

克勞德代碼： 從基礎到精通



—— 2026 年完整的代理式軟體開發指南 ——



安裝與
快速入門



提示工程



多代理
工作流程



安全性與
最佳實踐



真實世界
專案開發



Stripe 與
Wiz 案例研究



CI/CD
自動化

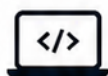


代理式編碼
的未來趨勢

一本全面、實用的指南，深入解析 Claude Code 的代理式編碼系統，
如何改變軟體的建構方式。從首次安裝到企業部署——精通工具、
技巧與工作流程，這些工具、技巧與工作流程都是頂尖工程團隊信賴的。



每章實用範例
即學即用



可立即實作的
專案範例



來自真實世界
團隊的實戰經驗



超過 100K+
使用者的研究基礎

STEVE T.

克勞德代碼：從基礎到精通

2026 年智慧軟體開發完全指南

Steve T. Publications

這本書的網址是 <https://leanpub.com/claudecodetw>

這個版本發布於 2026-07-15



這是一本Leanpub書。Leanpub 通過簡潔出版過程賦予作者和出版商能力。簡潔出版是使用輕量級工具和多次迭代來獲得讀者反饋，並在你有了正確的書之後再開始建立動力的一種出版進行中的電子書的行為。

© 2026 Steve T. Publications

Contents

2026 年代理式軟體開發完全指南	1
精通 Claude Code、多代理 workflow 與企業部署的綜合手冊	1
前言：不再敲鍵盤的工程師	3
本書將教您的內容	3
本書適合誰閱讀	3
如何使用本書	4
證據基礎	4
關於本書未涵蓋內容的說明	4
核心論點	5
2026 年的局勢	5
第 1 章：軟體開發中的代理式革命	6
從自動完成到代理：五年的演進	6
是什麼讓 Claude Code 不同：終端機優先的代理式架構	7
Boris Cherny 的工作流：創始人實際如何使用 Claude Code	8
勞動分工：人類為指揮者、代理為建構者	9
2026 年 AI 編碼代理的局勢	10
為何是現在？市場採用與向代理式工作流的轉變	11
批判性分析：代理式工具的權衡與限制	11
第 2 章：Claude Code 的解剖	14
代理循環：收集、行動、驗證	14
幕後模型：Sonnet、Opus、Fable 與模型選擇	14
Token 管理數學：理解您的花費	15
內建工具：檔案操作、搜尋、執行、Web、程式碼智慧	15
.claude 目錄：配置、CLAUDE.md 與記憶	15
情境視窗：如何填充、自動壓縮與 Token 管理	15
第 3 章：入門與設定	17
跨平台安裝：macOS、Linux、Windows、WSL、Homebrew、WinGet	17

認證與帳號設定	17
您的首次會話：從零到程式碼的逐步操作	17
許可權模式：預設、自動接受、Plan 與 Auto 模式	18
IDE 整合：VS Code、JetBrains、遠端控制、Web 介面	18
第 4 章：Claude Code 的提示工程	19
Claude 提示的藝術：具體性、情境與驗證	19
CLAUDE.md 作為持久指令層	19
少次範例、XML 結構化與角色分配	19
「訪問我」模式：讓 Claude 在編碼前提問	19
提示鏈接與跨會話狀態管理	19
常見失敗模式與如何避免它們	20
對抗性測試：壓力測試 Claude Code 的輸出	20
第 5 章：真實開發工作流	21
從零建構全端應用程式（實作專案）	21
重構遺留程式碼庫：Stripe 的 Scala 到 Java 遷移案例研究	22
除調複雜 Bug：逐步操作	22
大規模程式碼遷移：Wiz 的 Python 到 Go 遷移	22
熟悉不熟悉程式碼庫的入職培訓	23
第 6 章：測試、驗證與品質保證	24
給 Claude 一個可對照驗證的東西	24
測試生成與自我修復測試	24
Outcomes：自動化評分標準評級	24
以平行會話的撰寫者/審查者模式	24
程式碼審查技能與對抗性審查策略	24
CI/CD 整合：GitHub Actions、GitLab CI 與自動化 PR	25
第 7 章：自動化與擴展	26
無頭模式：用於腳本和管道的非互動式 Claude Code	26
扇出模式：平行處理數千個檔案	26
Claude Code Routines：排程任務和事件驅動工作流	27
Pre-commit Hooks、Linting 與自動化程式碼品質閘門	27
大規模成本管理與 Token 預算	27
第 8 章：進階代理式模式	28
子代理：授權調查與驗證	28
技能：建立可重用的領域知識和工作流	28
MCP 伺服器：連接外部系統（GitHub、資料庫、Jira）	28

Hooks：跨會話自動化策略強制執行	28
多代理協調：主導代理、專家子代理、共用檔案系統	28
Claude Managed Agents 與「Dreaming」功能	29
第 9 章：部署與生產就緒	31
將 Claude Code 建構的應用程式部署到雲端平台	31
環境配置與機密管理	31
監控與可觀察性：OpenTelemetry、使用分析、Token 追蹤	31
效能最佳化：提示快取、模型選擇、情境效率	31
從原型到生產：與樂天的案例研究	31
第 10 章：團隊協作與企業部署	33
大規模 Claude Code：Stripe 的 1,370 名工程師部署	33
受控設定、企業安全與合規	33
共用 CLAUDE.md、技能函式庫與外掛分發	34
將 Claude Code 整合到現有團隊工作流	34
第 11 章：安全深入探討	35
已記錄的漏洞：CVE-2025-59536 與 CVE-2026-21852	35
提示注入與惡意檔案內容	35
情境載入中的機密暴露	36
—dangerously-skip-permissions 標誌：何時如何安全使用	36
沙箱化：Docker、Dev 容器與 VM 隔離	36
企業安全架構：MDM、LLM 閘道、代理監控	36
第 12 章：代理式編碼的未來	37
Anthropic 的路線圖：Dreaming、Outcomes、多代理協調	37
關於專業知識的研究：為何領域知識勝過編碼背景	37
塑造 2026 年及以後代理式編碼的八大趨勢	37
軟體工程師角色的變化	37
這對工作和知識經濟的未來意味著什麼	38
第 13 章：結論：成為代理式工程師	39
綜合從新手到大師的旅程	39
軟體開發的新思維模型	39
建立您的 Claude Code 實踐：個人行動計畫	39
展望未來：接下來的五年帶我們去何處	39
參考文獻	40

2026 年代理式軟體開發完全指南

精通 Claude Code、多代理工作流與企業部署的綜合手冊

第一版，2026 年 7 月

本書涵蓋 Claude Code 至版本 2.1.131，以及 Claude 模型至 Opus 4.8、Fable 5、Sonnet 4.6 和 Haiku 4.5。所有定價、基準測試與功能描述皆反映截至 2026 年 7 月之平台狀態。

關於本書：這是一本關於 Claude Code 的綜合指南——Anthropic 的代理式編碼系統，它已根本性地改變了軟體的開發方式。無論您是剛接觸 AI 輔助程式設計的新手開發者，還是希望精通多代理工作流的資深工程師，本書將從首次安裝帶您一路走到企業級部署。內容涵蓋 Claude Code 的架構、提示工程技巧、來自 Stripe 與 Wiz 的真實案例研究、CI/CD 自動化、安全最佳實踐，以及代理式編碼的未來。每一章都包含實用範例、實作專案，以及基於對數十萬個真實 Claude Code 會話研究的可行指導。

前言：不再敲鍵盤的工程師

2025 年底，Anthropic 的 Claude Code 創始人兼負責人 Boris Cherny 提出了一個觀察——若放在僅僅三年前聽來猶如科幻小說。自去年十一月以來，他未曾手動寫過一行程式碼。取而代之的是，他描述自己想要的東西，Claude Code 便想辦法將其建構出來、執行測試、修復失敗、並推送合併請求。他同時運行五個本地終端會話和五到十個遠端執行個體，每天推送十到三十個合併請求 [1]。

工具的創作者在描述打造該工具的公司日常運作的方式，而這僅是一個更大故事中單一數據點。

截至 2026 年 5 月，Anthropic 自身生產程式碼庫中超過百分之八十的程式碼由 Claude 撰寫，較 Claude Code 於 2025 年 2 月以研究預覽形式推出前的個位數低占比大幅提升 [7]。Anthropic 的工程師不會從零開始手打實作方案。他們描述想要的東西，Claude Code 想辦法建構、執行測試、修復失敗，並推送合併請求。人類工程師則審查輸出結果、做出架構決策，然後繼續處理下一個問題 [8]。

本書將教您的內容

本書是關於 Claude Code 的綜合指南——Anthropic 的代理式編碼系統，它已根本性地改變了軟體的開發方式。無論您是剛接觸 AI 輔助程式設計的新手開發者，還是希望精通多代理工作流的資深工程師，本書將從首次安裝帶您一路走到企業級部署。內容涵蓋 Claude Code 的架構、提示工程技巧、來自 Stripe、Wiz 和樂天的真實案例研究、CI/CD 自動化、安全最佳實踐，以及代理式編碼的未來。每一章都包含實用範例、實作專案，以及基於對數十萬個真實 Claude Code 會話研究的可行指導。

讀完本書後，您將能夠：

- 在多個平台和 IDE 中安裝並配置 Claude Code
- 撰寫提示，首次嘗試即可穩定產出生產級別的程式碼品質
- 使用訪談模式和規格驅動開發，從零建構全端應用程式
- 建立測試與驗證循環，使 Claude 無需您的介入即可自我修正
- 透過無頭模式、扇出模式和排程常規來自動化重複性任務
- 設計多代理架構，包含主導代理、專家子代理與共用檔案系統
- 以企業規模部署 Claude Code，搭配受控設定、SSO 和治理政策
- 理解並減輕代理式編碼工具固有的安全風險，包括提示注入、機密洩露與供應鏈攻擊

本書適合誰閱讀

本書假設您對軟體開發具備基本熟悉度：您知道什麼是終端機、至少在某一種語言中寫過程式碼，並且了解版本控制、測試與部署等概念。您不需要在任何特定程式語言上成為專家。Claude Code 跨語言運作，而本書所教授的思維模型無論您的技術棧為何皆能遷移。

如果您是曾使用自動完成工具並覺得有幫助但有限的初級開發者，本書將向您展示超越自動完成之後的景象。如果您是正在為組織評估 Claude Code 的資深工程師或團隊負責人，本書將提供做出明智決策所需的技術深度與策略脈絡。如果您是產品經理、設計師、創始人或希望不逐行撰寫程式碼就能建構軟體的非技術專業人士，本書將向您展示 Claude Code 如何為您打開那扇門。

如何使用本書

本書分為三個部分，每一部分皆建立在前一部分的基礎上：

第一部分（第 1-3 章）：基礎。第 1 章和第 2 章建立概念框架。您將理解 Claude Code 在 AI 編碼工具演進中的位置、其代理式架構與自動完成及內聯助手的不同之處，以及其內部循環的底層運作方式。第 3 章將引導您完成安裝與首次會話。

第二部分（第 4-8 章）：核心技能。第 4 至 6 章教授您使用 Claude Code 的實用技能：提示工程、真實開發工作流與測試策略。第 7 和 8 章涵蓋自動化、擴展與進階代理模式。這正是您培養出區分高效會話與令人沮喪會話之肌肉記憶的地方。

第三部分（第 9-13 章）：生產環境與更遠之處。第 9 和 10 章處理部署、可觀察性與企業協作。第 11 章深入探討安全性。第 12 章展望代理式編碼的未來。第 13 章將所有內容綜合為成為高效代理式工程師的可行指導。

您可以從頭到尾閱讀本書，也可以跳至最符合您當前需求的章節。每一章皆設計為獨立深入的篇章，同時也在前面介紹的概念之上建立。

證據基礎

本書的證據基礎包含對 Anthropic 官方文件與工程部落格文章的廣泛分析、該公司針對約 40 萬個 Claude Code 會話（來自約 23.5 萬名使用者，時間範圍為 2025 年 10 月至 2026 年 4 月）的研究 [6]、Anthropic 於 claude.com/customers/ 發布的客戶案例研究、來自 Check Point Research 和其他安全公司的第三方技術分析，以及社群貢獻的工作流。無法獨立驗證的聲明皆標註其來源。證據不足或有爭議之處，文中亦會明言。

關於本書未涵蓋內容的說明

本書聚焦於 Claude Code 作為處理您自身程式碼庫的開發工具。它不涵蓋 Claude API 或使用 Claude API 建構應用程式，儘管本書中的概念將為您理解 Claude 底層模型的運作方式奠定堅實基礎。同時也不涵蓋 Claude Cowork（協作工作區）、Claude Design 或 Claude Security，這些是 Anthropic 產品組合中的獨立產品。

核心論點

本書的核心論點直截了當：Claude Code 代表從程式碼補全到代理式編碼的典範轉移。人類成為定義目標並審查結果的指揮者，而 Claude 負責執行。但此典範轉移伴隨著真實的風險：設定檔中的安全漏洞、情境管理挑戰、品質退化事件，以及企業級治理的需求。精通此工具不僅需要學習指令，更需要發展關於授權、驗證與自主代理協作的全新思維模型。

2026 年的局勢

Claude Code 並非唯一可用的代理式編碼工具。競爭局勢包含 GitHub Copilot、Cursor、OpenAI Codex、Google Antigravity 等，各具獨特策略與優勢。理解它們的差異對於選擇適合您工作流的工具至關重要。但 Claude Code 佔據獨特位置：它以終端機為優先卻跨越多個介面運行、它在專案層級而非檔案層級運作，並且在企業團隊中展現了最積極的採用率。

技術進展迅速。Claude Code 自推出以來已歷經重大變化，2026 年中期的主要功能發布在 5 月 6 日的「Code with Claude」會議上加入了 Dreaming（跨會話記憶策略）、Outcomes（自動化品質評級）以及多代理協調 [3]。模型陣容包含在 SWE-bench Verified 中得分百分之九十五點的 Claude Fable 5、得分百分之八十八點六的 Claude Opus 4.8，以及作為生產預設值的 Claude Sonnet 4.6 [2]。有效使用的核心原則、授權與驗證的思維模型，以及對 Claude Code 運作方式的架構理解，不因模型更新而改變。

隨後的章節提供了精通 Claude Code 的完整指南。

第 1 章：軟體開發中的代理式革命

從自動完成到代理：五年的演進

要理解 Claude Code 在局勢中的位置，追溯 AI 輔助編碼約五年的演進有助益。這個進步經歷了四個截然不同的階段，每一階段建立在前一階段之上並擴展了可能性的邊界。

第一階段涵蓋約 2019 至 2021 年，是基於規則的程式碼補全。像 TabNine 這樣的工具使用在開源程式碼上訓練的統計模型，在開發者輸入時建議下一行或函式。這些工具對样板程式碼和常見模式確實有幫助，但無法推理意圖、架構或多檔案變更。它們只補全行；它們不理解專案。

第二階段從 2021 年至約 2024 年中，是大語言模型（LLM）自動完成。GitHub Copilot 於 2021 年搭配 OpenAI 的 Codex 模型推出，代表了量子級的躍進。它不再使用統計預測，而是用神經語言模型產生程式碼建議。開發者可以標記一個函式並要求一組測試，或在註解中描述想要的東西而收到可運作的實作方案。這個體驗對許多開發者來說是顛覆性的，但它根本上仍是被動的。開發者輸入、模型回應，然後開發者接受或拒絕每個建議。

第三階段在 2024 年底出現並於 2025 年加速，引入了內聯助手和 AI 原生 IDE。像 Cursor（VS Code 的分支）和 Claude 自身整合的工具超越了單一檔案建議。它們能讀取整個專案情境、理解檔案之間如何關聯，並在多個檔案之間進行協調變更。開發者可以對程式碼庫提出問題並收到基於實際程式碼結構的答案。

Claude Code 作為通用可用工具於 2025 年 5 月 22 日搭配 Claude 4 推出，代表了第四階段：代理式編碼。與前代不同，Claude Code 不會等待建議被接受或拒絕。它接收目標、規劃一系列動作、使用真實開發工具執行它們（讀取檔案、編輯程式碼、運行測試、使用 git）、評估結果並調整方法。開發者定義目標並審查結果，而非指導每一步 [18]。

從自動完成到代理的轉變改變了人與機器之間的根本關係。有了自動完成，你做工作而工具提供建議協助。有了代理式系統，你將工作授權給能獨立規劃、執行和迭代的強大合作者。

Anthropic 自身的數據反映了這種演進。對約 40 萬個 Claude Code 會話（2025 年 10 月至 2026 年 4 月）的隱私保護分析顯示，在這七個月內工作的組成發生了戲劇性變化 [6]。花在修復破損程式碼上的會話占比從百分之三十三降至百分之十九，而操

作軟體則從百分之十四增至百分之二十一。撰寫新程式碼和分析數據大致翻倍，從約百分之十增至百分之二十。任務本身也變得更有價值：平均會話的估計經濟價值上升了百分之二十七。

隨著代理能力增強，開發者不再將它們用於小修補，而是開始用於端到端開發、部署和運維。這個工具不只再是撰寫程式碼；它正在運行管道、部署應用程式、分析數據並產生文件。

是什麼讓 Claude Code 不同：終端機優先的代理式架構

2026 年的 AI 編碼工具局勢已圍繞幾個主導平台整合，各具獨特策略和優勢集合。理解它們的差異對於選擇適合您工作流的工具至關重要。

GitHub Copilot 由 Microsoft 和 GitHub 維護，是最廣泛相容的選項。它整合了十個以上 IDE，包含 VS Code、JetBrains、Neovim、Xcode、Eclipse、Zed 和 SQL Server Management Studio。它的優勢在於分發與生態系整合。Copilot 支援跨 OpenAI (GPT-5.1 至 5.4)、Anthropic (Haiku 4.5、Sonnet 4.6、Opus 4.6)、Google (Gemini 3 Pro 和 Flash) 以及 xAI Grok Code 的模型彈性。企業層級新增自訂知識庫、合併請求摘要、細粒度管理控制和 SAML 單一登入。對個別開發者每月十美元、對商業層級團隊每席位每月十九美元的定價，使其仍是基本 AI 編碼協助最具性價比的選項。它的弱點是與 GitHub 生態系更緊密耦合以及有限的自主多步驟執行能力。

Cursor 由 Anysphere 開發，是完全 AI 原生的 IDE，建構於 VS Code 分支之上。它的優勢在於整合深度。每個編輯層都是 AI 感知的：tab 補全、用於多檔案代理式編輯的 Composer 2、背景雲端代理以及用於自動化合併請求審查的 Bugbot。Cursor 以每月二十美元（個別開發者）和每席位每月四十美元（團隊）提供最深的 AI 原生 IDE 體驗。它通過 SOC 2 Type 2 認證並進行年度滲透測試，支援 SCIM 席位管理和 SAML/OIDC 單一登入。主要限制是要求整個團隊遷移至 Cursor IDE，且其免費 Hobby 層級排除雲端代理和進階功能。

Claude Code 由 Anthropic 開發，採取根本不同的方法。它以終端機為優先卻跨越多個介面運行：CLI、VS Code 擴充套件、JetBrains 外掛、桌面應用程式、claude.ai/code 的 Web 介面、Slack 和 CI/CD 管道。其定義性特徵是代理式能力。Claude Code 在專案層級運作，讀取完整程式碼庫、跨越多個檔案規劃、執行變更、運行測試並在失敗上迭代。開發者定義目標並審查結果 [18]。

2026 年初的競爭局勢數據顯示了開發者偏好的顯著轉變。在 Pragmatic Engineer 對 15,000 名軟體工程師的 2026 年 2 月調查中，Claude Code 在開發者中獲得百分之四十六的「最受喜愛」評級，相比之下 Cursor 為百分之十九、GitHub Copilot 為百分之九 [50]。這不只是關於功能；它反映了日益增長的認知——代理能力比自動完成更重要。

下表總結了關鍵差異：

維度	Claude Code	GitHub Copilot	Cursor
主要介面	終端機 + IDE 擴充套件	IDE 外掛 (10+ 編輯器)	AI 原生 IDE (VS Code 分支)
代理式能力	完整多步驟自主性	內聯建議 + 工作區	多檔案編輯、雲端代理
模型選項	僅 Anthropic 模型	OpenAI、Anthropic、Google、xAI	多個模型供應商
外部工具整合	MCP 伺服器 (開放標準)	GitHub 生態系擴充套件	Cursor 專屬擴充套件
Slack 整合	原生非同步編碼 workflow	有限	無
企業安全	HIPAA 就緒、SSO、RBAC	SAML SSO、IP 賠償	SOC 2 Type 2、零數據保留
個人定價	\$20/月 (Pro)	\$10/月 (Pro)	\$20/月 (Pro)

在這些工具之間選擇不一定是非此即彼。許多大型工程團隊為多 IDE 廣泛性運行 Copilot，同時部署 Claude Code 或 Cursor 處理需要深度程式碼庫推理和自主執行的高槓桿複雜任務。關鍵因素是將工具匹配到任務：簡單自動完成從廣泛 IDE 相容性中受益，而複雜多檔案重構和端到端功能開發則從 Claude Code 的代理式架構中受益。

Boris Cherny 的工作流：創始人實際如何使用 Claude Code

關於 Claude Code 應如何使用的最佳證據來自其創始人。Boris Cherny 此前在 Meta 工作，負責整個公司的程式碼品質，現領導 Anthropic 的 Claude Code 團隊。他的職業軌跡從程式碼品質工程到建構 AI 編碼工具，使他對開發者需求有獨特見解 [49]。自 2025 年底以來，他自身百分之百的程式碼皆由 Claude Code 撰寫。他每天推送十到三十個合併請求，同時運行五個本地終端會話和五到十個遠端執行個體 [1]。

Cherny 的工作流是系統性的，而非隨意的。他避免自訂，依賴 Claude Code 開箱即用。每個本地會話使用獨立的 git checkout 而非分支或工作樹，防止平行會話之間的衝突。遠端會話透過 & 運算符置於背景並以 --teleport 轉移，儘管他指出十到百分之二十會因未預見的情境而被放棄 [2,29]。

在編碼方面，Cherny 偏好 Opus 4.5（現已被 Opus 4.8 取代）並啟用思考功能，指出相比 Sonnet 其品質、可靠性和工具使用能力更優越。他指出雖然 Sonnet 可能每步更快，但整體吞吐量以 Opus 更快，因為首次嘗試更常成功 [2,29,47]。

知識保留透過 git 中由團隊維護的 CLAUDE.md 文件制度化。此文件記錄錯誤、風格慣例、設計指南和 PR 模板以迭代改進 AI。Cherny 以 @.claude 標記同事的合併請求以注入新學習，將檔案維持在約 2.5k tokens [2]。這是個關鍵見解：CLAUDE.md 不是靜態設定檔而是活生生的文件，透過團隊集體經驗隨時間不斷改進。

他的流程始於使用 Plan 模式的迭代規劃階段，然後切換至自動接受編輯，通常首次嘗試即成功。像提交和 PR 這樣的日常操作透過存儲在 .claude/commands/ 中的斜線指令自動化，最小化明確提示。一個 /commit-push-pr 指令每天執行數十次，透過內聯預先計算 git 狀態來減少模型延遲。為避免格式不一致導致的 CI 失敗，他配置了一個 PostToolUse hook，在 WriteEdit 匹配時自動運行 `bun run format || true` [2,46]。

安全性透過 /permissions 授予特定安全的 bash 指令（例如 `bun run build:*、cc:*`）來處理，大幅消除了對 `--dangerously-skip-permissions` 的需求。危險標誌僅保留給沙箱化的長運行任務以防止重複中斷。Cherny 強調提供驗證回饋循環（如運行測試套件或瀏覽器自動化）至關重要，使最終輸出品質提升兩到三倍。他指出 Claude 使用 Claude Chrome 擴充套件測試每一個合併到 `claude.ai/code` 的變更 [2,48]。

這種結構化方法確保程式碼到達審查時已經過打磨，讓工程師能專注於方向引導和架構監督。模式很清楚：先規劃，然後自動化常規步驟，嚴謹驗證，讓 Claude 處理執行。

勞動分工：人類為指揮者、代理為建構者

Anthropic 對 40 萬個 Claude Code 會話研究中最重要發現之一，是人與代理之間出現的清晰勞動分工 [6]。在典型會話中，人類做出約百分之七十的規劃決策（做什麼、採取何種方法、什麼算完成），而 Claude 做出約百分之八十的執行決策（更改哪些檔案、撰寫什麼程式碼、運行哪些指令）。

這種分工不是偶然的。它反映了人類偏好和代理能力兩者。人類自然被吸引到規劃和決策，因為那是領域專業知識最重要的地方。一位指示 Claude 建構對帳腳本的會計師帶來了業務規則的知識、月末結算時的邊緣案例，以及定義成功的驗證標準。Claude 帶來了程式碼結構、函式庫 API、測試框架和除錯技術的知識。

數據在各職業中一致地顯示這種分工。電腦與數學職業（最大的使用者群體）在整體約百分之三十的會話中達到經驗證的成功 [6]。其他職業的使用者約百分之二十六次達到經驗證的成功。在產生程式碼的會話中，這些數字分別為百分之三十四和百分之二十九。十大非軟體職業群體中的每一個在成功率上都落在與軟體工程師

七個百分點之內。管理職業實際上在經驗證成功上排名最高，略高於軟體工程職業。

一個人帶入會話的領域專業知識越多，Claude 每條指令做的工作就越多。在典型新手會話中，每個提示觸發約五個 Claude 動作和約六百字的輸出。專家會話觸發超過兩倍長的動作鏈（12 個動作），承載五倍的輸出（3,200 字）[6]。此差距在每種工作和每個任務價值區間中都出現。

協作的形態影響你如何看待 Claude Code。它不是開發者的替代品。它是領域專業知識的增強因子。一位提出第一個 Rust 問題的高階工程師對 Rust 而言是初學者。一位從未使用過 Python 但告訴 Claude 一個 Python 腳本必須執行哪些對帳規則、並捕捉它在月末結算時處理不當的邊緣案例的會計師，在那個任務上是專家 [6]。這個工具獎勵的是對問題領域的理解，而非程式語言的熟練度。

訓練應著重於培養問題定義的清晰度、驗證標準和領域知識，而非學習新語法或 API 參考。最有效的 Claude Code 使用者不一定是最好的編碼者；他們是能闡述需要建構什麼、為何重要、以及如何驗證其運作的人。

2026 年 AI 編碼代理的局勢

Claude Code 並非唯一可用的代理式編碼工具。競爭局勢包含幾個值得注意的參與者：

OpenAI Codex，作為驅動早期 GitHub Copilot 功能的原始 Codex 模型的繼任者，已演進為獨立代理系統。它運作方式類似 Claude Code，具備多檔案編輯和自主執行能力。OpenAI 的優勢在於其龐大的開發者基礎和與 ChatGPT Workspace 的整合。弱點是企業使用的定價模型較為不透明。

Google Antigravity 2.0 於 2026 年初宣布，利用 Google 的 Gemini 模型處理編碼任務。它與 Google Cloud 和 Vertex AI 深度整合，使其成為已投資於 Google 生態系團隊的自然選擇。它在 SWE-bench 上的基準測試表現落後於 Claude Fable 5 和 OpenAI Codex [8]。

Devin，來自 Cognition Labs（被 Amazon 收購）的原始自主編碼代理，開創了能規劃並執行完整軟體任務的代理概念。雖然它建立了這個類別，但截至 2026 年中，其能力在多數基準測試上已被 Claude Code 超越。

Amazon Q Developer 建構於 AWS 生態系中，提供與 AWS 服務深度整合的代理式編碼能力。對於基礎設施高度依賴 AWS 的團隊而言是強有力的選擇。

這個領域的競爭動態正在快速變化。曾經的模型競賽（哪個 LLM 在編碼上最聰明）已變成馬具競賽（哪個平台最能協調模型能力、外部工具和人類監督）。Anthropic 在 2026 年 5 月「Code with Claude」活動中的策略強化了這一點：沒有宣

布新模型。取而代之的是推出了五個基礎設施功能：Dreaming、Outcomes、多代理協調、包含十個預建代理的 Claude Finance，以及 Add-ins [3,38]。訊息很清楚。生產代理系統的瓶頸不是模型能力。而是模型周圍的腳手架。

為何是現在？市場採用與向代理式工作流的轉變

2026 年代理式編碼採用的加速由幾個匯聚因素驅動。

首先，模型能力已達到自主代理可靠處理多步驟任務的門檻。Claude Fable 5 在 SWE-bench Verified 上得分百分之九十五 [22]，而 Claude Opus 4.8（2026 年 5 月 28 日發布）提供顯著更好的代理式判斷，被忽略的程式碼缺陷約為前代的四分之一 [28]。這些代表了可靠性的質性轉變。

其次，大規模運行代理的基礎設施已成熟。提示快取在 Claude Code 會話中達到百分之九十二的前綴重用率，將成本降低約百分之八十一，使持續代理使用在經濟上可行 [26]。模型情境協議（MCP），現由 Agentic AI Foundation 管理，已由 Anthropic、OpenAI Agents SDK、Cursor、Cloudflare Agents 和 Microsoft 實現 [17]，為代理到工具整合建立了開放標準。

第三，市場採用已跨越臨界點。具有編碼代理活動的 GitHub 專案占比自 2025 年底以來已增加一倍以上 [6]。Claude Code 至 2026 年 2 月年化收入超過 25 億美元 [8]。在 Anthropic 自身，百分之八十的新生產程式碼現由 Claude Code 撰寫 [7]。社群動量伴隨企業採用同步增長：GitHub 上的 Claude Code 儲存庫至 2026 年中已超過 12.7 萬顆星 [25]。

第四，經濟誘因令人信服。來自 Stripe、Wiz、Ramp 和樂天的案例研究展示了難以忽視的生產力增益。一個能將五萬行 Python 函式庫遷移到 Go 僅用二十小時而非兩到三個月的團隊，已根本性地改變了其容量限制。一家將功能交付從二十四天縮減到五天的公司，將其產品開發週期壓縮了約百分之八十。

向代理式編碼的轉變不是暫時趨勢或炒作週期。它代表了軟體建構方式的結構性變化，堪比從打孔卡到鍵盤或從組合語言到高階語言的過渡。工具正在改變、工作流正在改變，最重要的技能也在改變。

問題不再是代理式編碼是否會成為主流，而是團隊如何有效地適應它。本書旨在提供回答這個問題所需的技術深度與策略脈絡。

批判性分析：代理式工具的權衡與限制

在深入探討如何有效使用 Claude Code 之前，值得檢視代理式編碼工具做不到的事情，以及它們在傳統開發工作流之外引入了哪些新風險。理解這些限制能防止昂貴失誤。

情境輪換是真實的。即使有 1M token 視窗，累積了大量工具輸出、錯誤日誌和多檔案 diff 的會話最終會觸發自動壓縮。當壓縮發生時，Claude 可能丟失早期指令或會話前半段建立的微妙情境。Anthropic 2026 年 4 月的品質退化事件生動地展示了這一點：一個在每輪清除推理歷史的快取 bug 導致數千個會話出現遺忘、重複和奇怪的工具選擇 [12]。2026 年 4 月 23 日的事務分析揭示，三項獨立產品層變更導致了 2026 年 3 月 4 日至 4 月 20 日之間的退化：預設推理努力從高調整到中（3 月 4 日）、一個在每輪而非每個閒置小時丟棄推理的快取最佳化 bug（3 月 26 日），以及限制工具調用文字少於二十五字的系統提示冗長度降低（4 月 16 日）[12,33,34]。三者皆於 4 月 20 日在版本 2.1.116 中修復。教訓不是 Claude Code 不可靠；而是即使 LLM 周圍馬具中的單一微妙變更也能級聯為大規模用戶群中可測量的品質退化。

Claude Code 無法可靠處理未經測試的任務。SWE-bench 測量模型修復開源儲存庫中已知 bug 的能力。它不測量模型從零建構不熟悉應用程式的能力。Claude Code 兩者都能做，但信心水平差異顯著。Table 5 百分之九十五的 SWE-bench Verified 分數反映的是在具有已知測試案例和明確失敗訊號的特定基準上的表現。從高階描述建構新穎應用程式涉及模糊的需求、架構決策以及任何基準都無法捕捉的權衡。

理解基準表現與真實能力之間的差距需要了解 SWE-bench 如何運作。每個 SWE-bench 任務是來自真實開源專案的 GitHub issue，搭配該 issue 提出時的儲存庫狀態和一組定義修復是否正確的現有測試。模型接收 issue 描述、探索儲存庫，並嘗試撰寫使現有測試通過的程式碼變更。這個基準有三個使其比真實開發更容易的特性：問題是明確定義的（issue 描述了特定 bug）、成功標準是確定性的（測試要麼通過要麼失敗），且程式碼庫是熟悉的（模型在包含這些儲存庫的開源程式碼上訓練過）。

真實開發缺乏這三個特性。當你要求 Claude「建構任務管理 API」時，需求是模糊的（任務應有哪些欄位？什麼認證模型？）、成功標準是主觀的（API 感覺對嗎？錯誤處理足夠嗎？），且程式碼庫可能是完全新穎的（Claude 從未見過您的專案）。百分之九十五的 SWE-bench 分數不意味著 Claude 在真實開發任務上百分之九十五次成功。它意味著 Claude 在一類具有確定性測試套件的明確定義 bug 修復任務上百分之九十五次成功。

此區別對架構決策很重要。如果您使用 Claude Code 修復具有綜合測試套件的現有程式碼庫中的 bug，您可以對輸出有高度信心。如果您使用 Claude Code 從零設計和實作新系統，您需要顯著更多的人類監督、驗證和迭代精煉。基準分數是原始編碼能力的有用訊號，但不是新穎任務上真實成功的可靠預測因子。

Claude Code 在多步驟鏈中可能產生幻覺。單一動作可能產生正確輸出，但錯誤在十到二十個動作的鏈中會累積。第三步應該讀取的檔案可能在第五步被編輯過，而 Claude 對程式碼庫狀態的理解變得陳舊。這就是為何每一步的驗證至關重要。Huang 等人（2026）的研究測試了七個廣泛使用的 MCP 客戶端包含 Claude Code，發現工具中毒攻擊能透過來自連接外部系統的返回數據注入惡意指令 [14]。當 Claude

讀取包含嵌入指令的 GitHub README、資料庫查詢結果或 Slack 訊息時，那些指令成為其情境的一部分並可能操縱其行為。

Claude Code 無法對業務需求做出決策。它能實作您要求的內容，但無法決定功能是否應存在、哪個 API 設計最優，或哪些權衡可接受。這些是人類的判斷。對 40 萬個會話的研究確認了這一點：人類做出約百分之七十的規劃決策而 Claude 處理約百分之八十的執行決策 [6]。

Claude Code 為您的開發環境引入真實的安全風險。正如 Check Point Research 所記錄的，CVE-2025-59536 允許透過惡意專案設定檔在任何信任對話框出現前進行遠端程式碼執行，而 CVE-2026-21852 允許透過 ANTHROPIC_BASE_URL 操縱實現 API 金鑰竊取 [11]。這些是已記錄的漏洞，影響了真實部署並展示了傳統上被視為被動元數據的設定檔如何成為可執行的攻擊向量。

在 Claude Code 之外，更廣泛的 MCP 生態系面臨系統性安全挑戰。Ox Security (2026 年 4 月) 的漏洞警告指出，IDE 和編碼助手包含 Windsurf、Claude Code、Cursor、Gemini CLI 和 GitHub Copilot 皆易受透過其 MCP JSON 設定檔的命令注入攻擊 [17]。當時唯一發布的 CVE 是針對 Windsurf 的，但攻擊模式適用於整個生態系。NIST 將提示注入描述為「生成式 AI 最大的安全缺陷」，OWASP 在其 LLM 應用程式 Top 10 中將其列為第一號漏洞 [35]。

Claude Code 有使用限制和運算約束。2026 年初，許多 Pro 方案使用者報告在使用 Sonnet 僅十到十五分鐘內即觸及使用上限，而 Opus 在開始後不久即無法使用 [27]。Anthropic 於 2026 年 5 月為回應運算瓶頸將使用限制加倍，但根本約束仍然存在：Claude Code 是共用資源，大量使用最終會觸及速率限制 [39]。Pro 方案包含每月二十美元的代理信用點、Max 5x 包含一百美元、Max 20x 包含二百美元 [2]。對於持續的開發工作，多數團隊發現升級至 Max 5x 或更高是避免中斷的必要條件。

Claude Code 無法取代人類程式碼審查。每個 AI 產生的變更應經過與開發者撰寫程式碼相同的審查流程。每個涉及安全關鍵檔案的 AI 產生變更應要求兩位審查者。代理是合作者，而非自主推送者 [20]。這個原則由 Stripe 的經驗強化：領導 Stripe 開發者基礎設施團隊的 Scott MacVicar 強調，將 AI 視為替代品而非合作者是採用中最大的挑戰 [8]。

供應商鎖定與生態系依賴。Claude Code 設計為搭配 Anthropic 模型運作最佳，並與 Claude 平台緊密整合。雖然它使用像 MCP 這樣的開放標準，但從 Claude Code 遷移到另一個代理或回到傳統開發帶有真實成本：已學習的工作流、CLAUDE.md 文件、自訂技能、hooks 和 MCP 伺服器配置皆專屬於 Claude Code。團隊應評估生產力增益是否超過鎖定風險，特別是對具有漫長生命週期的專案而言。

這些限制本身不是 Claude Code 的弱點。它們是任何將工作授權給基於 LLM 的代理系統的系統的特性。理解它們是有效且安全使用 Claude Code 的第一步。

第 2 章：Claude Code 的解剖

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

代理循環：收集、行動、驗證

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

追蹤代理循環：完整的終端機逐步操作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

是什麼使代理循環與自動完成不同

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

中斷和重新導向循環

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

幕後模型：Sonnet、Opus、Fable 與模型選擇

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

選擇模型：實用決策框架

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

工具定義帶來的 Token 開銷

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Token 管理數學：理解您的花費

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

內建工具：檔案操作、搜尋、執行、Web、程式碼智慧

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

.claude 目錄：配置、CLAUDE.md 與記憶

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

情境視窗：如何填充、自動壓縮與 Token 管理

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

情境如何填充：Token 預算逐步操作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

自動壓縮如何底層運作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

控制什麼在壓縮中倖存

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

有效管理情境的策略

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 3 章：入門與設定

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

跨平台安裝：macOS、Linux、Windows、WSL、Homebrew、WinGet

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Claude Code 桌面應用程式

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

排除常見安裝問題

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

認證與帳號設定

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

您的首次會話：從零到程式碼的逐步操作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

終端會話追蹤：您的首次互動

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

逐步配置：.claudeignore、settings.json 與許可權模式

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

許可權模式：預設、自動接受、Plan 與 Auto 模式

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

IDE 整合：VS Code、JetBrains、遠端控制、Web 介面

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 4 章：Claude Code 的提示工程

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Claude 提示的藝術：具體性、情境與驗證

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

CLAUDE.md 作為持久指令層

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

少次範例、XML 結構化與角色分配

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

為何這些技巧有效：提示工程背後的機制

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

「訪問我」模式：讓 Claude 在編碼前提問

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

提示鏈接與跨會話狀態管理

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

常見失敗模式與如何避免它們

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

對抗性測試：壓力測試 Claude Code 的輸出

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 5 章：真實開發工作流

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

從零建構全端應用程式（實作專案）

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

階段 1：探索與規格

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

階段 2：實作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

階段 3：前端整合

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

階段 4：除錯真實 Bug

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

階段 5：對抗性測試

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

處理實作中的情境壓縮

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

此工作流教您的內容

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

重構遺留程式碼庫：Stripe 的 Scala 到 Java 遷移案例研究

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Stripe 案例研究：深入探討

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

大規模遷移工作流

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

審查瓶頸

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

除調複雜 Bug：逐步操作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

大規模程式碼遷移：Wiz 的 Python 到 Go 遷移

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

熟悉不熟悉程式碼庫的入職培訓

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 6 章：測試、驗證與品質保證

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

給 Claude 一個可對照驗證的東西

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

測試生成與自我修復測試

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Outcomes：自動化評分標準評級

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

撰寫 Outcomes 評分標準：逐步操作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

以平行會話的撰寫者/審查者模式

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

程式碼審查技能與對抗性審查策略

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

CI/CD 整合：GitHub Actions、GitLab CI 與自動化 PR

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

生產級 CI/CD 配置

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

帶有條件執行的自動化 PR 審查

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

自動化合併請求建立

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

CI/CD 中的安全考量

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

合規與審計需求

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 7 章：自動化與擴展

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

無頭模式：用於腳本和管道的非互動式 Claude Code

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

扇出模式：平行處理數千個檔案

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

順序扇出（簡單）

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

平行扇出（高通量）

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

帶有結果聚合的扇出

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

帶有併發控制和退避的扇出

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

扇出模式的成本分析

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Claude Code Routines：排程任務和事件驅動工作流

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Pre-commit Hooks、Linting 與自動化程式碼品質閘門

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

大規模成本管理與 Token 預算

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 8 章：進階代理式模式

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

子代理：授權調查與驗證

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

子代理 Token 經濟學

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

技能：建立可重用的領域知識和工作流

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

MCP 伺服器：連接外部系統（GitHub、資料庫、Jira）

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Hooks：跨會話自動化策略強制執行

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

多代理協調：主導代理、專家子代理、共用檔案系統

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

配置多代理協調：逐步操作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

解決多代理工作流程中的合併衝突

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

從多代理運行解析可觀察性數據

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Claude Managed Agents 與「Dreaming」功能

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Dreaming：自我改進的代理

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Outcomes：自動化品質評級

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Claude Managed Agents 定價與部署

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

多代理協調：模式與失敗模式

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Claude Finance 與 Add-ins

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 9 章：部署與生產就緒

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

將 Claude Code 建構的應用程式部署到雲端平台

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

環境配置與機密管理

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

監控與可觀察性：OpenTelemetry、使用分析、Token 追蹤

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

效能最佳化：提示快取、模型選擇、情境效率

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

從原型到生產：與樂天的案例研究

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

樂天的「AI-nization」策略

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

技術實作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

大規模平行開發

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

未來軌跡：環境代理

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 10 章：團隊協作與企業部署

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

大規模 Claude Code：Stripe 的 1,370 名工程師部署

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

受控設定、企業安全與合規

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

識別與 API 金鑰治理

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

集中式流量路由

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

MCP 工具治理

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

本地沙箱與許可權強制執行

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

端到端企業部署逐步操作

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

合規框架

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

基於角色的存取控制

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

共用 CLAUDE.md、技能函式庫與外掛分發

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

將 Claude Code 整合到現有團隊工作流

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 11 章：安全深入探討

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

已記錄的漏洞：CVE-2025-59536 與 CVE-2026-21852

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

提示注入與惡意檔案內容

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

透過檔案的提示注入

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

CI/CD 提示注入

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

隱私防護措施

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

額外防護措施

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

NIST 和 OWASP 分類

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

情境載入中的機密暴露

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

—dangerously-skip-permissions 標誌：何時如何安全使用

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

沙箱化：Docker、Dev 容器與 VM 隔離

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

企業安全架構：MDM、LLM 闡道、代理監控

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 12 章：代理式編碼的未來

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

Anthropic 的路線圖：Dreaming、Outcomes、多代理協調

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

關於專業知識的研究：為何領域知識勝過編碼背景

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

塑造 2026 年及以後代理式編碼的八大趨勢

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

供應商鎖定與生態系依賴

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

軟體工程師角色的變化

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

法律與倫理局勢

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

這對工作和知識經濟的未來意味著什麼

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

第 13 章：結論：成為代理式工程師

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

綜合從新手到大師的旅程

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

軟體開發的新思維模型

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

建立您的 Claude Code 實踐：個人行動計畫

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

展望未來：接下來的五年帶我們去何處

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>

參考文獻

這部分內容並未包含在樣本書中。您可以在 Leanpub 上購買這本書，網址是 <https://leanpub.com/claudecodetw>