查、遙測與訪談。

上述所有方法都能有效掌握使用者是否能獨立完成工作。問卷調查資料也顯示，不收集有關平台的意見回饋會帶來負面影響。

次要發現：專屬平台團隊的影響

有趣的是，成立專屬平台團隊對個別員工的工作效率幾乎沒有影響，不過團隊工作效率提高了 6%。成立專屬平台團隊雖然對個別員工有幫助，但對團隊的整體影響更為明顯，這種不一致的影響讓我們感到意外。

團隊中的不同開發人員會具備不同的技能並各司其職，因此相較於個別工程師，他們處理的工作類型自然更多樣化。成立專屬平台工程團隊或許可以提高平台的實用性，為工作類型多樣化的團隊提供更完善的支援。

整體來說，提供內部開發人員平台對工作效率有正面影響。

關鍵因素如下：

應以使用者為中心，打造可以自主完成的自助式服務和工作流程，讓開發人員能夠獨立工作。別忘了，這裡所說的平台使用者是您的內部工程和開發團隊。

平台工程與其他轉型計畫相同，都適用「J 曲線」，代表如果能持續進步，工作效率的提升也會越趨穩定。

意料之外的缺點

雖然平台工程肯定有一些優點，像是可以提高團隊和個別員工的工作效率，以及組織的整體表現，卻也有意料之外的缺點：處理量和變更穩定度降低。

很巧的是，我們發現變更穩定度降低與工作倦怠之間，有著非常值得探究的關係。

處理量

相較於未使用平台，使用平台後的處理量降低了近 8%。針對背後的可能原因，我們提出了下列假說。

首先，由於增加了這項機制，變更必須先通過平台，才能部署至正式環境，造成整體變更處理量降低。一般來說，如果採用內部開發人員平台來建構及推送軟體，軟體在不同系統之間「轉手」的次數通常會增加，團隊之間的交接負擔也會悄悄加重。

舉例來說，程式碼推送至原始碼控管機制之後，不同系統會自動認領程式碼，以便執行測試、安全性檢查、部署和監控的工作。

每當執行上述轉手程序時，整體程序就有可能耗費更多時間，導致處理量降低，不過從結果來看，工作成效仍然有所提高。

其次，針對按照規定「在整個應用程式生命週期，只能使用該平台執行工作」的受訪者，我們發現處理量降低了 6%。雖然兩者之間不一定有關聯性，但第一個假說或許也能解釋這個現象。

如果採用平台之後，開發及發布軟體時需要使用更多系統和工具，規定使用的平台不適用，或是程序因平台而自然延宕更久，這時限定使用該平台就有可能造成工作效率降低。

要解決這個問題，執行平台工程計畫時，請務必秉持以使用者為中心的心態，並盡量讓員工能夠獨立工作。

變更穩定度降低與工作倦怠

透過內部開發人員平台開發及運作的應用程式方面，我們驚訝地發現變更穩定度降低了 14%。這代表使用平台時，變更失敗率和重新處理率大幅提高。

更有趣的是，研究發現採用平台後產生的不穩定度與工作倦怠程度提高具有關聯性。這並不代表平台是工作倦怠的罪魁禍首，但不穩定度和採用平台這兩件事如果同時發生，特別容易引發工作倦怠。如同處理量降低的情況，我們無法百分百肯定為何工作倦怠程度會變動，但是有一些推測。

首先，平台可讓開發人員和團隊更有信心地推送變更，因為就算變更品質不佳也能迅速修復。在這種情況，不穩定度變高就不一定是壞事，因為團隊可以在平台試驗及推送變更，而這會讓變更失敗率和重新處理率提高。

第二種推測是，平台無法有效確保變更和/或正式環境部署項目的品質。

另一種可能性是平台具備自動測試功能，會執行應用程式中的所有測試，不過應用程式團隊首重處理量（而非品質），而且未改良測試，因此未充分運用這項功能。在這兩種情境，品質不佳的變更都會通過程序，因此必須重新處理。

第三種可能性是，如果團隊的變更不穩定度和工作倦怠程度偏高，他們打造平台的目的通常是為了提高穩定度並減少工作倦怠。平台工程做法的目的往往就是為了減少工作倦怠，以及提高持續推送小型變更的能力，因此這個可能性有其道理。如果這個假說成立，採用平台工程做法的組織應該有工作倦怠和變更不穩定的問題。

在前兩種情境，採用平台會讓重新處理率變高，進而產生負擔並有可能引發工作倦怠。這個情況在第二種情境尤其明顯，平台可能會通過品質不佳的變更，進而加重工作倦怠的情形。不過，個別員工和團隊都能在這兩種情境推送變更和功能，不會覺得工作效率受到影響。在第三種情境，我們可以依據變更不穩定度和工作倦怠情形，預測組織想要建立平台工程計畫來解決這些難題。

權衡利弊

雖然平台工程並非萬能，不過就整體的軟體開發和運作程序而言，這種做法有著強大潛力。但別忘了，任何做法都有利有弊。

就我們的研究來看，您在展開平台工程計畫時，可以多加留意兩大面向，藉此權衡利弊，讓您的組織享有平台工程帶來的效益，同時監控及管理潛在的負面影響。

首先，打造平台功能時，請著重於讓開發人員能夠獨立工作，並提供自助式服務機制。此時，請注意要兼顧平衡，以免因為規定只能使用該平台來處理應用程式生命週期中的所有層面，而阻礙開發人員獨立工作。

理想上，優質的平台應該讓使用者學會如何不仰賴平台中的工具和自動化機制，進而培養獨立工作的能力。不過，這麼做可能會提高複雜性。要減輕因而產生的負面影響，您可以成立專屬平台團隊，由他們積極與平台使用者合作並收集意見回饋。

平台計畫中的合作和意見回饋，可提高以使用者為中心的程度，促進平台發揮長期成效。資料顯示，收集意見回饋的方式不止一種，不妨採用多種做法，盡可能瞭解使用者的想法。

第二，請仔細監控應用程式變更的穩定度。如果穩定度降低，請確認是不是故意為之。提供平台就有機會針對不穩定度進行實驗、增進工作效率，並大規模提高表現。

不過，不穩定度也有可能造成工作倦怠。因此，在實行平台工程計畫期間，您必須仔細監控這類情況及查明成因，並請務必瞭解您對不穩定度的忍受度。您可以運用網站可靠性工程 (SRE) 中的服務水準目標 (SLO) 和錯誤預算，衡量自己的風險承受能力和平台成效，以便安全地進行實驗。

內部開發人員平台非常重視開發人員的體驗，不過要有效地推送並執行軟體，還需要許多團隊的協助，包括資料庫管理員、資安人員和運作團隊。

實行平台工程計畫時，除了要在各個團隊形塑「以使用者為中心」和「持續改良」的文化，也要確保行動符合組織目標。

這樣一來，平台的功能、服務和 API 就能充分滿足個別員工和團隊的需求，協助他們推送軟體及創造業務價值。



- ¹ Matthew Skelton 與 Manuel Pais, 2019 年。《Team Topologies: Organizing Business and Technology Teams for Fast Flow》(團隊拓樸：有效組建業務與技術團隊，實現高速工作流程)，IT Revolution Press：<https://teamtopologies.com/>
- ² <https://cloud.google.com/blog/products/application-development/richard-seroter-on-shifting-down-vs-shifting-left>
- ³ <https://dora.dev/capabilities/continuous-integration/>
- ⁴ <https://dora.dev/capabilities/test-automation/>
- ⁵ <https://dora.dev/research/2023/> 和 <https://dora.dev/research/2016/>
- ⁶ <https://dora.dev/research/2024/questions/#platform-engineering>

開發人員體驗



重點摘要

軟體不會憑空生成，即便有 AI 輔助也需要人工作業，而他們的工作經驗是組織邁向成功的基石。

今年的報告再度發現，只要開發人員建構的內容契合使用者的需求，員工和組織的表現就會提高。如果開發人員抱持以使用者為中心的心態來建構軟體，就會更有效率、較不容易工作倦怠，而且更有可能建構出優質產品。

軟體終究是為了服務使用者而生，因此組織有責任建立合適的環境，讓開發人員專心建構能提升使用者體驗的軟體。我們也發現如果環境穩定（意即優先要務不常變動），員工的工作效率就會微幅提高，工作倦怠情形則會降低，這點至關重要（兩者皆有統計顯著性）。

環境因素會大幅影響產品的開發品質，以及產品開發人員的整體工作體驗。

以使用者為優先，幾乎所有成果都會水到渠成

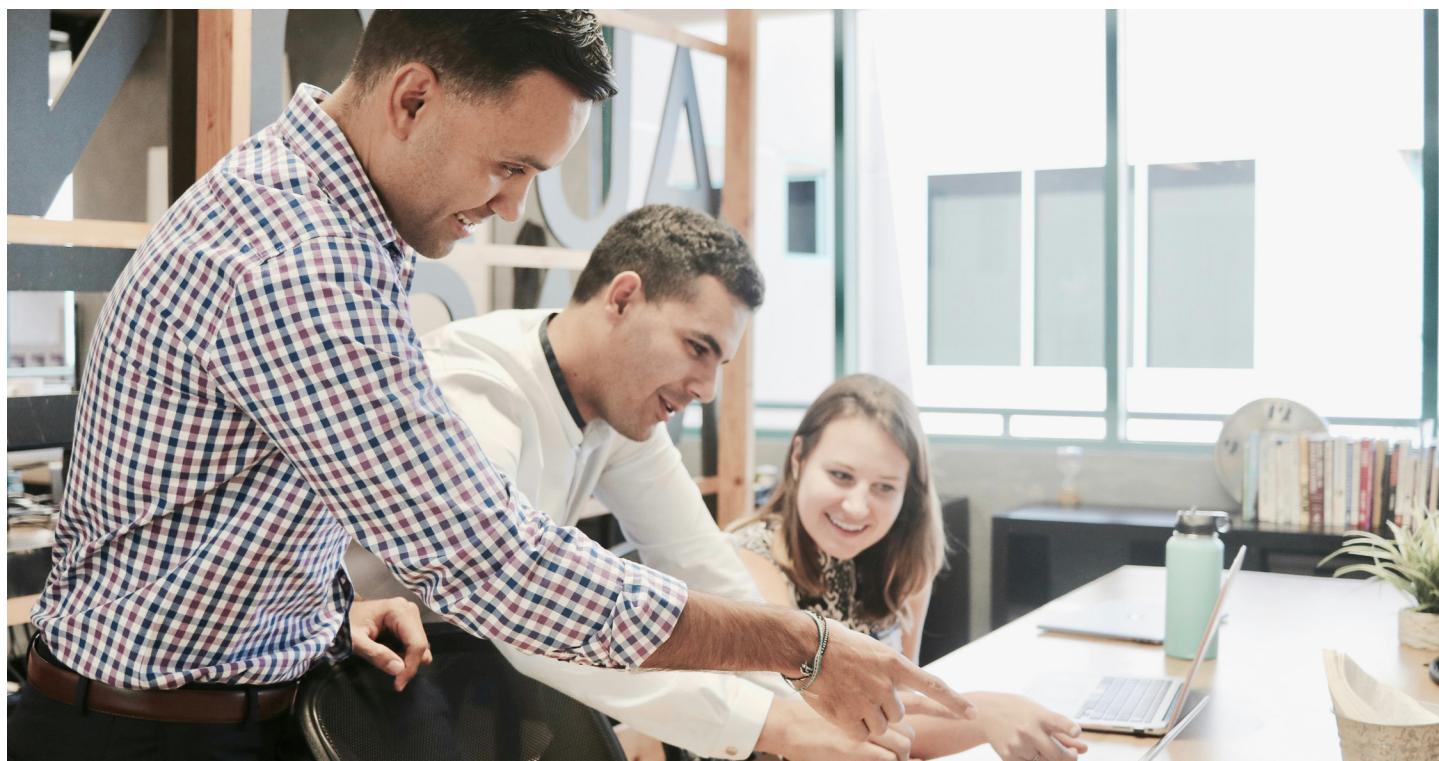
開發人員的工作很有意思，他們站在技術進展的最前線，形塑人類生活、工作、與世界互動的方式。

開發人員打造軟體和應用程式供人們使用，工作內容本質上與人息息相關。他們的工作環境往往講求功能與創新，不過較少強調要瞭解這些功能是否為產品使用者提供價值。

我們的研究提供了有力證據，顯示採用以使用者為優先的軟體開發做法，同樣會對員工和組織帶來正面影響。

今年的問題主要想瞭解開發人員是否具備下列特質：

1. 會參考使用者的意見回饋，回頭修改功能或調整優先順序
2. 知道使用者想透過某項應用程式/服務達成的目標
3. 相信「以使用者為重」是業務成功與否的關鍵
4. 認為使用者體驗是第一要務





調查結果及其意義

資料清楚指出，懂得注意使用者需求和痛點的組織，能打造出更優質的產品。

我們發現，注重使用者的文化可以提高工作效率和滿意度，同時降工作低倦怠的風險。

更重要的是，這些效益會從個別員工擴及組織。在前幾年的研究，我們提到表現出色的組織可以快速可靠地推送軟體。也就是說，軟體推送成效是成功的必要條件。

不過，資料顯示另一個做法也能帶領組織邁向成功：

組織本身、開發人員及其雇主可以建立以使用者為中心的軟體開發做法。

我們發現，組織如果能掌握並理解使用者需求，那麼軟體推送作業的穩定度和處理量，未必是達成產品品質的條件。優先追求使用者體驗，品質自然會高。

如果組織不重視使用者、未在開發時參考使用者的意見回饋，就必須加倍追求推送作業的穩定度和速度，才能提高產品品質（詳見圖 15）。

我們瞭解某些組織可能會傾向於專心打造功能及翻新技術。從表面來看，這種做法沒有問題，畢竟開發人員肯定比一般使用者更瞭解技術細節。

不過，如未實際瞭解使用者體驗，而是依據猜測開發軟體，開發人員較有可能建構出看似光鮮亮麗，實則乏人問津的功能¹⁰。

如果組織和員工瞭解使用者與世界互動的方式，就較有可能建構出可以滿足實際需求的功能，進而提高這些功能的使用率。

專注於滿足使用者需求，就能打造出出色的产品。

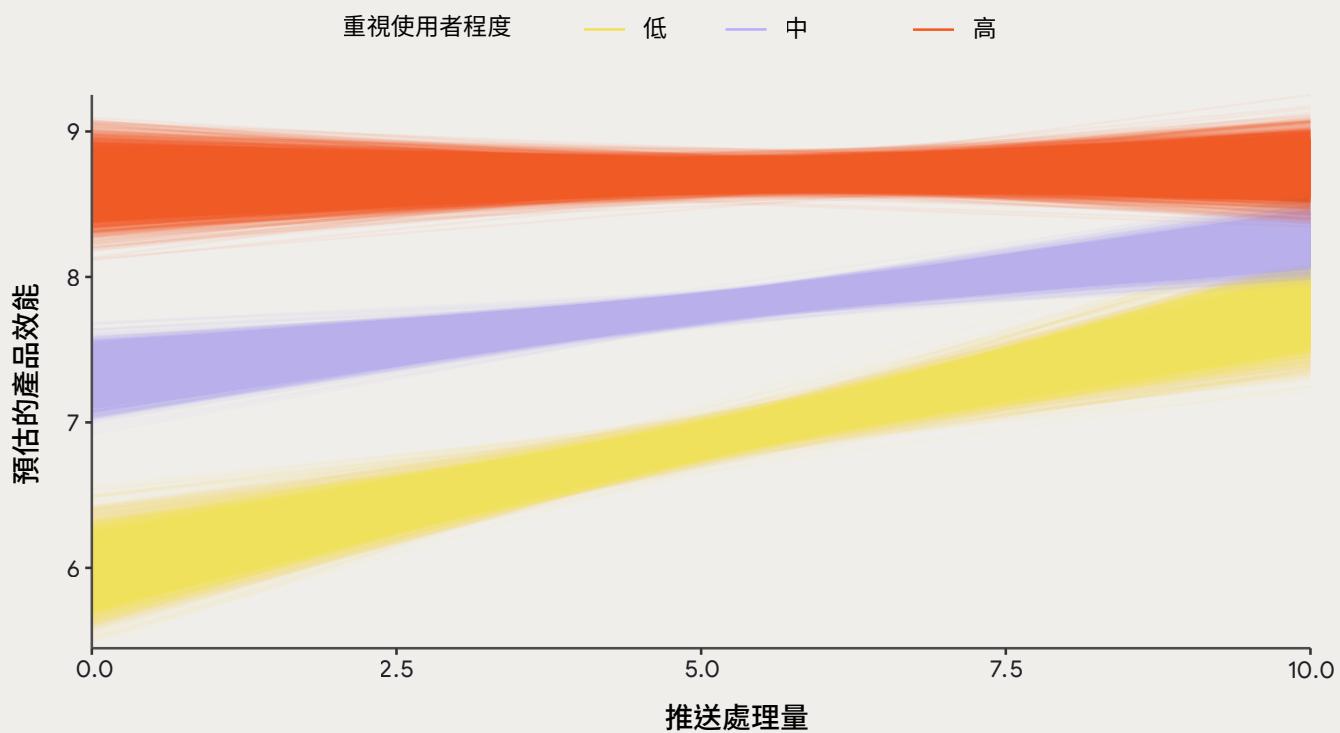


圖 15:3 種以使用者為中心的程度與產品效能和推送處理量的關係

為何以使用者為中心的軟體開發理念與做法這麼受歡迎？

學術研究指出，如果知道工作的目的，員工和組織的表現都會提高^{2,3}。

舉例來說，日前的一項問卷調查顯示，93%的工作者認為知道自己的工作有何意義十分重要⁴。類似的情況是另一項問卷調查發現，如果能從事長久有意義的工作，受訪者平均願意減少 23% 的未來收入⁵。

員工願意取捨的程度讓人驚訝，但也說明了他們工作的動力來源，可看出大家想將時間投注於重要的事。

「如果每個人都能透過工作，為公司以外的個人或當地社群帶來正面影響，那一定很棒。但不是大家都這麼幸運，有時甚至不可能辦到。舉例來說，自動駕駛技術的目標非常宏大，希望駕駛就算在高速公路也能安然入睡。這不是我想做的事，我想幫助不會開車的人去想去的地方、做想做的事。」

(訪談對象 2)⁶

指明方向：

以使用者為中心進行軟體開發，可以從根本改變開發人員對工作的看法。開發人員不會漫無目的推送功能，也不需要猜測使用者會不會使用，而是可以依據使用者的意見回饋排定建構優先順序。

採取這種做法後，開發人員會相信自己處理的功能有存在的理由，手上的工作也會突然產生意義：他們需要確保產品和服務帶給使用者出色的體驗，開發出的軟體與真實世界再也不會毫無關聯。

開發人員可以透過自己打造的軟體，直接瞭解辛勤工作產生的影響。

「我們公司面臨的推送壓力很大，有很多夢幻的理想和研擬改良方法的論點，譬如最近就調整了組織結構。這些都是為了推送內容，而不是因為關注品質。我自己並不喜歡這種做法。」(訪談對象 9)

增進跨職能合作：

即便是最有才華的開發人員也無法獨自建構軟體。由不同員工多方合作，才能打造出優質產品。大家的專長往往不一樣，卻能相輔相成。

採取以使用者為中心的開發做法後，開發人員將能在整個組織進行跨職能合作。此時，他們的職責就不只是推送軟體那麼單純，而是以團隊為單位，為使用者創造出色體驗。

這種軟體開發做法可以打破開發人員之間的隔閡、促進共識、推動團隊合作，並帶來相互學習的機會。解決問題的方式也有所不同，開發人員除了要解決技術問題，更要懂得在過程中盡可能滿足使用者的需求。

這種做法可以提高員工參與度，並創造更能激發創意的環境，減少進度停滯帶來的挫折感，進而避免連帶的工作倦怠情形。

組織可以採取什麼行動？

依據研究結果，我們建議組織投注時間和資源來認識使用者，瞭解誰會使用您建構的內容，以及他們面臨的難題。我們認為這項投資絕對值得。

請避免憑空猜測，您可以在日常生活觀察使用者、提問，並虛心接受他們的意見回饋來調整方向。這樣一來，開發人員的工作效率就會提高、出現工作倦怠的可能性降低，同時提供更優質的產品。

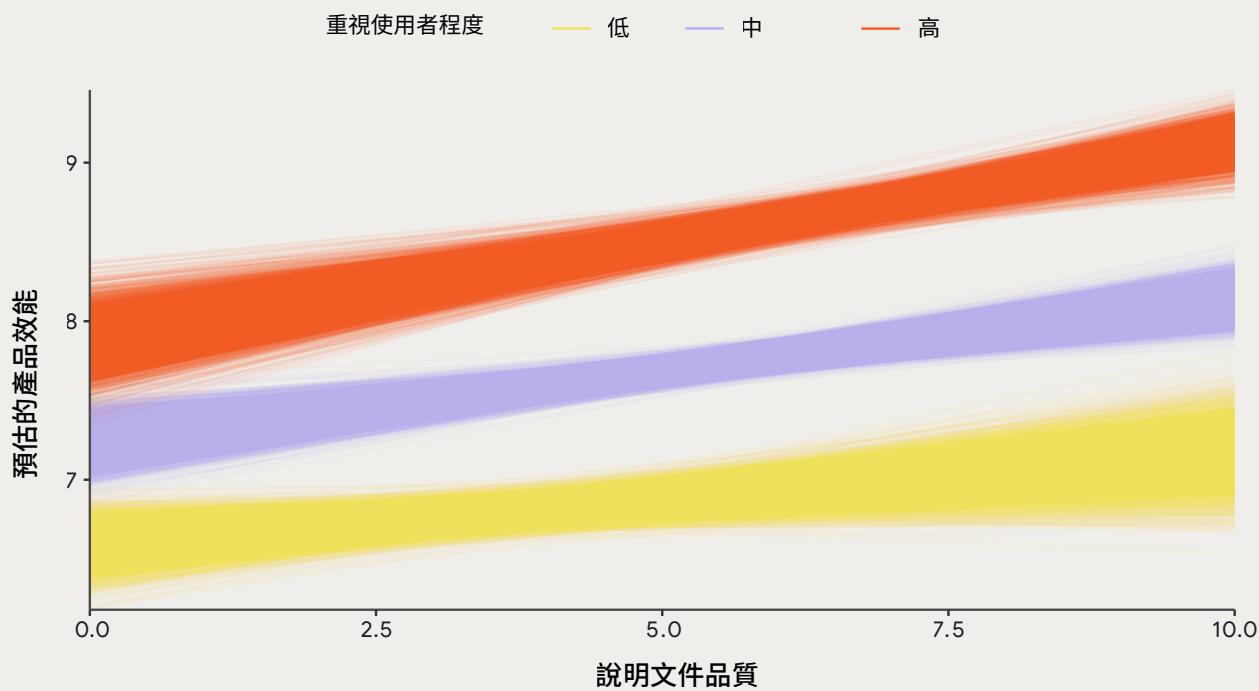
以使用者為中心開發軟體並搭配有效說明文件，成果再加倍

重視使用者的團隊可以打造出效能更卓越的產品。如果搭配優質的內部說明文件，產品效能還可進一步提升（詳見圖 16）。這項發現與過往的研究結果相似：說明文件可以放大技術功能對組織表現的影響⁷。

說明文件可讓整個團隊瞭解使用者信號與意見回饋，進而改良產品本身。

我們發現如果缺少使用者信號，內部說明文件對預估產品效能的影響就不明顯。不過，如果團隊擁有優質的內部說明文件，當中的使用者信號將可對產品效能產生更大的影響。

我們自 2021 年起關注這部分，而後每年都會觀察到優質說明文件的廣大影響。今年的研究結果列出了內部說明文件對預估產品效能的影響。



圖形由 12,000 條線構成，反映了我們模擬預測最可信的模式
圖 16:3 種以使用者為中心的程度與產品效能和說明文件品質的關係

編寫說明文件的文化

敏捷軟體開發宣言提倡「可用的軟體重於詳盡的文件」⁷。不過，我們持續發現，優質說明文件是軟體正常運作的關鍵。

「詳盡的文件」這個說法或許代表不健康的做法，可能包含編寫說明文件本身。有問題的說明文件編寫做法包括：只是為了應付行政規範而製作文件，或是為了掩蓋管理階層與員工互不信賴。另外某些企業可能只編寫說明文件，但不維護或統合，這樣也不健康。

在上述情況下，我們計算出的說明文件品質分數可能偏低。這類內容的目標讀者不正確，因此在工作時參考的效果較差。此外，說明文件太多或太少，也都會讓人很頭痛。

衡量說明文件品質時，我們採用尋獲度和可靠性等指標。別忘了，內部說明文件的主要讀者是自家同事，您日後要完成特定工作時也有可能參考⁸。團隊必須滿足這些讀者的需求，才能營造良好的說明文件編寫文化。這也印證了「重視使用者」的重要性。

您可以按照下列做法製作優質的說明文件，在所屬團隊建立良好的編寫文化：

記錄重要用途。

參加技術寫作訓練。

定義說明文件更新工作的權責歸屬和程序。

將說明文件編寫工作分派給團隊成員。

將維護說明文件視為軟體開發生命週期的一環。

刪除過時或重複的說明文件。

在績效考核和升遷程序中，肯定員工對說明文件編寫工作的貢獻。

優先要務不斷變動的壞處

您應該對下列情境有同感：您在過去幾個月埋首建構一項新功能，覺得幹勁十足，因為這項功能可以幫助到使用者。突然之間（或表面上如此），主管團隊決定改變組織的優先要務。您還不清楚手上的專案會不會暫停、作廢、與其他專案整併，或是徹底改變。

這種情況如果經常發生，會對員工和組織造成很大的影響。一起來看看組織不斷改變優先要務會發生什麼事。

調查結果及其意義

整體來說，我們發現如果組織的優先要務時常改變，工作效率會微幅下降（有統計顯著性），工作倦怠情形則會大幅增加。

資料顯示，員工因而感到工作倦怠後，情況會難以改善。我們想瞭解強大的主管階層、出色的內部說明文件，還有以使用者為中心的軟體開發做法，是否能減少改變優先要務引發的工作倦怠情形。

答案是沒有辦法。即便組織具備上述正面特性，只要優先要務時常改變，員工還是有可能工作倦怠。

為何組織優先要務時常改變會影響員工的身心健康？

我們推測如果組織時常改變優先要務，員工就會缺乏明確的預期、控制感降低，工作負擔也會加重，進而增加工作倦怠的情形。

有一點要澄清，「改變優先要務」本身沒有問題。業務目標和產品方向本來就會經常調整，因此組織可以彈性調整優先要務其實是好事。

我們認為優先要務的改變頻率才是損害員工身心健康的原因。依據優先要務時常改變產生的不確定性，我們可以推測優先要務改變頻率會有長期的影響。

數十年來的學術研究證實，長期壓力對身心健康會有負面影響⁹，這些研究與我們的調查結果相呼應。長期不穩定的環境會讓不確定性升高，並導致員工的控制感降低，這兩項因素結合就很容易引發工作倦怠。

優先要務穩定後會如何？

我們發現了讓人略微困惑的現象：優先要務穩定下來之後，軟體推送成效反而降低，推送速度放慢且較不穩定。

我們推測原因可能是如果組織的優先要務穩定，他們的產品和服務通常相當良好，因此變更的頻率較低，或是可以推送比建議批量更多的變更，因此推送頻率較低。

儘管如此，我們仍然十分訝異。您覺得為何組織優先要務穩定下來，軟體推送速度和穩定度反而降低呢？

為使用者建構 AI 服務可以防止優先要務時常改變，但無法提高推送穩定度

為使用者提供 AI 輔助服務可讓組織優先要務穩定下來。這句話就像是俗爛的 AI 技術宣傳標語，但是我們不認為這項發現與 AI 本身有關。

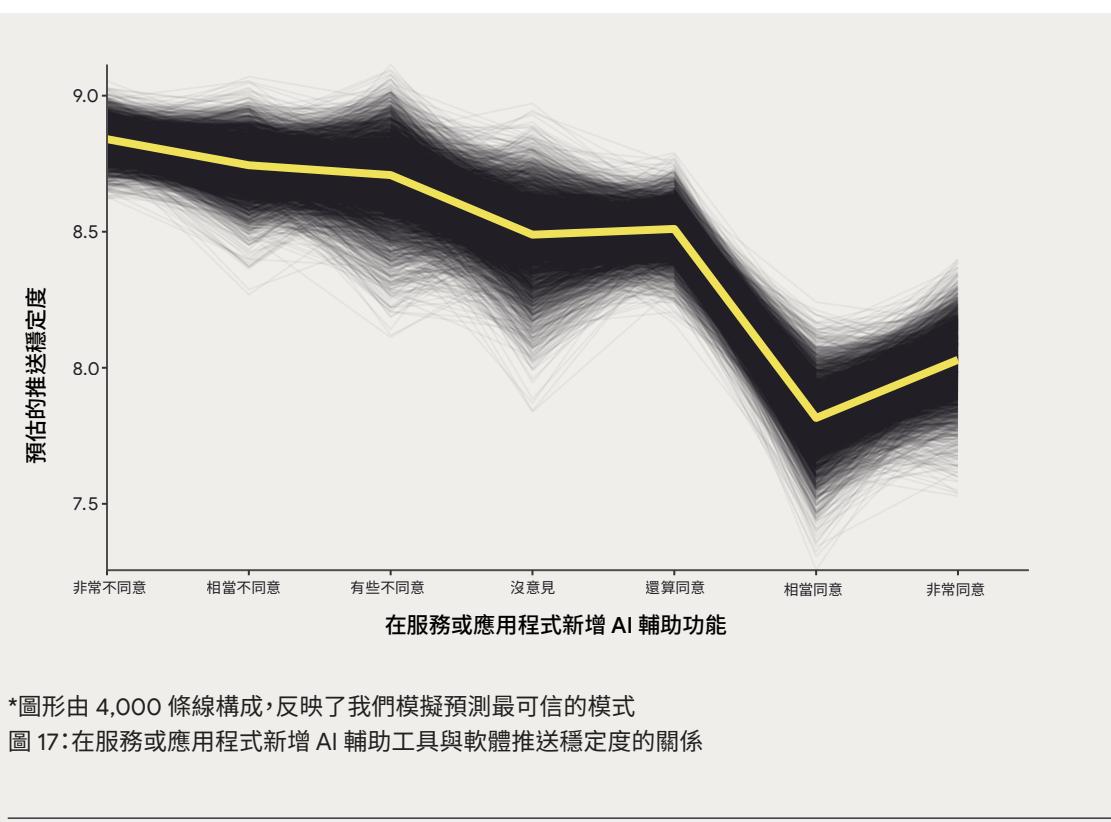
反之，我們推測如果將重心改為建構 AI 服務，方向就會明確，組織也能朝著核心目標前進。比起 AI，這個明確的方向才是組織優先要務穩定下來的原因。

這點值得特別強調，因為這讓我們瞭解新技術問世時，組織會發生什麼情況。新技術帶來改變，組織需要時間適應。在這段期間，主管階層可能會想找出對組織最有利的做法，

因此優先要務會改變。塵埃落定之後，組織就能擬定明確的後續行動，優先要務則會開始穩定下來。

不過，即便優先要務不再改變，軟體推送程序也不會馬上穩定下來。分析結果顯示，改為在服務或應用程式新增 AI 輔助功能之後，您會面臨難題並歷經陣痛期。

相較於沒有改變的團隊，方針改變的團隊在軟體推送穩定度方面降低了 10% 之多。關於這個難題，請見下方圖表瞭解詳情。



*圖形由 4,000 條線構成，反映了我們模擬預測最可信的模式

圖 17：在服務或應用程式新增 AI 輔助工具與軟體推送穩定度的關係

組織可以採取什麼行動？

答案很簡單，但不一定容易執行。依據研究結果，我們建議組織盡量不要改變優先要務，這樣才能確保員工不會因為優先要務時常改變而受到影響，引發工作倦怠。

優先要務時常改變會帶來哪些負面影響？

研究結果顯示，組織可能會難以擁有出色的管理階層、優質的說明文件，還有以使用者為中心的軟體開發做法。因此我們相信，要想防止工作倦怠，除了創造穩定感之外，組織能做的並不多，但可以盡量(1)避免時常改變優先要務；(2)即使需要經常變更，也不能影響員工的日常工作。

1. <https://www.nngroup.com/articles/bridging-the-designer-user-gap/>
2. <https://executiveeducation.wharton.upenn.edu/thought-leadership/wharton-at-work/2024/03/creating-meaning-at-work/>
3. <https://www.apa.org/pubs/reports/work-in-america/2023-workplace-health-well-being>
4. <https://bigthink.com/the-present/harvard-business-review-americans-meaningful-work/>
5. <https://hbr.org/2018/11/9-out-of-10-people-are-willing-to-earn-less-money-to-do-more-meaningful-work>
6. 訪談對象 [數字] (例如「訪談對象 1」) 用來代稱訪談對象。
7. <https://cloud.google.com/blog/products/devops-sre/deep-dive-into-2022-state-of-devops-report-on-documentation> 與《加速發展：2023 年開發運作現狀》- <https://dora.dev/research/2023/dora-report>
8. <https://agilemanifesto.org/>
9. 有其他讀者，包括管理階層、監管單位或稽核機構。
10. Cohen S、Janicki-Deverts D、Miller GE,《Psychological Stress and Disease》(心理壓力與疾病),JAMA。2007;298(14):1685–1687.
doi:10.1001/jama.298.14.1685

引領轉型

想要順利轉型，必須先滿足多項條件。今年的研究發現，表現出色的團隊具備多項特質，包括重視穩定度、以使用者為重、擁有優秀的主管階層，並會編寫優質的說明文件。這一節會分享一些實用做法，協助您擬定成功的轉型計畫。

我們發現成功轉型的關鍵在於抱持「持續進步」的心態。研究中，表現出色的團隊會瞭解受阻的原因，並將 DORA 做為基準指標，按部就班持續進步。雖然各項做法都需要相當出色，才能取得長期成功，但過去十年的 DORA 研究已歸結出四種方法，可帶來深遠的影響，協助您開始在組織內推動轉型。



轉型領導力

轉型領導力模式中，主管階層會賦予員工價值和目標感，激發出員工更出色的表現，推動大規模的組織變革。

這些主管會從下列層面，鼓勵團隊成員朝共同的目標邁進：

願景

對團隊和組織的未來有明確的願景。

鼓勵式溝通

稱讚團隊；讓員工以身為組織的一分子為榮；以及鼓勵員工將改變視為契機。

激發創意

邀請團隊成員從新的角度檢視舊有問題，以及重新思考自己對工作做出的一些基本假設。

支持型領導力

行動前會考量其他員工的個人感受；做事周全，會顧及其他員工的個人需求。

個人肯定

團隊成員表現優異時給予稱讚；工作表現進步時給予肯定。

今年的研究結果指出，轉型領導力可以大幅提高員工的工作效率。轉型領導力提高 25% 時，員工的工作效率增加了 9%。

轉型領導力的助益並非只有提高工作效率，擁有優秀的主管也能帶來下列好處：

- 員工工作倦怠情形減少
- 工作滿意度提高
- 團隊表現提高
- 產品效能提高
- 組織表現提高

2017 年時，我們發現上述領導階層特質與 IT 成效有顯著的統計關聯性。以表現出色的團隊來說，他們的主管在全部五項特質的分數都很高，反觀表現不佳的團隊，他們的主管則是分數墊底。另外，我們觀察到轉型領導力與員工淨推薦值（簡稱 eNPS）高度相關。eNPS 代表員工推薦在所屬公司任職的可能性。

不過，單憑轉型領導力並不會帶來出色的表現，其作用比較像是催化劑。

對於想要採用技術和產品管理功能與做法的團隊，轉型領導力相當關鍵，因為這有助於(1) 分派權責及賦予團隊自主性；(2) 提供所需的指標和商業智慧，以便解決問題；以及(3) 建立獎勵制度，表揚團隊為使用者提供價值，而非只關注功能推送。

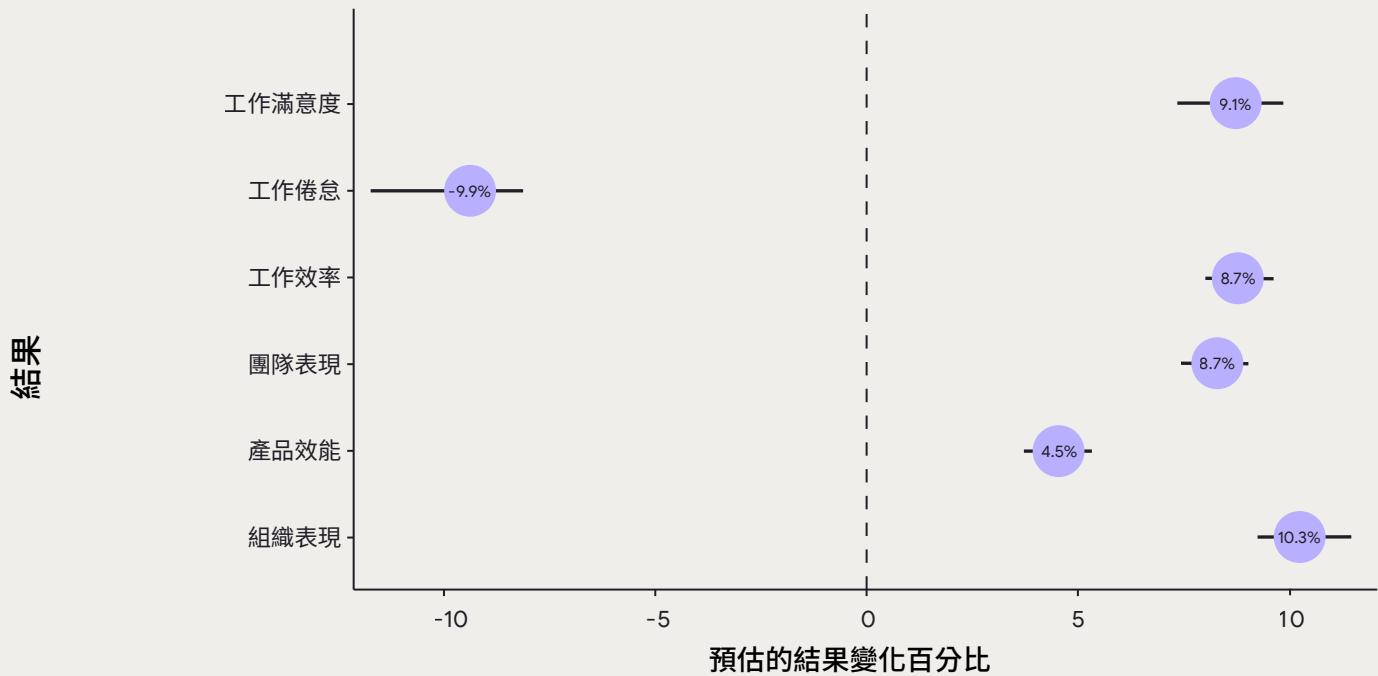
轉型需要時間和工具，主管階層也必須分配專屬資源給改良工作。出色的主管扮演了重要的角色，他們需要為團隊提供必要的時間與資金來增進表現，而且不應要求工程師在下班時間學習新知及自動處理工作，應將這些事項排入他們的工作內容。

研究結果推翻了過往的論點，證明 IT 並非成本中心，而是可以取得業務成功的投資標的。我們在 2020 年發布了有關 DevOps 投資報酬率的白皮書²，內含多項算式，可協助您說明投資 IT 改良計畫的潛在價值。

金錢方面的回報只是這項投資的其中一項效益。我們在 2015 年進行的研究指出，「組織在 DevOps 方面的投資與多項因素高度相關：1) 組織文化；2) 開發、運作和資安團隊

達成雙贏局面的能力；3) 較少的工作倦怠情形；4) 較有成效的領導力；以及 5) 有效的持續推送軟體更新與精實管理做法」³。因此，我們推薦分派一定的專屬資源給改良計畫。

如果轉型領導力提高 25%...



圓點 = 預估值

誤差線 = 89% 的不確定性區間

圖 18：轉型領導力對不同結果的影響。

持續以使用者為中心

今年的研究顯示，如果組織擁有出色的主管，並致力於建構可滿足使用者需求的軟體，就能享有強大效益，進而開發出更優質的產品。只要開發軟體時以使用者為中心，主管階層就能傳達明確的願景。

終極目標是打造出使用者喜愛的產品。如同《開發人員體驗》一章中所述，「以使用者為重」才是產品功能的意義所在。如果開發人員知道他們可以增進使用者的體驗，就能放心建構這些功能。

我們發現如果團隊非常想瞭解及滿足使用者的需求，同時會收集、追蹤使用者意見回饋並採取相應措施，他們所屬的組織可以發揮最高水準的表現。事實上，即便組織的軟體開發速度和穩定度不高，只要他們重視使用者，同樣能取得成功。2023 年的研究結果顯示，相較於不重視使用者的團隊，以使用者為中心的團隊在組織表現方面高出 40%⁴；在 2016 年，我們也發現以使用者為中心的團隊可以帶來較高的組織表現。

今年的研究印證了過往的結果，重視使用者的團隊可以提供更優質的產品。

除了產品品質提高之外，員工對工作的滿意度也更高，較不容易工作倦怠。

有了快速、穩定的軟體推送體驗，組織就能獲得更多試驗及學習的機會。理想情況下，這些實驗與疊代會以使用者的意見回饋為基礎。只要您能夠快速穩定地推送軟體，就能進行實驗、充分瞭解使用者的需求，並迅速調整他們不滿意的部分。

如果享有穩定快速的推送體驗，您也能輕鬆因應市場變化或競爭。

別忘了，內部開發人員也是使用者。如果您的組織設有內部開發人員平台 (IDP)，就能為開發人員提供價值，進而由他們為外部使用者或其他內部使用者創造價值。

我們的研究顯示，必須抱持以使用者為中心的心態，並秉持開發產品的思維，才能打造出成效卓越的 IDP，讓開發人員能夠獨立工作。只要按照上述方式部署，IDP 就能提高個別員工的工作效率、團隊工作效率和組織表現。

成為依據資料行事的組織

能夠具體呈現進展的能力至關重要。過去 10 年來，我們提供清楚的證據，證明組織依據資料行事很重要。DORA 的四項關鍵指標⁵已成為衡量軟體推送成效的全球標準，不過 DORA 的價值不只如此，我們找出了超過 30 項可以提高組織表現的能力與程序⁶。

指標的價值在於幫助我們瞭解是否有進步。建議您將四項關鍵指標用於應用程式和服務，而非組織或業務範圍。指標應該做為具體呈現所做努力的方式，讓您瞭解持續改良的過程中進步多少，而非用來比較不同團隊，也絕非用於比較不同員工。

指標也不適合用來評估應用程式或服務團隊的成熟度。低、中、高或優異的分類或許有趣，不過轉型過程中，這些名詞的意義不大，參考時請謹慎小心。

隨著研究持續發展及演進，我們會鼓勵您不要侷限於這四項關鍵指標。如今研究結果已相當明確，使用者意見回饋指標與四項關鍵

指標同樣重要。我們認為這是因為多數團隊已打造出能正常運作的解決方案，進而提高速度與穩定度。因此，如果各方面的表現都提高了，速度和穩定度帶來的效益就會減少。

如要從整體的角度看待轉型，建議您建立資訊主頁及製作圖表，同時掌握技術指標（例如 DORA 的四項關鍵指標和可靠性指標）和業務指標，這樣即可串接由上而下和由下而上的轉型工作，以免出現落差。這也有助於將您的核心目標、OKR 與員工目標連結至 IT 投資，以便計算投資報酬率。

我們相信指標是出色表現的必要條件，也是制定決策時的重要依據。收集越多的量化和質性指標，就越能制定明智適當的決策。人們對資料的價值或意義永遠不缺意見，不過依據資料制定決策往往比仰賴意見或直覺更妥當。



全面採用雲端或繼續留在資料中心

2018 年起，我們一直研究 NIST 定義的雲端運算 5 項特性⁷與組織表現之間的關係。前述的 5 項特性是指隨需自助式服務、多元的網路存取方式、資源集區、靈活彈性和測量的服務，也稱為「有彈性的基礎架構」。我們發現相較於表現不佳的團隊，表現出色的團隊更有可能運用有彈性的基礎架構。

去年的研究呈現了一個至今最讓我們震撼的結果：如果只使用雲端，但未善用五項特性，可能會帶來負面影響，組織表現也有可能降低。如果組織不想徹底改造應用程式或服務，留在資料中心可能是較理想的做法。當然，要做到這件事不只需要採用工具或技術，經常也需要全新的應用程式設計、建構、部署及執行模式。如要推動大規模改變，建議您從少數服務著手，並採取疊代做法，幫助團隊和組織邊學習邊成長。

總結

過去 10 年的研究持續發現轉型是成功的必要條件。許多組織誤解了一件事：轉型並非目的地，而是一段持續改良的旅程⁸。我們的研究指出，沒有持續進步的公司其實就是落於人後。相反地，如果公司秉持不斷進步的思維，就會取得最亮眼的成果。

提醒您，旅程中可能會有一些難關和痛苦。我們的研究發現，採用 DevOps⁹、SRE¹⁰ 和今年的平台工程做法之後，成效一開始會先降低，而後大幅提高（也稱為「J 曲線」）。這是正常現象，只要您持續進步，情況就會好轉，最終會比一開始更出色。

想到轉型之路沒有盡頭可能會讓人卻步，如果一心想規劃、設計出完美的轉型計畫，會很容易卡關。其實，成功的關鍵在於開始行動，組織和團隊只要以「超越昨天一點點」為目標就已足夠。我們過去 10 年和未來的研究，目標都是幫助您更輕鬆地轉型。

-
1. 《Dimensions of transformational leadership: Conceptual and empirical extensions》(轉型領導力的各個面向：理論與實證應用) - Rafferty, A. E. 與 Griffin, M. A.
 2. 《DevOps 轉型的投資效益》- <https://dora.dev/research/2020/>
 3. 《2015 年開發運作現狀報告》- <https://dora.dev/research/2015/2015-state-of-devops-report.pdf#page=25>
 4. 《加速發展：2023 年開發運作現狀報告》- <https://dora.dev/research/2023/dora-report/2023-dora-accelerate-state-of-devops-report.pdf#page=17>
 5. DORA 的四項關鍵指標：<https://dora.dev/guides/dora-metrics-four-keys/>
 6. DORA 推薦的能力與程序：<https://dora.dev/capabilities/>
 7. NIST 定義的雲端運算 5 項特性：<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
 8. 持續進步的旅程：<https://cloud.google.com/transform/moving-shields-into-position-organizing-security-for-digital-transformation>
 9. 《加速發展：2018 年開發運作現狀報告》- <https://dora.dev/research/2018/dora-report/>
 10. 《加速發展：2022 年開發運作現狀報告》- <https://dora.dev/research/2022/dora-report/>

DORA 研究 十年沿革



歷史回顧

在 2009 年，兩場主題相關但除此之外沒有關聯的活動，催生了 DevOps 運動。該年 6 月，John Allspaw 和 Paul Hammond 在 Velocity 大會演講。大會名為「10 deploys per day: Dev & ops cooperation at Flickr」（一天部署 10 個項目：Flickr 的開發和運作協作）¹。幾個月之後，Patrick Debois 帶領一群志工在比利時根特舉辦了第一場 DevOpsDays 活動²。

接著，DevOps 社群就迫不及待想瞭解這方面的演進。任職於 Puppet Labs 的 Alana Brown 在 2011 年進行了一項問卷調查，以便進一步瞭解 DevOps。這項問卷調查證實「DevOps 工作型態已成為 IT 產業的新興業務模式」。

隨著 DevOps 運動持續拓展至更多產業和組織，Alana 憑藉這次的成功經驗，於 2012 年和 IT Revolution Press 合作進行另一項問卷調查，並在《2013 年開發運作現狀報告》發表研究結果³。

Nicole Forsgren 博士在隔年加入研究團隊，為這項計畫提供更多科學基礎。《2014 年開發運作現狀報告》⁴ 證實了軟體推送成效與組織表現之間的關聯性，研究結果指出「上市公司若擁有表現出色的 IT 團隊，三年間的市值成長比 IT 團隊表現不佳者高出 50%」。

2016 年時，每年發布報告的趨勢確立，Forsgren、Jez Humble 和 Gene Kim 於同年創立了 DevOps 研究與評估 (DORA) 計畫。為協助採用 DevOps 做法的團隊衡量投資效益，該年度的《開發運作現狀報告》增加了算式。發布於 2020 年的《DevOps 轉型的投資報酬率》⁵白皮書詳細說明了這項成果。

《Accelerate: The science behind devops: Building and scaling high performing technology organizations》(ACCELERATE：精益軟體與 DevOps 背後的科學)⁶ 由 Forsgren、Humble 和 Kim 共同撰寫而成，於 2017 年由 IT Revolution Press 出版。這本書總結了 DORA 研究計畫的早期成果，並著重於可以提高效能的能力。

DORA 公司在 2018 年發布了一份獨立報告《Accelerate State of DevOps: Strategies for a New Economy》(加速發展：開發運作現狀與新經濟策略)⁷。Puppet Labs 團隊同年與 DORA 分道揚鑣，繼續發布自己的報告系列⁸。

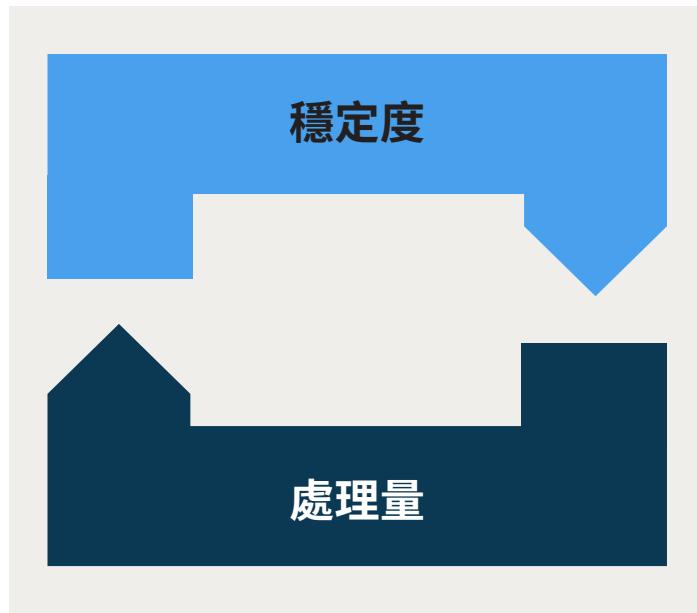
2018 年末，Google Cloud 收購了 DORA⁹，繼續進行這項各平台通用的科學研究。DORA 報告今年正式邁入十週年¹⁰。很高興能分享我們的研究成果，也感謝您的支持！

DORA 的關鍵洞察

團隊不必犧牲速度來換取穩定度

技術導向團隊需要制定方法來衡量成效，以便評估目前的情況、排定改善項目的優先順序，以及確認進展。DORA 已找出並檢核四項關鍵的軟體推送成效指標，讓團隊有效地衡量自己的軟體推送程序成果。這些軟體推送成效衡量方法已成為業界標準。

研究顯示變更處理量和穩定度具有連動性，各行各業都有團隊在這兩項指標表現優異。



團隊可以透過多種方式衡量四項關鍵指標，包括：

- 在團隊會議期間進行討論與反思
- 進行 DORA 快速檢驗
(<https://dora.dev/quickcheck>)
- 使用軟體工程智慧 (SEI) 類別中的商業工具和可取得原始碼的工具¹¹
- 針對團隊使用的工具量身打造整合功能

軟體推送和運作成效對組織表現有正面影響

DORA 透過四項關鍵指標衡量軟體推送成效，並於 2018 年首度研究運作成效。這項指標可以衡量組織對軟體產品或服務做出的承諾和聲明，以及是否有能力履行。

我們發現，必須同時具備出色的軟體推送和運作成效才能取得最佳結果，增進組織表現和員工身心健康。

技術導向團隊中的從業人員瞭解在推送程序減少阻礙的重要性，也會滿足應用程式使用者對可靠性的期望。

成效

軟體推送 四項關鍵指標

可靠性 服務水準目標 (SLO)



會影響



結果

組織表現

身心健康



文化是取得成功的首要條件

組織文化是工作表現的明確指標。充滿信任的文化可以帶動學習及合作的風氣，效果相當強大，這在過往的研究屢見不鮮。舉例來說，2022 年的研究中，我們發現文化是最大的預測指標，可以用來推估組織的應用程式開發安全性做法¹²。

文化具有許多面向，且會不斷變動，影響研究的各個層面。過去幾年來，我們參考其他研究（例如 Westrum 的組織文化分類學）¹³，以不同的指標衡量文化。舉例來說，衡量身心健康時，我們採用工作倦怠、工作效率和工作滿意度等指標。

轉型更輕鬆

我們會建議團隊設定成長目標，讓自己更輕鬆地轉型。如要提高成效，就需要「持續進步」的心態和做法，包括評估現今的狀況、安排改善工作的優先順序，以及導入意見回饋機制來衡量進展。

嘗試進步可能會成功，但失敗也在所難免，無論結果如何，團隊都能從中汲取經驗，並據此採取具有實效的行動。

展望未來十年

回首過去十年，我們從彼此身上學到了許多。感謝各位參與我們的年度問卷調查、加入 DORA Community of Practice¹⁴，以及在您的組織應用 DORA。

隨著技術環境不斷演進，DORA 會持續研究能力與做法，幫助技術導向團隊與組織邁向成功。我們還是會優先從人的層面探討技術，並致力於發布各平台通用的研究結果，幫助您推動轉型。

面對新興的技術和做法，許多過往的 DORA 洞察至今仍有參考價值，我們相當期待與您攜手取得新的洞察！

DORA 致力於研究 DevOps 運動中永遠存在的基礎原則，包括文化、合作、自動化、學習，以及採用技術來達成業務目標。不同職務的人員提供各式各樣的意見，滋養了 DORA 社群與研究，即使是與 DevOps 沒有直接關聯的受訪者，也很有幫助。不過，日後的研究重點可能會從 DevOps 慢慢轉移。

今年的報告花了不少篇幅探討人工智慧 (AI) 的使用與影響。如前文所述，AI 的採用率持續提高，這個領域有許多試驗空間。針對 AI 和其他新興技術與做法，DORA 會持續調查未來的影響。希望您結合過往研究與新的發現來提高 AI 的採用率，同時提升團隊所有成員的體驗。

1. 投影片：<https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>；影片：<https://www.youtube.com/watch?v=LdOe18KhtT4>
2. <https://legacy.devopsdays.org/events/2009-ghent/>
3. <https://www.puppet.com/resources/history-of-devops-reports#2013>
4. 《2014 年開發運作現狀報告》- <https://dora.dev/research/2014/>
5. 《DevOps 轉型的投資效益》- <https://dora.dev/research/2020/>
6. Forsgren, Nicole, Jez Humble 與 Gene Kim, 2018 年。《Accelerate: The Science Behind DevOps: Building and Scaling High Performing Technology Organizations》(ACCELERATE：精益軟體與 DevOps 背後的科學)。IT Revolution Press。
7. 《加速發展：開發運作現狀與新經濟策略》- <https://dora.dev/research/2018/dora-report/>
8. <https://www.puppet.com/resources/history-of-devops-reports#2018>
9. <https://dora.dev/news/dora-joins-google-cloud>
10. 雖然 DORA 的創立年份晚於 2014 年，但我們將 Forsgren 博士加入計畫的那年（2014 年）視為第一版 DORA 報告。2020 年未發布報告，因此 2024 年為第十版。
11. <https://dora.dev/resources/#source-available-tools>
12. 《加速發展：2022 年開發運作現狀報告》- <https://dora.dev/research/2022/dora-report/>
13. Ron Westrum,《A typology of organisation culture》(公司文化分類學)，BMJ Quality & Safety 13, no. 2(2004), doi:10.1136/qshc.2003.009522
14. <https://dora.community>

結論

過去十年來，DORA 已成為深受信賴的研究結果、洞察和資訊來源。隨著業界陸續採用平台工程和人工智慧等新做法和技術，DORA 會繼續陪伴您探索有助於增進團隊表現的工作方式，感謝您一路以來對 DORA 的信任！

重現研究結果

今年度報告的研究領域和結果相當複雜，有時出現較不明確，甚至相互衝突的發現。我們鼓勵您試著重現 DORA 的研究結果。挑選一個團隊或組織來實驗，就有機會深入瞭解結果。

在您的組織進行實驗

您可以將 DORA 的研究結果當做下次實驗的假設，藉此進一步瞭解團隊的運作情形，看看哪些區塊適合運用 DORA 研究結果加以改進。

在組織內部進行問卷調查

參考這份報告和今年的問卷調查題目¹，設計出專屬的內部問卷調查。問卷調查題目可以與目標對象有關，而且更加精細²。歡迎參閱「[研究方法](#)」一節，進一步瞭解我們如何執行研究。最後，別忘了依據調查結果採取行動！

分享所學

將您在實驗中學到的寶貴知識與全組織的人員分享。您可以正式向眾多相關人員報告、與社群進行非正式的做法交流，也可以與同事閒聊分享等。不妨多方嘗試，瞭解哪種方法最適合您的組織環境與文化，這也是一種實驗！



您會如何運用今年的研究結果？

歡迎加入 DORA 社群 (<https://dora.community>)，與其他成員分享您的經驗、相互學習，從他人的持續改進計畫汲取靈感。



¹ 2024 年問卷調查：<https://dora.dev/research/2024/questions/>

² 在軟體公司內部進行 DORA 問卷調查的經驗分享：<https://www.infoq.com/news/2024/08/dora-surveys-software-company/>

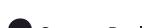
特別銘謝

我們今年達成了一個特別的里程碑：發布第 10 版 DORA 報告。感謝研究人員、專家、從業人員、主管和轉型顧問的投入與付出。因為有您，這份報告才能成形並持續改進。

從 Puppet Labs 和 IT Revolution Press 發布的第一版《開發運作現狀報告》以來，我們取得了長足進展，誠摯感謝 DORA 創辦人奠定的基礎。環境自那時起變化良多，我們在這些年也學到了很多，回想起來十分感動。

非常感謝今年研究的所有參與者。引領業界並影響實務做法是一項重責大任，您的貢獻對我們來說相當寶貴。

從早期到現今這令人期待的 AI 世代，我們衷心感謝曾經參與這趟旅程的每一個您。您的支持與見解讓我們獲益良多，期待未來十年能夠繼續合作及探索新知！

DORA 報告團隊	Marie-Blanche Panthou Miguel Reyes Yoshi Yamaguchi Jinhong Yu	Gene Kim 與 IT Revolution Laura Maguire 博士 James Pashutinski Ryan J. Salva Majed Samad
James Brookbank Kim Castillo Derek DeBellis Benjamin Good		
Nathen Harvey Michelle Irvine Amanda Lewis Eric Maxwell Steve McGhee Allison Park	DORA 顧問 Lisa Crispin Steve Fenton Denali Lumma Betsalel (Saul) Williamson	Harini Sampath Robin Savinar Sean Sedlock Dustin Smith Finn Toner
Dave Stanke Kevin Storer Daniella Villalba	現場顧問/專家 John Allspaw Birgitta Böckeler Sander Bogdan	Gold 贊助者  catchpoint  chronosphere   Deloitte.  Excella   liatrio  Middleware 
編輯	Michele Chubirka	Silver 贊助者
Seth Rosenblatt	Thomas De Meo	 autorabbit   XOps Apps  gather  honeycomb.io   Octopus Deploy 
志願譯者	Jessica DeVita Rob Edwards Nicole Forsgren 博士	
Andrew Anolasco Mauricio Meléndez		

作者



Derek DeBellis

Derek 是 Google 使用者體驗量化研究人員和 DORA 首席研究員，主要負責問卷調查研究和記錄檔分析，以及設計概念衡量方法，以便檢核產品或功能是否能創造資本價值。Derek 過去發表的研究探討了真人與 AI 之間的互動、COVID-19 確診對戒菸的影響、針對自然語言處理錯誤進行設計、使用者體驗在隱私權討論的角色、團隊文化，以及 AI 與員工身心健康和工作效率之間的關係。他正在進行一項業餘研究，探討如何模擬信仰和權力的傳播過程。



Kevin M. Storer

Kevin M. Storer 博士是 Google 開發人員體驗研究人員，並擔任 DORA 質性研究團隊的負責人。Kevin 具備專業的軟體工程經驗，研究所時接受社會和人文科學的跨學科訓練，自 2015 年起一直主導以人類為中心的軟體開發人員研究。他會探索各式各樣的問題脈絡、參與者的身分背景和研究方法。Kevin 的研究結果刊登於頂尖的科學期刊，主題涵蓋人工智能、資訊檢索、嵌入式系統、程式設計語言、普及運算和互動設計。



Amanda Lewis

Amanda Lewis 是 [DORA.community](#) 的開發主管，也是 Google Cloud 的開發人員關係工程師。她的工作是幫助開發人員、營運人員、產品經理、專案經理和主管建立關係。Amanda 曾任職於電子商務平台、內容管理系統、觀測工具的開發團隊，以及開發人員支援團隊。透過這些互動交流和對談滿足客戶需求，創造更好的業務成果。她會在工作時運用自身經驗和同理心，協助團隊瞭解及導入軟體推送與人工智慧做法。



Benjamin Good

Ben Good 是 Google 的雲端解決方案架構師，熱衷於透過雲端技術與自動化機制改良軟體推送做法。他運用解決方案架構師的專業知識，協助 Google Cloud 客戶解決問題，包括提供架構指引、發布技術指南及提供開放原始碼。加入 Google 之前，Ben 曾在美國丹佛博爾德地區的數間企業執行雲端作業，在過程中實踐 DevOps 做法。



Daniella Villalba 是 Google 使用者體驗研究人員，專門透過問卷調查研究，探索哪些因素能提升開發人員工作效率和滿意度。加入 Google 前，Daniella 曾研究冥想訓練的優點，以及影響大學生體驗的社會心理因素。她在佛羅里達國際大學 (Florida International University) 取得實驗心理學博士學位。

Daniella Villalba



Eric Maxwell 領導 Google 的 DevOps 轉型實務並提供建議，協助全球最優秀的公司持續進步，更快創造價值。Eric 以第一線工程師的身分度過前半段職涯，會建立自動化機制來處理各種作業，也會設身處地為其他從業人員著想。Eric 共同制定了 Google Cloud 應用程式翻新計畫 (CAMP)，同時也是 DORA 團隊成員。加入 Google 之前，Eric 任職於 Chef Software，並與同事有一段愉快的合作時光。

Eric Maxwell



Kim Castillo 是 Google 使用者體驗專案經理，自 2022 年起領導 DORA 計畫背後的跨職能作業，會監督研究運作情形，以及出版這份報告。Kim 也研究 Google Cloud Gemini 的使用者體驗。加入 Google 之前，她在科技業從事技術計畫管理和敏捷教練能力工作。Kim 從社會心理研究起家，主要探討家鄉菲律賓的法外處決、都市貧民窟發展和社區復原力。

Kim Castillo



Michelle Irvine 是 Google 的技術撰寫專員，主要研究說明文件和其他技術通訊內容。加入 Google 之前，她曾在教育出版業工作，以及擔任物理模擬軟體的技術撰寫專員。Michelle 擁有滑鐵盧大學 (University of Waterloo) 的物理學士學位，以及修辭與傳播設計碩士學位。

Michelle Irvine



Nathen Harvey 領導 Google Cloud 的 DORA 團隊。Nathen 從幾個出色的組織、團隊和開放原始碼社群汲取經驗，並與員工分享。他是多年 DORA 報告的共同作者，也是《97 Things Every Cloud Engineer Should Know》(雲端工程師不得不知道的 97 件事) 的貢獻者與編輯，該書於 2020 年由 O'Reilly 出版。

Nathen Harvey

受訪者和 公司統計結構

問卷調查對象

過去十年來，DORA 研究計畫致力於研究表現出色、技術導向的組織具備哪些能力，以及採取什麼做法和措施。約 39,000 位所屬組織規模和產業各異的專業人士向我們提供回覆，感謝各位與我們分享寶貴見解！今年，來自全球各行各業近 3,000 位專業人士分享了他們的經驗，協助我們瞭解表現出色的技術導向組織如何誕生。

今年的受訪者和公司統計結構問題採用 Stack Overflow 進行的研究。

在 2023 年，超過 90,000 名受訪者參與了 Stack Overflow 開發人員問卷調查¹⁰。該項問卷調查並未涵蓋所有技術從業人員，但收集到的資料已相當全面，具有代表性。

得力於該項問卷調查的規模，我們能夠找出 DORA 資料中的回覆偏誤，並瞭解預計的研究結果一般化程度。另外，我們認為該項 Stack Overflow 開發人員問卷調查中的受訪者和公司統計結構問題都經過精心設計，值得參考。

簡言之，DORA 和 Stack Overflow 研究的樣本並無重大差異，這代表我們充分相信 DORA 可以反映業界情況。

產業別

我們請問卷調查受訪者依據所屬組織的主要營運範疇，提供所屬的產業類別，共計 12 類。受訪者人數最多的產業依序為科技業 (35.69%)、金融服務業 (15.66%) 和零售/消費性/電子商務產業 (9.49%)。

產業別	受訪者百分比
科技	35.69%
金融服務	15.66%
零售/消費性/電子商務	9.49%
其他	5.94%
工業和製造	5.49%
醫療照護和製藥	4.60%
媒體/娛樂	4.26%
政府機關	3.89%
教育	3.66%
能源	3.03%
保險	2.39%
非營利	1%

員工人數

我們請問卷調查受訪者提供所屬組織的員工人數，共有九個區間，其中最常見的員工人數為超過 10,000 名 (24.10%)、100 至 499 名 (18.50%)，以及 1,000 至 9,999 名 (15.60%)。

組織規模	百分比
1 人公司	2.0%
2 至 9 人	3.2%
10 至 19 人	4.3%
20 至 99 人	14.5%
100 至 499 人	18.5%
500 至 999 人	11.2%
1,000 至 4,999 人	15.6%
5,000 至 9,999 人	6.7%
10,000 人以上	24.1%

身心障礙

我們按照《Washington Group Short Set》(華盛頓身心障礙團隊簡短問題集),提供六類身心障礙情形²⁰。DORA 研究已是第五年詢問身心障礙的問題,回答有身心障礙的受訪者百分比從 2022 年的 11%、2023 年的 6%,下降至 2024 年的 4%。

身心障礙	受訪者百分比
不符合任何身心障礙類別	92%
至少符合一個身心障礙類別	4%
不方便透露	4%

性別

我們請問卷調查受訪者提供性別,男性占 83%、女性占 12%、希望由自己描述者占 1%,拒答者占 4%。

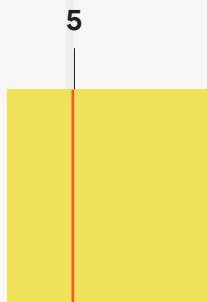
性別	百分比
男性	83%
女性	12%
自行描述	1%
不方便透露	4%

經歷

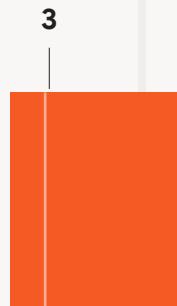
我們請問卷調查受訪者回答他們在目前職務和團隊的年資，並發現總工作經驗、在目前職務的年資，以及在目前團隊的年資中位數分別為 16 年、5 年和 3 年。

問題

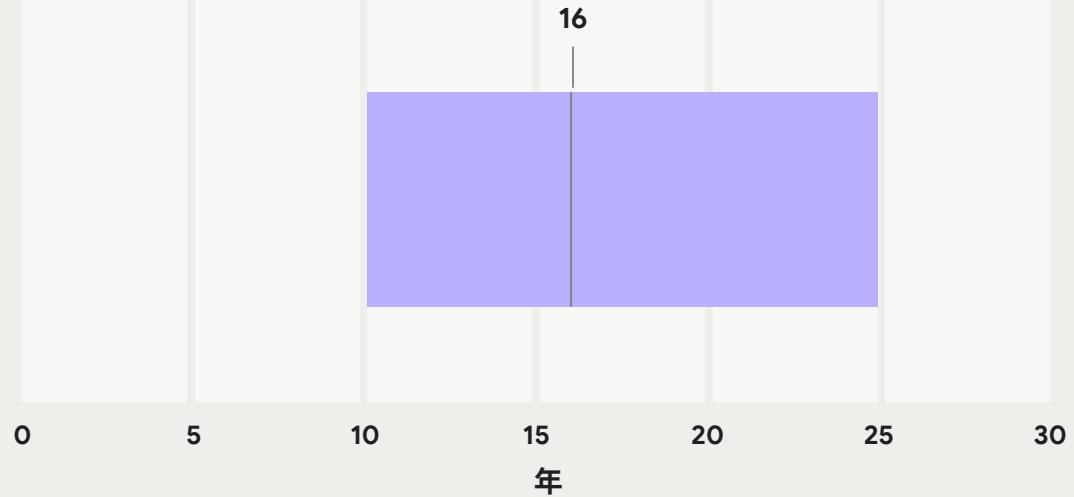
您在團隊從事與目前職務類似的工作已有幾年？



您在目前所屬的團隊已工作幾年？



您有幾年工作經驗？

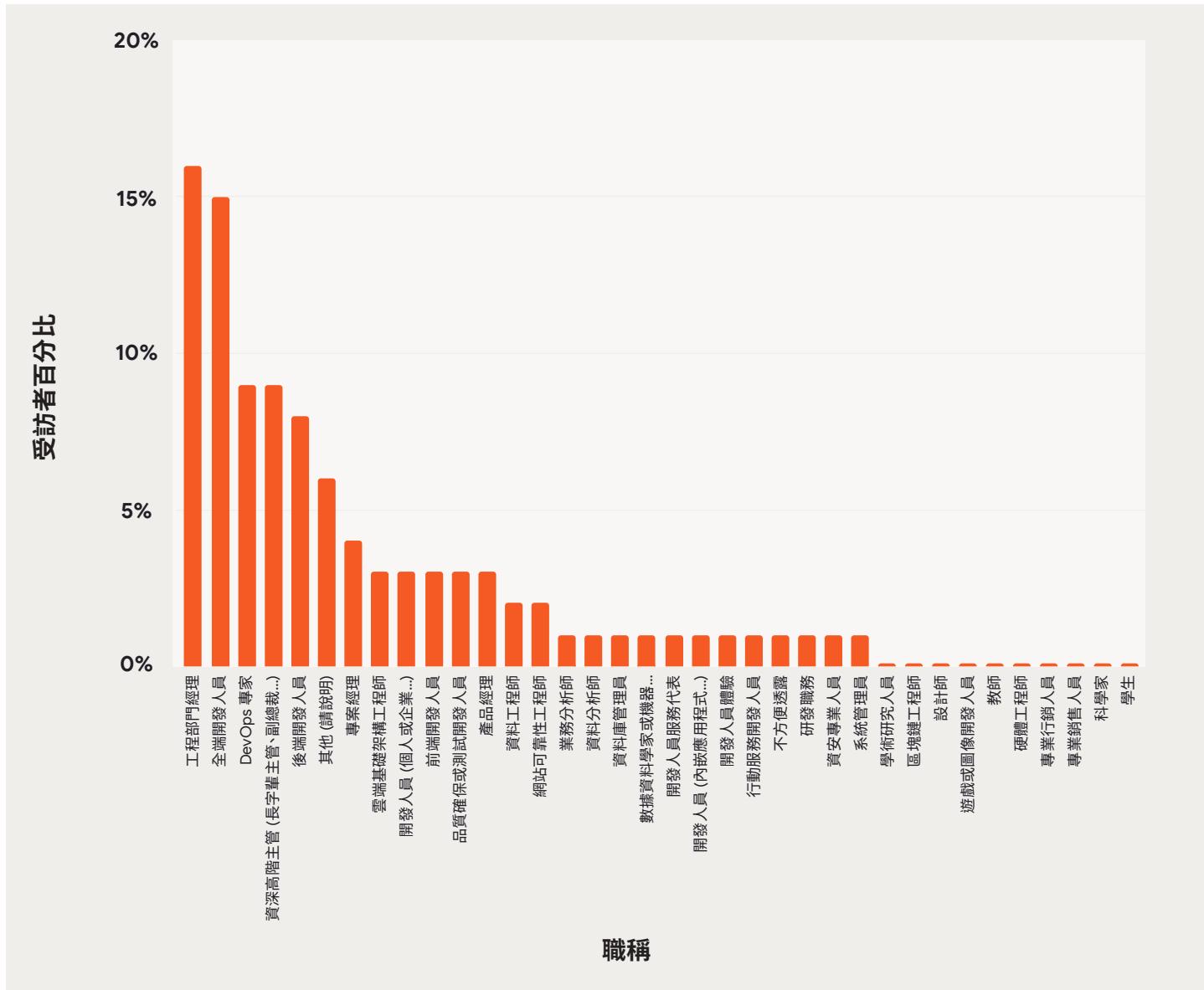


方塊寬度代表第 25 和第 75 個百分位數之間的範圍，貫穿方塊的直線為中位數。

職務

分析資料時，我們合併了某些職務，讓受訪者人數占比偏小的職務也有一定的代表性。研究資料中，下列其他類別的受訪者占比相當高：

- 開發人員，占受訪者的 29%。
- 主管，占受訪者的 23%。
- 高階主管，占受訪者的 9%
(比 2023 年多了 33%)。
- 分析類職務，約占受訪者的 5%。



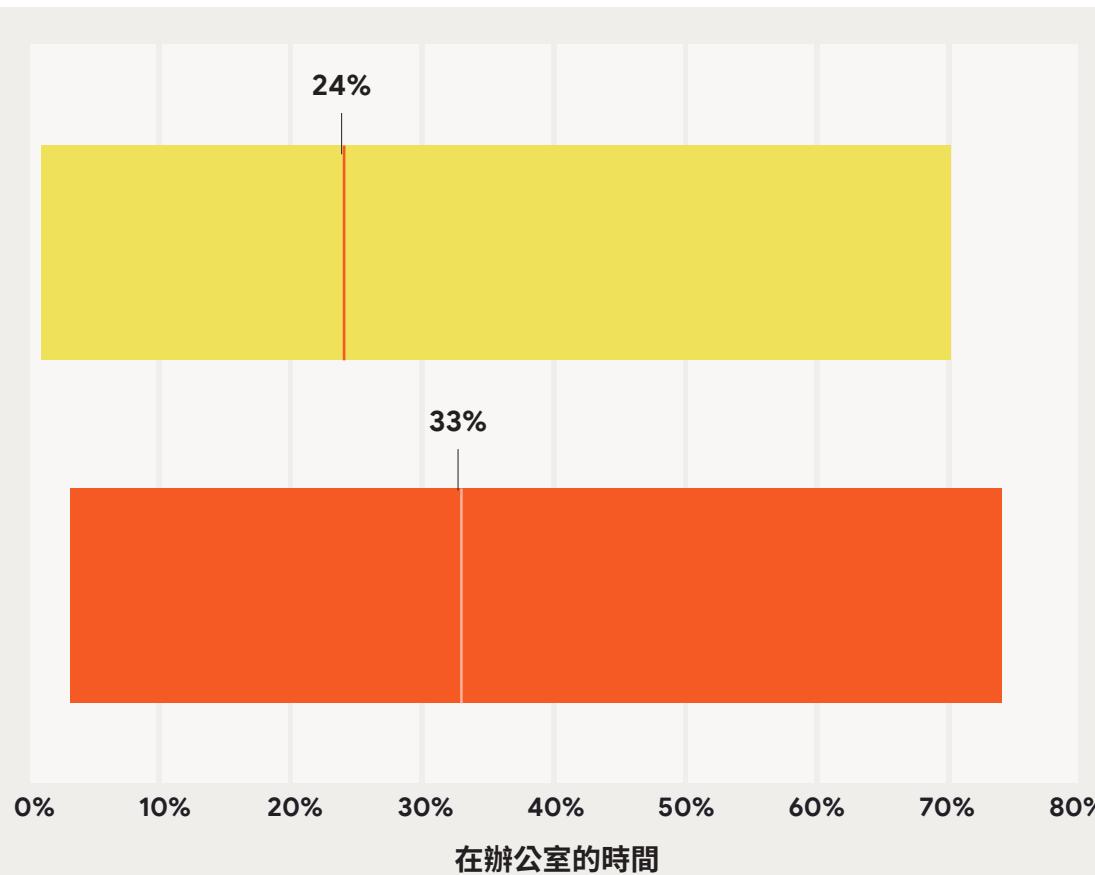
聘僱狀態

我們調查了問卷調查受訪者目前的聘僱狀態，並發現絕大多數 (90%) 為組織的全職員工。

聘僱類型	百分比
全職約聘員工	6%
全職員工	90%
兼職約聘員工	1%
兼職員工	2%

工作地點

即使企業今年仍然要求員工重返辦公室，我們發現分布模式與去年大致相同。在分布資料的尾端，這個現象尤其明顯。中位數較去年增加 37.5%，這確實呈現出「混合辦公」或至少「定期進辦公室」日漸成為主流。



方塊寬度代表第 25 和第 75 個百分位數之間的範圍，貫穿方塊的直線為中位數。

國家/地區

受訪者來自 104 個國家/地區。我們向來很開心收到全球從業人員的問卷調查回覆，感謝各位鼎力相助！



國家/地區

美國	義大利	新加坡	冰島	盧森堡	瓜地馬拉
英國	瑞士	阿爾巴尼亞	伊朗	尼加拉瓜	香港特別行政區
加拿大	阿根廷	喬治亞	約旦	巴基斯坦	馬爾他
德國	墨西哥	希臘	肯亞	秘魯	模里西斯
日本	葡萄牙	菲律賓	沙烏地阿拉伯	韓國	摩洛哥
印度	奧地利	匈牙利	斯洛伐克	斯里蘭卡	尼泊爾
法國	羅馬尼亞	塞爾維亞	斯洛維尼亞	突尼西亞	巴拉圭
巴西	芬蘭	阿富汗	泰國	安道爾	史瓦濟蘭
西班牙	土耳其	阿爾及利亞	烏茲別克	巴貝多	敘利亞阿拉伯共和國
澳洲	保加利亞	埃及	安哥拉	貝里斯	台灣
荷蘭	愛爾蘭	印尼	亞美尼亞	貝南	前南斯拉夫馬其頓共和國
中國	以色列	俄羅斯聯邦	波士尼亞與赫塞哥維納	玻利維亞	千里達及托巴哥
瑞典	比利時	烏克蘭	多明尼加共和國	布吉納法索	烏拉圭
挪威	智利	越南	厄瓜多	葛摩	委內瑞拉玻利瓦共和國
紐西蘭	哥倫比亞	孟加拉	愛沙尼亞	象牙海岸	
波蘭	捷克共和國	白俄羅斯	哈薩克	薩爾瓦多	
南非	馬來西亞	哥斯大黎加	拉脫維亞	衣索比亞	
丹麥	奈及利亞	克羅埃西亞	立陶宛	甘比亞	

種族和族群

我們請問卷調查受訪者提供自己的種族和族群，占比最高者為白人 (32.4%) 和/或歐洲人 (22.7%)¹。

種族或族群	百分比	種族或族群	百分比
白人	32.4	中東人	1.3
歐洲人	22.7	混血	0.4
亞裔	9.9	中美洲人	0.4
北美洲人	4.6	不確定	0.4
印度人	4.1	北非人	0.4
不方便透露	4.1	加勒比海人	0.2
西班牙或拉丁美洲裔	3.5	中美洲人	0.2
南美洲人	3.2	南亞人	1.7
東亞人	2.5	族教群體	0.2
非洲人	1.8	太平洋島居民	0.2
南亞人	1.7	原住民 (例如美國或澳洲原住民)	0.1
多元種族	1.5		
自述種族	1.5		
東南亞人	1.4		
黑人	1.3		

¹ <https://survey.stackoverflow.co/2023/>

² <https://www.washingtongroup-disability.com/question-sets/wg-short-set-on-functioning-wg-ss/>

研究方法

研究方法理應像食譜，可以幫助您重現我們的研究結果，以及判斷 DORA 產生及分析資料的方式是否能帶來實用資訊。由於篇幅有限，我們無法呈現所有細節，不過希望這節內容可以幫助您開始思考這些面向。

問卷調查建立過程

挑選問題

判斷是否要將某個題目納入問卷調查時，我們會考量下列面向：

這道題目是否…

- 已有基礎，可以與過往的結果相互比較？
- 涉及業界想要實現的結果（例如出色的團隊表現）？
- 涉及業界考慮投注資源的能力（例如 AI）？
- 涉及可以幫助人們達成目標的能力（例如優質的說明文件）？
- 有助於我們評估樣本的代表性（例如職務或性別）？
- 能防範出現推論偏誤（例如程式設計語言或職務）？
- 能精準回答絕大多數受訪者的問題？

判斷是否將題目納入問卷調查時，我們會參考文獻、與 DORA 社群交流、進行認知訪談和平行質性研究、與領域專家合作，以及舉行團隊研討會，以便做出明智的決定。

問卷調查體驗

我們非常重視提高 DORA 問卷調查的可用性，並會透過認知訪談和可用性測試，確保問卷調查符合特定要求，包括：

- 平均來說，問卷調查必須能在短時間內填寫完畢
- 問卷題目清晰易懂
- 填寫起來應該相當輕鬆。不過這相當有挑戰性，因為領域的技術性很高

資料收集

本地化版本

每年都會有世界各地的人員填寫問卷調查，為觸及更多人，今年的問卷調查提供英文、西班牙文、法文、葡萄牙文、日文和簡體中文版本。



收集問卷調查回覆

我們透過多種管道收集問卷調查回覆，可以分為「自然募集」和「固定樣本」兩類。

「自然募集」是在我們可以運用的所有社群媒體管道，公開邀請大眾填寫問卷調查。其中包括發布網誌文章、進行電子郵件廣告活動，以及在社群媒體上發布貼文，我們也請社群成員採取同樣做法（也就是「雪球式抽樣」）。

同時，我們運用「固定樣本」來彌補自然募集管道的不足。我們從廣大技術社群中募集一般視為弱勢族群的受訪者，也試著向特定類型的產業和組織收集足夠的回覆。

簡單來說，由於自然募集管道無法控管對象，我們透過固定樣本補足所需的研究對象。另外，固定樣本管道可以確保募集到足夠的受訪者，因為我們無法確定自然募集的回覆內容是否符合所做分析類型的要求。今年的自然募集受訪者人數足以進行分析，我們也透過固定樣本補足了某些族群的缺口。

問卷調查流程

我們今年有非常多想探究的問題，但時間不夠。我們有下列幾個選擇…

- 提供非常長的問卷調查
- 挑選幾個領域深入研究
- 隨機將受訪者分派至不同主題

我們不想放棄任何感興趣的領域，因此決定隨機將受訪者分派至三種流程的其中一種。三種流程中有許多部分重疊，不過各流程都深入探究不同的領域。

三種路徑如下：

- AI
- 工作場所
- 平台工程

問卷調查分析

驗證測量指標

問卷調查研究的概念類型相當多元。描述這些概念的方法有很多種，其中一種做法是將概念測量指標稱為變數，這些變數是模型的構成要素，模型則用於我們的研究。大致來說，我們可以從內部和外部分析這些測量指標的效力。

為了瞭解評估指標的內部效力，我們會研究可代表某個概念的具體行為。例如，「優質說明文件」可能指的是用說明文件解決問題。

多數變數由多項指標構成，因為我們想探討的概念具有多種面向。

為瞭解變數的多面向特性，我們會測試用來代表構想的項目，評估其共通程度。如果共通程度高（也就是共同變異性高），我們會假設這些項目的背後有共同的概念，例如利益。

舉例來說，「幸福感」是一個多面向的概念。一個人感到幸福時，我們預期他會有特定感受、行為和思考模式，也就是說，我們假設幸福是產生特定感受、想法和行為的基礎。

因此，我們預期幸福感會與特定幾種感受、想法和行為同時出現。隨後，我們會詢問有關這些情緒、想法和行為的問題，並使用驗證性因素分析，測試這些元素是否確實同時出現。

我們今年使用 lavaan¹ R 套件進行這項分析工作。Lavaan 會傳回多種合理度統計資料，幫助我們瞭解受訪者的回答模式是否確實反映某些概念。

如果某個概念的指標沒有共通性，代表我們尚未找到可靠的測量方法，因此可能需要修改或捨棄這個概念。

至於概念的外部效力，我們則是評估該概念在真實世界中的合理度。我們可能會預期某個概念與其他概念之間有特定關係，有時甚至呈現負相關，例如快樂與悲傷。

如果測量出來之後，發現快樂與悲傷呈現正相關，就有理由懷疑測量方法或理論有誤。

同樣地，我們可能預期兩個構想之間呈現正相關，但並非高度正相關。例如，工作效率和工作滿意度之間可能呈現正相關，但我們不認為這是兩種相等的概念。如果關聯性過高，代表我們測量的可能是同一個概念。也就是說，我們的測量方法不足以準確區分這兩個概念之間的差異，或是並沒有我們假設的差異。

模型評估

我們依據多項假說建立假說模型，試著瞭解世界運作方式的某些面向，並檢核模型與問卷調查資料的適合程度。我們以簡約法評估模型，也就是從非常簡單的模型開始²，然後不斷提高複雜程度，直到模型不再合理為止。

舉例來說，我們預測軟體推送和運作成效之間的交互作用會影響組織表現。最簡單的模型未計入交互作用：

組織表現 ~ 軟體推送成效 + 運作成效

第二個模型加入交互作用：

組織表現 ~ 軟體推送成效 + 運作成效 + 軟體推送成效 × 運作成效

我們依據《Regression and other stories》(迴歸分析與其他實例)³ 和《Statistical Rethinking》(統計反思)⁴ 的建議，使用留一交叉驗證 (LOOCV)⁵ 和渡邊–赤池資訊量準則⁶判斷是否需要再提高複雜程度。

以有向無環圖進行因果推論

經過驗證的模型可以說明推論因果關係時須留意的事項，來看看進行因果推論時會出現的難題。

我們「試著」找出因果關係的幾個理由如下：

您的問題本質上有因果關係，而想瞭解做某件事是否會帶來特定結果。如果沒有因果關係，就不會投注資源來嘗試。

分析結果取決於我們對世界因果關係的理解。如果迴歸分析時使用的資料有變，得到的實際數字就會不同。我們判斷資料產生的方式（即因果論述）會影響迴歸分析採納的資料，因此應該清楚明確。

因果推論是好奇心的體現，會耗用許多時間。我們經常會想瞭解世界不同面向之間的關係與原因，但是不需要針對生活中的所有現象進行實驗，也能找出因果關係。

因果推論是行動的核心，希望能透過這份報告協助您做出行動決策。

經過驗證的模型可以讓我們知道，瞭解成效時應該考量的部分。簡單來說，我們可以透過 A/B 實驗取得資料，也就是建立兩個完全相同的情境，其中只有一處不同。按照邏輯推論，兩個情境之間的任何差異都來自當初那個不同之處。

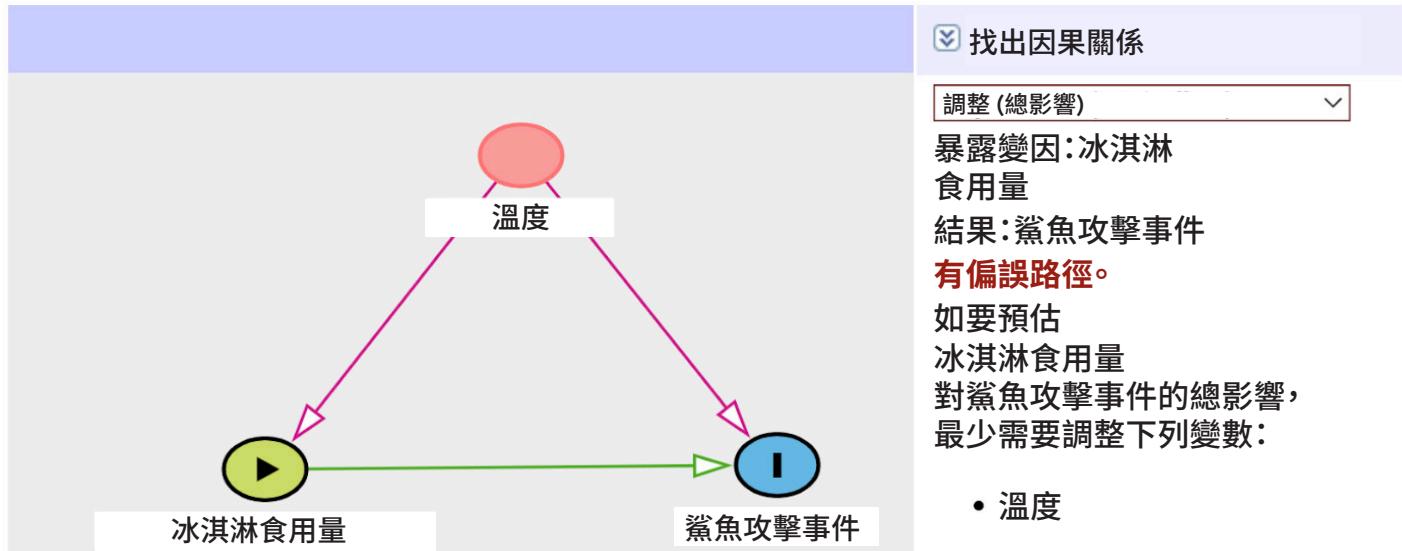
可觀察資料和問卷調查資料中的關聯性沒有那麼涇渭分明，受訪者本身就有許多差異，因此難以判斷。進行因果推論時，我們會為了仿照實驗，試著在一次試驗考量這些差異，也就是除了一項變因（例如 AI 採用率），其餘部分均相同。

請看這個經典的例子：冰淇淋「引發」鯊魚攻擊事件。這項觀察有問題，即人們會在炎熱的天氣吃冰淇淋，也會去海邊。另外，「人們會吃冰淇淋及去海邊」的情境與「人們不會吃冰淇淋及去海邊」的情境不同。因此，這項資料不符合實驗邏輯，原因出在「溫度」這項干擾變數。

有向無環圖 (DAG) 有助於發掘情境中有何不同，並找出解決方法，這樣就能仿照實驗，讓情境中只有一處不同。接著看看 DAG 如何引導我們剖析前述例子，我們想要量化冰淇淋食用量對鯊魚攻擊事件的影響：

我們在工具畫出模型，並指定想要瞭解的影響，接著工具會指出哪些因素會導致預估出現偏誤。針對這個例子，工具指出如要預估冰淇淋食用量對鯊魚攻擊事件的影響，就必須針對溫度調整模型，透過這種統計方式確保冰淇淋食用量和其他變因的影響均一致，接著才能觀察冰淇淋食用量的變化是否仍會讓鯊魚攻擊次數出現波動。

如要查看模型的概要資料，請參閱「[模型](#)」一章。



圖片來源：<https://www.dagitty.net/dags.html>。

有向無環圖可以指出分析特定影響時須考量的部分。

舉例來說，分析 AI 採用情形對工作效率的影響時，需要考量哪些因素？

貝式統計

我們採用貝式統計方法來分析這次的資料，這個方法有許多優點：

- 避免從顯著或不顯著的角度思考（請 10 個人解釋頻率統計中的 p 值時，您可能會得到 10 種答案）
- 有助於瞭解資料已知時，某個假設成立的機率，而非假設成立時，該筆資料出現的機率
- 可將已知資訊納入模型，或是至少明確指出不清楚的部分⁷
- 必須面對模型建立程序採用的基礎假設
- 有助於檢視事後分布，瞭解規模和不確定性，以及模型解釋資料的整體方式與成效。最終，這讓我們得以充分瞭解資料已知時，自己知道和不知道的部分
- 這個框架富有彈性，能以非常一致的方式解決許多統計問題

何謂「模擬」？

「模擬」並非編造資料。我們使用貝氏統計方法計算事後機率，掌握「不同參數值的預估出現機率」⁸。「模擬」是指計算這個事後機率超過 1,000 次，藉此找出資料已知時最可信的參數值 (平均值、Beta 權重值、Sigma 值、截距值等)。

「事後機率就像是一個桶子，當中裝滿 0.1、0.7、0.5、1 等參數值。桶子內，個別值的出現機率與其事後機率呈正比，因此接近高峰的值會比接近尾部的值常見得多」⁹。

也就是說，我們可以透過模擬找出可能的資料解讀方式，並瞭解不確定性有多高。您可以將個別模擬視為一個小型 AI，這個 AI 只知道我們的資料和幾項規則，並要依據資料猜出參數值。只要模擬 4,000 次，即可針對指定參數得出 4,000 個小型 AI 的猜測值。

這些猜測值非常有用，例如我們可以計算平均猜測值、89% 的猜測值¹⁰介於哪個區間、多少猜測值超過特定門檻、猜測值之間的變異度有多大等。您甚至可以合併多個模型的猜測值 (模擬) 等，進行有趣的操作。

如果圖形中有許多線條或潛在值分布，代表我們呈現的是最合理的資料解讀方式，以及不確定性有多高。

整合研究結果與社群看法

DORA 研究結果為技術導向的團隊和組織提供珍貴觀點，不過如要充分瞭解其價值，仍須對話及相互學習。與 DORA 社群互動之後，我們取得多樣化的見解、挑戰假說，並發掘新的方式來解釋及應用這些調查結果。

誠摯邀請您加入 DORA 社群 (<https://dora.community>)，分享自己的經驗、從他人身上學習，以及探索實作這些建議的做法。歡迎與我們一同探索這些洞察的最佳運用方式，在您的組織實現深具意義的變革。

研究結果中的 跳躍式推論

為了讓年度問卷調查更加完善，我們今年加入了深入的半結構化訪談，藉此結合多元觀點、提供脈絡，以及闡明量化研究結果。訪談指南比照問卷調查主題設計，並透過 Google Meet 遠距進行訪談，個別訪談的時長約為 75 分鐘。

我們總共訪問了 11 人，他們都符合問卷調查納入條件。訪談過程都有錄影及錄音。訪談時長介於 57 至 85 分鐘之間，從所有受訪者身上總共收集了 14 小時又 15 分鐘的資料。訪談對象的資料已匿名化處理，並以「訪談對象 (數字)」的形式代稱，N 指的是接受訪談的順序。

所有訪談內容都使用自動化軟體轉錄，再進行手動編碼 (事前依據問卷調查主題設計而成)。最終版報告中的引言都是重新播放後手動轉錄而成，接著才列入內文。報告作者群在訪談對象引言加入的文字以中括號 ([]) 表示，移除的文字則以刪節號 (...) 表示，只有在需要釐清意義時才會加以編輯。

我們的目標是務實地探討這個世界的現象，幫助所有讀者改良工作方式，因此難免會省略一些複雜的脈絡，而這就是模型的意義所在。Jorge Luis Borges 在《On Exactitude in Science》(論科學的精確性) 曾提及一則小故事，描述某個帝國試著繪製 1:1 比例的疆域圖¹¹。故事的荒唐之處在於，這張地圖完全無用 (至少我認為如此)。簡化不應犧牲實用性。

即便如此，我們還是要澄清一些跳躍式推論。

因果關係

按照 John Stuart Mill 的說法，必須滿足三個條件，才能說 X 造成 Y¹²：

- **相關性**: X 會與 Y 共變
- **時間順序在前**: X 的發生時間早於 Y
- 已考量偏誤路徑 (如前文 DAG 一節所述)

我們相信自己對相關性的瞭解正確無誤，這是標準統計程序中常見的條件。問卷調查想瞭解的是特定時間點的情況，因此時間順序只是理論層面的討論，無法透過資料呈現。

至於偏誤路徑的部分，如同前文提及的結構方程式模式和有向無環圖，我們已考量偏誤路徑。不過相較於「時間順序在前」，偏誤路徑牽涉高度理論化的討論，資料中會探討其影響。

我們想說的是，問卷調查並非縱向研究或非常精密的實驗，不過我們認為因果推論是理解世界的方式，並會盡可能運用這個領域的新興技術來提供準確預測。相關不代表有因果關係，但這確實反映了人們思考因果關係的方式。

微觀現象 -> 宏觀現象

我們經常會觀察個別員工的能力與更高層級之間的關聯性。舉例來說，我們會將個別員工採用 AI 的情形連結至應用程式或服務的效能，還有團隊表現。這種關係乍看之下不太直觀。如果是宏觀現象引發個別員工的現象，通常較容易察覺。舉例來說，通貨膨脹（宏觀）會影響我是否買雞蛋（微觀），這聽起來比「我不買雞蛋，因此造成通貨膨脹」更合理。

組織表現（宏觀）會影響個別員工的身心健康（微觀）也是同樣的道理。依據經驗法則，組織對個別員工的影響力可能比個別員工對組織的影響力來得大。

既然如此，為何要說個別員工的行為會影響團隊表現或組織表現？對此，我們做出了跳躍式推論，但並非全無道理。從宏觀的角度來看，我們假定下列描述通常都成立：

$$p(\text{個別員工做 } X \mid \text{組織做 } X) > p(\text{個別員工做 } X \mid \text{組織不做 } X)。$$

我們認為，個別員工任職的組織或團隊會做某件事（X），這位員工也做 X 的機率較高。因此，如果個別員工會做 X，代表所屬團隊和組織也經常做 X。當然，這項推論的干擾因素非常多，不過應該會有上述模式。因此，這樣的假定可以讓我們找出幾項重要的能力。

讓我們先跳離 DORA 的世界，看看另一個例子：假設兩個國家的國民平均身高不同，其中一個國家是 168 公分，另一個是 188 公分，兩國身高資料的標準差相同。如果您隨機從這兩個國家中各抽出一人，比較高的那個人較有可能來自哪個國家？如果重複執行這個動作數千次，您會發現高個子通常來自平均身高較高的那個國家，抽中者的身高也會大致接近該國的國民平均身高。

雖然不見得有必要，但我們快速模擬了上述情況，驗證這樣的假說是否成立：

```
#R code  
  
#set seed for reproducibility  
set.seed(10)  
  
#6'2 and 5'6  
height_means = c(6 + 1/6, 5.5)  
  
#constant standard deviation at 1/4 of  
#foot  
std_dev = 0.25  
  
#random draws  
draws = 1000  
  
#random draws from country A  
country_a <- rnorm(draws, mean = height_  
means[1], sd = std_dev)  
  
#random draws from country B  
country_b <- rnorm(draws, mean = height_  
means[2], sd = std_dev)  
  
#how of the draws represent the correct  
#difference  
represented_difference = sum(country_a >  
country_b) / 1000  
  
#show results as percentage  
represented_difference * 100
```

結果並不讓人意外。1,000 次隨機抽樣中，97.2% 的值都符合假設。當然，如果採用非隨機抽樣方式、兩國之間的平均身高差較不明顯，或是樣本數偏少，可能就會推論錯誤。但無論如何，我們的推測正確：宏觀層面的差異通常也會體現在微觀層面。

1. Rosseel Y (2012)。《lavaan: An R Package for Structural Equation Modeling》(結構方程式模式專用的 R 套件)。Journal of Statistical Software, 48(2), 1-36。https://doi.org/10.18637/jss.v048.i02
2. 這種做法也需要檢核潛在干擾因素。
3. Gelman、Andrew、Jennifer Hill 與 Aki Vehtari, 2021 年。《Regression and Other Stories》(迴歸與其他相關分析)。N.p.:Cambridge University Press。
4. McElreath、Richard, 2016 年。《Statistical Rethinking: A Bayesian Course with Examples in R and Stan》(統計反思：R 與 Stan 中的貝氏統計教學與範例)。N.p.:CRC Press/Taylor & Francis Group。
5. Gelman、Andrew、Jennifer Hill 與 Aki Vehtari, 2021 年。《Regression and Other Stories》(迴歸與其他相關分析)。N.p.:Cambridge University Press。
6. McElreath、Richard, 2016 年。《Statistical Rethinking: A Bayesian Course with Examples in R and Stan》(統計反思：R 與 Stan 中的貝氏統計教學與範例)。N.p.:CRC Press/Taylor & Francis Group。
7. 我們的事前推測通常不夠有力 (偏向猜想、中性，而且資訊偏少)，不過經過確認，研究結果未受到這些事前推測影響。
8. McElreath、Richard, 《Statistical rethinking: A Bayesian course with examples in R and Stan》(統計反思：R 與 Stan 中的貝氏統計教學與範例)。Chapman and Hall/CRC 出版, 2018 年, pg.50
9. McElreath、Richard, 《Statistical rethinking: A Bayesian course with examples in R and Stan》(統計反思：R 與 Stan 中的貝氏統計教學與範例)。Chapman and Hall/CRC 出版, 2018 年, pg.52
10. 我們依據 McElreath 在《統計反思》第 56 頁的理由選用 89%「為何選擇這些值？沒有原因... 而且這些值避開了傳統偏好的 95% 信賴區間，因為這信賴區間會讓許多讀者無意識地想進行假說檢核。」報告提供的區間只是想呈現「參數值與模型和資料相配的合理範圍」。
11. Borges, J. L. (1999)。《Collected fictions》(虛構集)。Penguin。
12. Duckworth、Angela Lee、Eli Tsukayama 與 Henry May, 《Establishing causality using longitudinal hierarchical linear modeling: An illustration predicting achievement from self-control》(以縱向階層線性模型確認因果關係：詳述如何透過自制力預測成就)。Social psychological and personality science 1, no. 4 (2010):311-317。

模型

我們往年是建立一個巨大的模型，並透過多種結構方程式模型建立技術（偏最小平方、共變異數和貝氏統計）加以驗證。2023 年的報告中，我們改為探討多個小型模型，以便瞭解特定程序。

舉例來說，我們建構了精密的模型來研究優質說明文件的影響。為瞭解特定影響而量身打造專屬小型模型，有幾個重要優點¹：

- 可以輕鬆找出模型不適配的領域
- 加入模型的所有變數都有影響力。如果模型變大，就非常難以瞭解變數之間的各種交互作用
- 避免將不恰當的變數設為條件，進而推論出錯誤的關係²

我們如何使用模型？

我們想探究的問題非常多，不過許多關鍵問題的形式大同小異：

如果做了 X, Y 會出現什麼變化？

X 通常是指做法，例如編寫優質的說明文件、採用 AI 技術，或是投注資源來營造文化。

Y 通常是我们想要達成或避免的情況，從個別員工層級（例如提高工作效率）到組織層級（例如提高市占率）都有可能。

我們的目標是建構、評估及使用模型³，藉此回答這種形式的問題。我們致力於準確預測做 X 帶來的重要結果變化⁴，並會呈現這些影響的兩個關鍵面向：

1. 我們對於該項影響的**方向**有多肯定，意即有清楚這項做法會帶來正面還是負面影響？
2. 我們對於該項影響的**規模**有多肯定。我們會預估特定做法的相對影響力，並指出這些預估的不確定性程度。

以下是 DORA 今年想瞭解的幾項能力：

- **AI 採用情形**
- **平台使用情形**
- **平台存在時間**
- **轉型領導力**
- **優先要務的穩定度**
- **以使用者為中心**

以下是我們今年想觀察的結果和結果群組：

- **個別員工的表現與身心健康（例如工作倦怠）**
- **團隊表現**
- **產品效能**
- **開發工作流程（例如程式碼集的複雜程度和說明文件品質）**
- **軟體推送成效**
- **組織表現**

模型結構不變

我們選出的結果本身，都是組織最終的目標，至少當中某些目標是如此。如果您發現組織和團隊表現與軟體推送成效無關，或許不需要太擔心軟體推送成效偏低。

不過，即便您發現個別員工的身心健康不會影響組織表現，還是建議您優先照顧員工的身心健康。

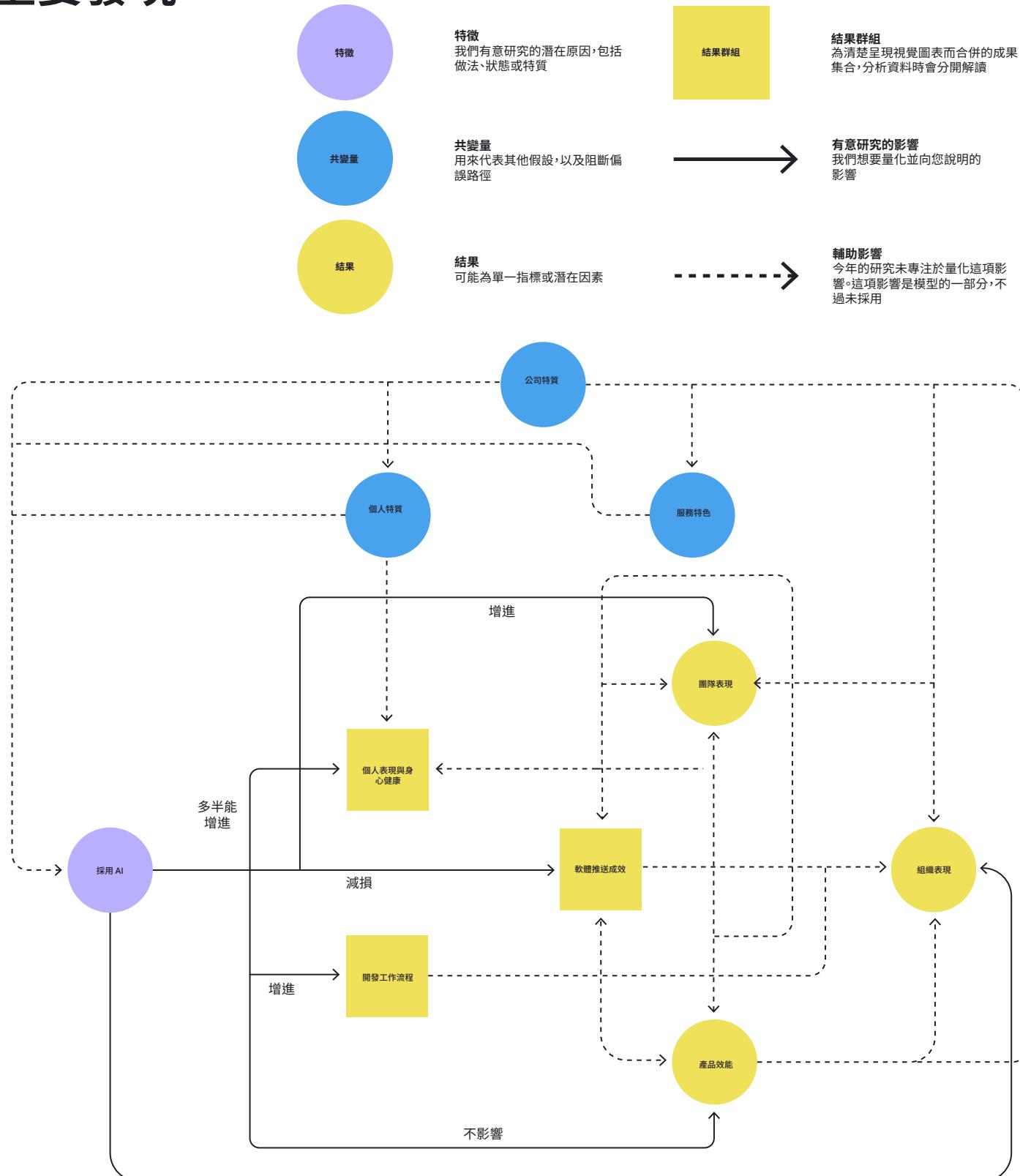
過去三年來，我們建立並探索了許多精細的假說，尤其是調節效果和中介效果。

我們今年研究這類假說的時間較少，而是專心預估某項能力對結果的影響。也就是說，針對各項能力建立的模型大致相同。

因此，「AI 採用情形的影響」和「以使用者為中心的影響」適用的模型在設計上非常相似。雖然我們可以複製模型再修改能力名稱，但這可能會降低模型的實用度。

報告只會呈現研究 AI 採用情形的模型，但請記得其他個別模型的架構或形式大致相同。如果您有興趣自行分析，可以使用 [DAGitty](#) 等工具建立這類模型，這樣應該就能採用與我們最接近的迴歸分析做法。不過為方便閱讀，報告呈現的模型已略微簡化。另外，雖然各能力的模型非常相似，但是這些能力的影響並不相同。舉例來說，您會在下文發現「AI 採用情形」通常會對軟體推送成效產生負面響，「內部說明文件」和「以使用者為中心」則相反。如要瞭解詳情，請見相關章節。

重要發現



1. Gelman 等人在《Regression and other stories》(迴歸與其他相關分析) 的第 495 頁至 496 頁，分享了一些有啟發性的重要見解，包括 B.6 提及的「適合多個模型」，以及 B.9 的「以針對性的方式進行因果推論，而非將因果關係視為大型迴歸分析的副產物」。
2. 如要詳細瞭解這個部分，請參閱《Statistical Rethinking》(統計反思) 一書的第 6 章，尤其是有關對撞因子偏誤的段落。
3. 請參閱「研究方法」一節，瞭解這些模型與有向無環圖的關係。
4. 「研究方法」一節簡短說明了因果關係。

推薦參考資源

加入 DORA 社群來討論、學習和合作，一起提升軟體推送和運作成效：<https://dora.community>

進行 DORA 快速檢驗：<https://dora.dev/quickcheck>

探索哪些能力有助於營造學習風氣，並建立快速的流程和意見回饋機制：<https://dora.dev/capabilities>

增進開發人員對生成式人工智能的信任：<https://dora.dev/research/2024/trust-in-ai/>

推薦讀物：《Accelerate: The science behind devops: Building and scaling high performing technology organizations》(ACCELERATE：精益軟體與 DevOps 背後的科學)。IT Revolution：<https://itrevolution.com/product/accelerate>

推薦讀物：《Team Topologies: Organizing Business and Technology Teams for Fast Flow》(團隊拓樸：有效組建業務與技術團隊，實現高速工作流程)，IT Revolution Press：<https://teamtopologies.com/>

DORA 研究計畫出版品，包括往年的 DORA 報告：<https://dora.dev/publications>

研究和報告的常見問題：
<http://dora.dev/faq>

勘誤表 - 閱讀這份報告並提交變更、修正和澄清事項：<https://dora.dev/publications/errata>

確認您閱讀的是不是最新版的 2024 年 DORA 報告：<https://dora.dev/vc/?v=2024.3>

《加速發展：2024 年開發運作現狀報告》
由 Google LLC 編製，並已取得 CC BY-NC-SA 4.0 授權

