

對本書的讚譽

《敏捷開發的藝術》的第二版無疑是一本傑作。它把現代軟體交付的方法濃縮成一本簡短、易讀、且令人愉悅的書。對於剛接觸迭代式交付的人來說，它為普遍的實務做法提供了一個絕佳的概述。對於那些迷失在過度設計的「規模化敏捷」流程裡的人來說，它提供了逃離那些地獄的好點子。第一版在二十年前對我的職涯產生了巨大影響，而我相信第二版同樣會幫助數以百萬的開發者改善他們交付軟體的方式。

—Gojko Adzic,

著有《*Running Serverless, Impact Mapping, Specification by Example*》

本書涵括了從程式碼到交付產品的所有一切。它讓數十年來，來之不易的知識變得易懂易讀——對於任何與軟體開發團隊合作或從事其中的你來說，這是一本你一定要擁有的書。

—Avi Kessner, Forter 主任工程師

這本書將會一直待在我最容易拿到的書架上。

—Krishna Kumar,

Exathink Research 創辦人暨執行長

本書的第一版讓我深深地著迷於其中的內容，使得我一直將它留在書架上作為參考之用。第二版不僅保留了原來的真知灼見，並且增加了更多過去十年的見解。

—Benjamin Muskalla, Github 資深工程師

這是我曾經讀過關於敏捷軟體開發最全面的一本書籍。無論哪種技術組合、團隊規模、或產業領域，非常務實的內容加上強而有力的範例能讓你輕鬆地應用到任何軟體開發的專案上。肯定是工作場合裡，你應該隨身攜帶的瑰寶。

—Luiza Nunes, Thoughtworks 計畫經理

這是一本我最喜歡且關於敏捷開發的書，而且涵蓋了技術和管理主題。我會在我的課程裡介紹其中的內容，並且總是將它推薦給我的客戶。

—Nicolás Paez,
Universidad de Buenos Aires 教授和軟體工程師

Jim 透過易於閱讀且把概念與實務做法連結起來的章節內容，詳盡地介紹了基於經驗的敏捷軟體開發方法。

—Ken Pugh, 首席顧問,
著有《*Prefactoring: Extreme Abstraction, Extreme Separation, Extreme Readability*》

在數以千計的敏捷書籍裡，哪本才是你應該閱讀的？我建議你考慮本書。James 經歷了早期的敏捷，並且了解自己所面對的事情。本書沒有我們產業裡那些胡謔的內容（那些無所不在又毫無意義的「敏捷」），並且提供了一個透徹且整體的方法。這個方法雖然並不是簡單快速就能奏效，但卻值得採用。我喜歡《敏捷開發的藝術》，它是一本有個性的書！

—Bas Vodde, LeSS 的共同創作者

James Shore 運用新的工具、技術和過去十年的經驗，全面地更新了《敏捷開發的藝術》一書。本書中寶貴的內容將幫助進化你的工作方式，成為一種真正敏捷且有效的方法。

—Bill Wake, XP123, LLC

序

問：要如何才能登上卡內基音樂廳的表演台？

答：練習、練習、練習！

我想要幫助你掌握敏捷開發的藝術。

敏捷開發就像其他以團隊為基礎的軟體開發模式。它本質上就是一門關於人的藝術——受制於每個人本身的獨特性與彼此之間的互動。你必須學會時時刻刻評估無數的可能性，並且直觀地採取最好的行動方案，才能成功地掌握敏捷開發。

要怎樣才能學會如此困難的藝術？唯一的方式就是「練習」！

首先，本書是一本操作指南，它詳細地描述了一種實踐敏捷開發的方法。它不僅以 **Extreme Programming**（極限程式設計，XP）為基礎，同時也引入了 **Scrum**、看板、**DevOps**、精實軟體開發、精實創業等的想法和實務做法，進而成為一本實用的終極指南。它可以讓你成功地將敏捷開發帶入你的團隊和組織——或者幫助你了解敏捷對你來說並不是一個好選擇。

其次，本書旨在幫助你掌握敏捷開發的藝術。掌握敏捷性代表超越現有的實務做法。由於軟體開發極易受到所處情境的影響，以至於任何一種方法都無法簡單地完美匹配。另外，為了適於每個情境，所需調整的細節差異太多，因而沒有一本書能夠教你如何掌握它。精通來自於內化：來自於經驗和對每次選擇所造成的影響與變化的直觀理解。

我無法也不會嘗試告訴你，做怎樣的選擇對組織會產生怎樣的影響。你必須在每次的選擇裡，理出其中細微的差別並且體會，而這就是掌握藝術的唯一途徑。遵循實務的做法、看看會發生什麼事、思考這些實務做法為什麼有效或者沒有效，並且思索「什麼是相同的？」、「有什麼不同？」、「為什麼？」，然後一次又一次地試驗。

一開始，你或許會苦於了解如何應用每一個實務做法。這些做法讀起來簡單，但實際運用卻是相當困難。請持續練習，直到實際運用它們變得容易。

隨著你更加地熟稔敏捷，你會發現我所提供的一些建議，對你來說並沒有用。一開始，你會無法判斷問題是出在我提供的說明，還是出在你遵循它們的方式中。繼續練習，直到你確定為止。當你確定問題時，打破規則並且修改我所提供的指引，才能更好地適用於你所面對的具體情況。書中提及的每個實務做法都包括一個「替代方案和試驗」的章節，書中提及的每個實務做法都包括一個「替代方案和試驗」的章節，提供了用於探索的想法與點子。

總有一天，書中的規則將不再激起你任何的興趣。畢竟，敏捷並不是遵循某些規則。你會認為「它代表著簡單和回饋、溝通和信任」。「敏捷意味著交付價值——並且有勇氣在正確的時間做正確的事。」你將時時刻刻評估著無數的可能性，並直觀地選擇最佳行動方案。

當你掌握敏捷開發後，就算本書可能已經破舊不堪，也請你把這本書轉送給其他需要的人。這樣一來，他們也能因此掌握敏捷開發的藝術。

給那些講求實用的人

如果你並不想掌握所謂的「藝術」呢？如果你只是想開發出好的軟體？

不必擔心！這本書仍然適合你。我將多年的敏捷開發經驗提煉成一個單一的、明確定義的、全面的方法。

書中使用簡單明瞭的語言來解釋如何運用敏捷，並且包含了很多實用的技巧。同時也坦率地描述了我的方法何時行不通，以及行不通時，應該考慮哪些替代方案。第2章將協助你著手進行敏捷開發。

只討論一種方法會產生一個缺點，那就是沒有一種方法可以適合所有的人。我的建議可能不適合你的團隊或組織。請閱讀第4章和第5章，以便了解成功進行敏捷開發所需的各種條件，並且查看每個實務做法的「前提條件」，以便了解具體適用的情況。

但不要假定某個實務做法對你是沒有用的。本書中的一些做法或許並不直觀，也或許聽起來並不有趣，但大多數這類實務做法卻能與其他實務做法搭配得相當好。如果情況允許，

請按照書中所述的做法，實際採用幾個月，以便獲得一些在你的環境中運用它們的實際經驗，然後基於這些經驗改變它們。

20 多年來，我一直將這些想法付諸實踐。在合適的環境中，它們確實有效。與我嘗試過的任何其他軟體開發方法相比，敏捷開發更有趣，也更成功。快來加入我們的行列吧！

第二版的新增內容

《敏捷開發的藝術》第二版徹底重寫了第一版的內容。它保留了第一版務實的風格和大部分的實務做法，但由於這 14 年來敏捷實務做法的進展——更不用說我這 14 年來的經驗累積，幾乎所有的做法都已經被重新改寫。

我完全重組了本書的內容，讓團隊得以漸進式地採用敏捷，並且更好地反映實際使用敏捷的情況。第一版第三部分中所討論的原則和客製的方法已經被編排到各個實務做法裡，以便讓這些方法更容易讓讀者注意並且查閱。另外，我也為每個實務做法新增了進行試驗的相關建議。

值得注意的新主題有：

- 一個基於我與 Diana Larsen 創建的敏捷熟練度模型（Agile Fluency¹ Model），提供如何導入敏捷、並依據公司需求進行客製的深入指南。
- 我根據自己過去協助的大、小公司的經驗，所撰述的敏捷規模化新章節。
- 一個關於 DevOps 的新章節，包含有關維運和安全的全新內容。此外，本書各章節也受到了 DevOps 的啟發，而有對應的更新。
- 與遠距團隊合作的敏捷指南；許多新的實務做法、故事和想法；以及許多難以逐一詳述的改善與修改。

本書的讀者

本書適用於正在或希望未來能與敏捷團隊一起工作的每個人，包括了程式設計師、經理人、高階管理者、領域專家、測試人員、產品經理、專案經理、架構師、維運人員、資安工程師、設計師和商業分析師。敏捷團隊是由不同職能的人所組成的，而本書也恰恰體現了這個本質。

本書的第二部分到第四部分中的每個實務做法都能被單獨閱讀的方式來編寫。因此，讀者可以將本書作為參考工具書，也可以從頭到尾逐章詳讀。本書紙本版本頁面空白處的

1 「Agile Fluency」是由 Agile Fluency Project LLC 所註冊的商標。

「關聯 (Ally)」框與電子版本中的超連結，則是提供給你作為交叉參考之用。紙本書特別採用了易於挑選與翻閱排版的設計。讀者在翻閱本書的過程中，能於任何吸引你的文字標註處駐足，並且進一步地深入閱讀。

如果你是一名經理人或高階管理者，而正想了解敏捷能帶來哪些好處，或是想知道敏捷是否適合你的公司，請閱讀本書的第一部分。假如你是團隊的管理者，那請額外閱讀第 10 章「管理」一節，更建議你進一步閱讀第 10 章裡的其他實務做法。

如果你是一名團隊成員或經理人，並且有興趣將敏捷引入你的公司，或提升公司當前的敏捷實踐方式，請從頭到尾閱讀整本書。本書的第一部分將幫助你了解如何引入敏捷思想，其餘部分將幫助你了解如何將敏捷付諸實踐。

如果你是敏捷團隊的一員，只是想獲得足夠的知識來完成你的工作，你可以專注於第二部分和第三部分中的實務做法。從第 1 章獲得完整的概念，然後通讀你所處團隊正在使用的實務做法。如果你的團隊使用了目錄中未列出的做法，請查看索引，因為這些做法可能使用不同的名稱。

如果你不是敏捷團隊的一員，但你正在和一個敏捷團隊一起工作，請向他們請教需要閱讀的資訊。產品經理、產品負責人和設計師，請從第 8 章和「目的」這個章節開始閱讀。安全和維運的工程師，請查看「為維運構建軟體」、「盲點探索」和「事故分析」這三個章節。測試人員，請查看第 16 章。

如果你只是對敏捷開發感到好奇，請先閱讀第 1 章，然後再看看第二部分到第四部分的內容。你可以從看起來最有趣的做法開始閱讀，並且按任何順序閱讀它們。

本書的客座作者

我很幸運有幾位著名的合作者參與了第二版的編著。Gitte Klitgaard 在「安全感」這個章節專業地涵蓋了心理安全的話題。Diana Larsen 在「團隊動態」和「消除阻礙」兩個章節，帶來了她在組織和團隊發展方面數十年的經驗。至於第一版的合著者 Shane Warden 雖然無法參與第二版的編著，但他仍然提供了相當有價值的諮詢與回饋，而且我們在第一版上的成果，也構成了這個版本的基礎。

Gitte Klitgaard

Gitte Klitgaard 是一名敏捷教練、培訓師和導師，她專注於幫助組織構築心理安全、責任感和當責力。Gitte 是位貨真價實的專家；她可以準確而有效地幫助大家展現自身的能力與特質，從而取得成功。

如何追求「真」敏捷？

你要如何將各式各樣的敏捷想法落實到實際的敏捷團隊上？

練習！大量的練習！

實踐敏捷

即使沒有正式記錄下來的文件，每個團隊都有自己遵循的工作方式——**流程或方法**。該方法反映了軟體開發的基本理念，儘管該理念很少被闡明，而且也不一定是能夠自圓其說。

要追求「真」敏捷，你需要改變你的流程來體現敏捷的理念。相較於聽起來，要實現這句話既容易，卻又困難。容易是因為在大多數情況下，你可以從眾多現成的敏捷方法（例如本書中的方法）開始做起；困難是因為你需要改變自己的工作方式，而這代表會改變很多習慣。

要追求「真」敏捷，你需要改變你的流程來體現敏捷的理念。

敏捷社群將這些習慣稱為**實務做法**。本書的大部分內容都專注於這些做法，比方說規劃會議、自動化建置和向利害關係人展示（**stakeholder demos**）之類的做法。大多數的做法已經存在了幾十年，只是敏捷方法以獨特的方式將它們結合起來，突出那些支持敏捷哲學的部分，捨棄其餘部分，並同時混合一些新的想法。最終產出的是一個精實的、強大的、能自我強化的成套做法。

敏捷的實務做法常常能夠承擔兩倍和三倍的任務、同時解決多個問題、並以巧妙和令人驚訝的方式相互支持。直到你看到它的實際運行一段時間後，你才會真正理解敏捷方法有效的理由。

因此，儘管「從一開始就將敏捷方法客製成你想要的樣子」的想法相當誘人，但是最好的做法仍然是照著本書告訴你的方式進行。或許你會很想要去除最不熟悉的實務做法，但如果你想要追求「真」敏捷，它們往往是你最需要的做法。

精通敏捷之道

掌握敏捷開發的藝術需要實際的經驗，而這些經驗必須是基於具體的、有明確定義的敏捷方法。從書本上所描述的方法開始，然後將它們全面地付諸實踐，並花費幾個月的時間來完善你的用法，以及理解它為什麼有效。接著才是進行客製。從一個需要改善的地方開始，對實際的狀況進行有根據的假定，然後重複這樣過程，直到完善。

本書就是為了讓你精通敏捷之道而編寫的。這是一套在實務中得到證明且精心策劃的敏捷實務做法。如果想要使用它來掌握敏捷開發的藝術，或者僅僅只是想要使用敏捷的實務做法來取得更大的成功，請遵循以下步驟：

1. 選擇一部分的敏捷想法來掌握。第 3 章將幫助你進行選擇。
2. 根據第一步選定的想法，盡可能地採用對應的實務做法。本書的第二部分到第四部分會介紹這些實務做法。敏捷的實務做法能強化彼此的價值，所以當你一起使用它們的時候，可以得到最好的效果。
3. 嚴格並且一致地運用這些實務做法。如果某一個實務做法沒有產生作用，請試著更仔細地遵循該做法。剛接觸敏捷的團隊往往沒有充分地運用這些實務做法，預計需要兩到三個月的時間才會開始習慣這些做法，並且需要再花費兩到六個月，讓這些做法成為本能。
4. 當你確信你正確地運用了這些實務做法的時候，再花費幾個月的時間，開始嘗試改變這些做法。本書中的每個實務做法都討論了實務做法為何有效以及如何進行客製。每次對做法進行改變時，請觀察發生的情況，然後再進一步改善。
5. 沒有最後一步。敏捷軟體開發是一個不斷學習和改進的過程。永遠不要停止練習、實驗和進化。

圖 2-1 說明了上述的整個過程。先遵守規則；然後打破規則；最後把規則拋在腦後¹。

1 這過程的靈感來自於 Alistair Cockburn 對 Shu-Ha-Ri 的討論。

精通敏捷之道

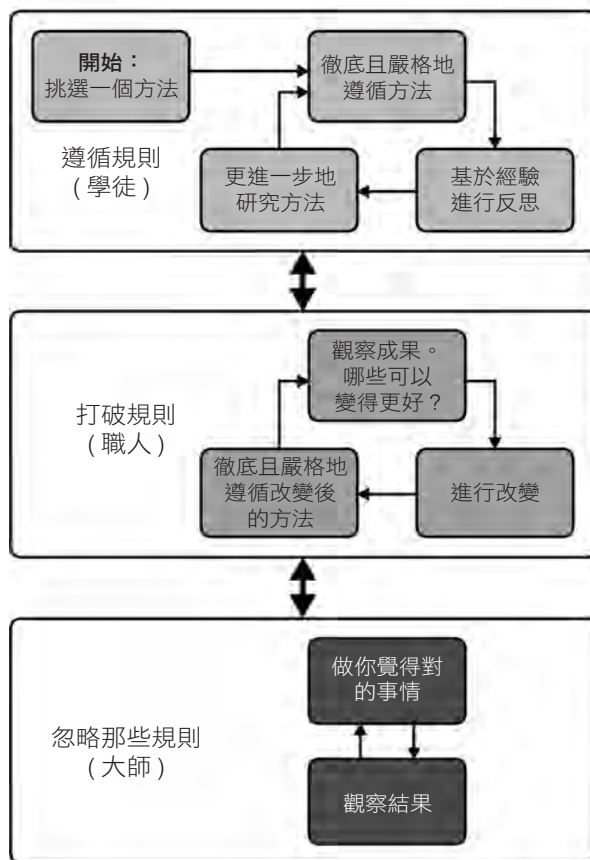


圖 2-1 精通敏捷之道

如何開始

你的第一步取決於你想要完成什麼目標。你是想加入現有的敏捷團隊、想把敏捷導入一個或多個團隊，還是想改進已有的敏捷團隊？

加入敏捷團隊

如果你計畫加入現有的敏捷團隊，或者只是想了解敏捷實務上是如何發生效用的，你可以直接跳到第二部分到第四部分。每個部分都以「生活中的一天」的故事作為開始，描述敏捷的樣貌。每個敏捷團隊都是不同的，因此你加入的團隊也不會完全相同，但這些故事會讓你了解將會發生什麼事情。

讀完故事後，請直接跳到你感興趣的實務做法。每一個做法的設計都以能被單獨採用的方式編寫。如果你的團隊使用的實務做法不在目錄中，請查詢索引，因為它可能使用了不同的名稱。

導入敏捷

如果你正在幫助你的組織創建敏捷團隊，或者你想說服組織採用敏捷，那麼第一部分中的其餘章節將告訴你如何開始。請使用以下清單，來保持一切的過程都井井有條。

首先，確認敏捷是否適合你的公司：

- 選擇一個達成敏捷的方式，而且是組織所支持的方式。（請查閱第 3 章）
- 為成功地達成敏捷，確定組織需要做些什麼？（請查閱第 4 章）
- 為試驗敏捷，尋求支持。（請查閱第 5 章）
- 如果你有多個團隊，請決定如何將敏捷規模化。（請查閱第 6 章）

在團隊嘗試敏捷之前的幾週內：

- 確定團隊的教練或教練群是誰，並且確定至少一個人擔任團隊的產品經理。（請參閱第 76 頁 — 「完整團隊」）
- 讓團隊的產品經理與來自高階管理的團隊支持者和主要的利害關係者會面，以製定目標草案。（請參閱第 116 頁 — 「目的」）
- 確保團隊擁有實體或虛擬的團隊空間。（請參閱第 91 頁 — 「團隊空間」）
- 安排並主持團隊的章程會議。（請參閱第 122 頁「規劃你的章程會議」）
- 要求團隊審視新的實務做法。提供本書副本供成員自學，建議他們在當前工作中嘗試一些實務做法，並思考為看起來具有挑戰性的實務做法提供培訓。（第二部分到第四部分會對這些實務做法進行解釋）

當一個團隊準備好開始時，深呼吸並且：

- 讓團隊成員規劃他們開始的第一週。（請參閱第 221 頁「你的第一週」）

改進已有的敏捷團隊

如果你已經擁有敏捷團隊並且希望他們變得更好，那麼所選擇的方法取決於你想要進行什麼樣的改進。

如果你有興趣微調團隊的現有流程，請直接跳到第二部分到第四部分，並且閱讀感興趣的實務做法。如果想做出更大的改進，這個過程與導入敏捷相同，只是你可以專注於想要改變的事情。請使用「導入敏捷」主題中的清單作為指南。

如果敏捷不適用於你的組織，請參閱第 39 頁「問題排除指南」。

採用個別的敏捷實務做法

當你全心投入敏捷的轉變時，能獲得敏捷最大的好處，但如果這樣的要求並不適合你，你可以將一些敏捷實務做法加到你現有的流程中，而以下這些地方是很好的切入點：

- **每日計畫**。如果你經常為工作被中斷所困擾，請嘗試採用為期一天的迭代（請查閱章節一「規劃任務」）。每一天的開始都使用規劃遊戲（請參閱第 187 頁「規劃遊戲」）和團隊所評估的產能（請參閱第 225 頁）「產能」來進行聯合規劃會議，然後把所有會產生工作中斷的事情推遲到第二天的計畫會議裡。務必確保成員對他們的任務進行評估。
- **迭代**。如果你沒有經常苦於被中斷工作，但仍想改進你的規劃能力，請嘗試使用每週迭代（請參閱第 211 頁「規劃任務」）。在這種情況下，每日站立會議（請參閱第 248 頁「站立會議」）和定期向利害關係人展示（請參閱第 279 頁「向利害關係人展示」）仍可能對你是有幫助的。隨著時間的推移，請考慮使用索引卡進行規劃，並使用大圖表來顯示接下來要處理的工作，如第 173 頁「視覺化規劃」中所述。
- **回顧**。經常性的回顧（請參閱第 316 頁「回顧」）是你的團隊調適與改善流程的絕佳方式。第 11 章中的其他實務做法可能也會對你有所幫助。
- **快速回饋**。快速、自動化的建置將對你的生活品質產生重大的影響，並且還能為其他的改善創造機會。更多相關資訊，請參閱第 378 頁「無摩擦」。
- **持續整合**。持續整合（指的是這個實務做法的理念，而不是專注於工具的使用）不僅減少了整合的問題，還推動了建置和測試的改進。詳細的介紹請查閱章節「持續整合」。
- **測試驅動的開發**。儘管測試驅動開發（請查閱章節「測試驅動開發」）不像其他實務做法那樣容易被採用，但它能帶來相當大的好處。測試驅動開發是減少錯誤、提高開發速度、提高重構能力和減少技術債的基礎。掌握它可能需要一些時間，所以要有耐心。

第二部分到第四部分中的其他實務做法也可能是有用的。敏捷實務做法之間相互依賴，所以一定要注意每個實務做法的「關聯」方框和「前提條件」的部分。

如果你在採用個別的實務做法時遇到困難，請不要感到沮喪。選擇一組具有連貫性的實務做法，並且全心投入，反而會更快且更容易。這就是我們接下來的章節想要探討的。

選擇期望的敏捷力

為了敏捷而敏捷是毫無意義的。相反地，你應該問問自己兩個問題：

1. 敏捷會幫助我們變得更成功嗎？
2. 敏捷會帶領我們達成預期的目標嗎？

當你能夠回答上述這兩個問題，你就能知道敏捷是否適合你？

組織重視什麼？

比起營收獲益，成功能帶來更多好處。以下列出部分的好處：

- 改善財務績效：獲利、營收成長、股票價值、成本節約
- 達成組織目的：策略目標、原創研究、慈善事業
- 提升市場地位：品牌覆蓋程度、差異化競爭、客戶忠誠度、吸引新客戶
- 獲得領會：策略資訊、分析、客戶回饋
- 減少風險：安全、法規需求、稽核
- 提升產能：聘僱、保留人才、道德、技能發展、自動化

敏捷熟練度模型

2014 年，我和 Diana Larsen 合作，分析了為什麼公司的不同敏捷團隊會產生如此不同的成果。我們倆從一開始就和敏捷團隊合作。多年來，我們注意到這些團隊傾向於追求截然不同的成果，而這些成果則分屬在不同的「領域」中。我們把這些觀察到的結果彙整到敏捷熟練度模型裡。圖 3-1 為敏捷熟練度模型的簡易圖。[Shore2018b]

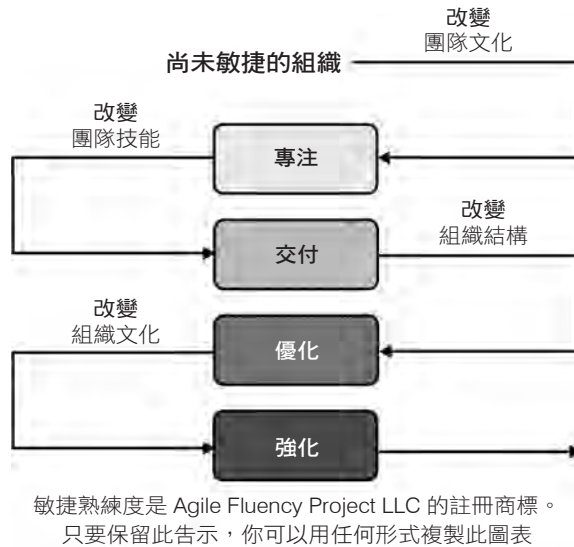


圖 3-1 敏捷熟練度模型簡易圖

每個領域都對應著一組好處。團隊需要熟練該領域才能獲得這些好處。當一個團隊能夠很自然地使用著領域中所需要的技能而沒有勉強之處，該團隊便熟練了該領域。

NOTE

雖然該圖顯示了從一個領域到下一個領域的一條直接路徑，但真實的情況卻是更為複雜。雖然圖上順序是典型的進展方式，但團隊可以從任一領域，並且以任一種順序來達到熟練。

達到熟練所需的技能會在本書第二部分到第四部分中進行介紹，但是熟練度並不是團隊可以憑一己之力便能實現的。這還有賴於你的組織對團隊的熟練度進行投資。這意味著不只是口頭上對敏捷想法的支持，而是必須做出實際且有意義的改變。這些改變需要花費時間、金錢和政治資本。

你從敏捷團隊獲得的結果取決於你的公司對敏捷理念的認可程度。當一家公司未能從敏捷中獲得他們想要的結果時，通常是因為他們沒有進行必要的投資。他們甚至經常沒有意識到需要投資哪些事情。

明白地做出選擇，來投資敏捷力。仔細地考慮每個領域，以及它們各自的成本和所能獲得的好處。選擇有最佳成本效益權衡和適合自己情況的領域。

有意識地做出選擇，
來投資敏捷力。

你可能無法說服你的公司對每個領域進行投資。沒關係！與能力成熟度整合模型（CMMI）等成熟度模型相比，熟練度模型並不需要循著基礎技能到高階技能的發展模型，相反地它展現了多種投資 / 效益的選擇。雖然圖示顯示了最常見的進展方式，但每個領域都能獨立地被採用，而且都有各自的價值。

熟練度與成熟度

熟練度是一種新出現的團隊屬性，而非用於個人。熟練度不代表團隊中的每位成員都具備該領域的所有技能，反而是團隊成員能以團隊整體的角度讓合適的人在合適的時間承擔相應的責任。

每個領域都對應數個成熟度級別：

1. 學習（*Learning*）。團隊正在學習領域內的技能。
2. 精通（*Proficient*）。當團隊全神貫注於技能時，能夠展現這些技能。
3. 熟練（*Fluent*）。當團隊內有敏捷教練陪同時，團隊能夠自發地展現領域內的技能。
4. 獨當一面（*Independently Fluent*）。團隊不需要任何敏捷教練或某位熟練成員的陪同下，便能自發地展現領域內的技能。

專注領域

專注領域與敏捷的基本原則有關：專注於商業成果、團隊合作、勇於任事。熟練於此領域的團隊聚焦於團隊的核心目標、先發佈最有價值的功能、並且根據變動的商業需求，來改變方向。他們持續地專注組織最有價值且最優先的目標。

對於大多數團隊和組織而言，這需要改變他們對團隊的看法。尚未敏捷的組織（Pre-Agile organization）習慣預先制定計畫、要求團隊進行評估，並期望獲得基於這些評估的工作進展報告。專注團隊時常（至少每個月）修改他們的計畫，並透過展示他們已完成的工作來展示進度。

尚未敏捷的組織習慣把計畫分解為任務，並把這些任務指派給團隊中的每個人，並根據他們完成任務的程度來判斷個人的績效。專注團隊會自己分解任務、自己決定誰來完成每項任務，並期待被當作一個團隊來評判他們創造價值的能力。

要使得團隊達成「專注」，組織需要投入團隊結構、管理和工作環境的改變，來支持技能上的變化（我會在下一章詳細介紹）。這既是好消息也是壞消息！壞消息是這正是見真章的時刻，一些組織並不願意對這樣的改變進行投資。好消息是如果組織拒絕這樣的改變，他們可以早點知道自己的確並不在敏捷哲理的道路上。你正好可以避免自己因追求形式上的敏捷，而身陷經年累月的挫折與心痛。

如果你能夠獲得認可，熟練專注會需要每個團隊大約 2-6 個月的全神貫注才能實現。在適當的支持下，他們將在 1-4 個月內超越之前的績效¹。本書第二部分會描述團隊所需的實務做法。

交付領域

敏捷團隊可能隨時改變他們的計畫。對於大多數團隊來說，這會慢慢降低程式碼的品質。他們會逐漸失去能力做出具有成本效益的改變，最終會說出需要丟棄目前的軟體成果並且重新開發，而這是一個昂貴且浪費的提議。

交付團隊通過卓越的技術來避免這個問題。他們設計出能夠面對頻繁變動的程式碼。並且維持程式碼的高品質，所以他們不會浪費時間來尋找錯誤。他們改進了軟體生產的生命週期，所以能夠毫無痛苦地發佈成果，並且能有效管理維運。他們有能力在產品最具有商業價值時，交付可靠且低缺陷的軟體。

實現這些結果需要大量投資團隊成員的開發技能，並且進行團隊結構的改變，好讓具有測試與維運能力的人員能整合到每個團隊中。

如果你的公司進行這些投資，交付熟練度會需要每個團隊花費 3-24 個月的時間來發展，而你會在 2-6 個月內看到改善的成果。每個團隊需要的確切時間，取決於現有程式碼的品質和團隊獲得訓練與指導的程度。本書第三部分有相關的實務做法。

優化領域

大多數公司都會因為熟練專注和交付而感到滿意，但敏捷期望的是更多好處。在它所達成的輝煌成就中，敏捷代表著一個世界，身處於其中的團隊能夠應對不斷變化的市場條件，輕盈地旋轉曼舞。他們試驗和學習、開發新市場、並且智取競爭對手。

1 本章中的所需時間都是根據我的經驗得出的大致數字，而可能與你的經歷有所不同。

優化團隊可以讓敏捷力達到這種水平。他們了解市場需要什麼、商務上需要什麼，以及如何在兩者之間架起橋樑。或者就像在創業環境中一樣，他們知道需要學習什麼，以及如何去學習它。他們持續地優化產品計畫，來實現最大的價值。

這需要組織結構的轉變。制定最佳計畫需要具有深厚商業和產品專業知識的人員全心地投入，這意味著讓產品和市場專家全職地加入開發團隊，而同時也代表著讓這些團隊對他們的產品預算和計畫負全部責任。

這種結構變化需要高階管理層的許可，而它可能不容易獲得。團隊通常需要花費至少一年的時間，透過交付熟練度來建立信任，然後才能獲得這些投資的許可。一旦取得許可，「優化」熟練度需要額外 3-9 個月的時間進行發展，儘管你會在 1-3 個月內看到改進，但即便如此，優化也是一個永無止境的試驗、學習和發現的過程。本書第四部分會描述如何開始熟練「優化」。

強化領域

敏捷熟練度模型有一個最後的領域，它主要是基於推測而來：敏捷可能的未來。它也只適用於處在管理理論和實務做法前沿的組織。這超出了本書的範圍。簡而言之強化領域涉及提煉團隊的共同見解並將其引導到組織改善當中。如果想了解更多資訊，請參閱第 19 章。

敏捷熟練度領域摘要

專注：

- 主要好處：專注優先的商業目標、團隊工作進展的可視度、改變方向的能力
- 投資：團隊結構、管理、工作環境
- 大致的時間節奏：1-4 個月的績效下降；2-6 個月達到熟練

交付：

- 主要好處：低瑕疵與高產出、產品的技術壽命
- 投資：開發技能、整合測試與維運
- 大約的時間節奏：2-6 個月的績效下降；3-24 個月達到熟練

優化：

- 主要好處：更有價值的發佈與更好的產品決策
- 投資：產品管理的授權、預算與計畫的團隊主導權
- 大約的時間節奏：1-3 個月的績效下降；3-9 個月達到熟練

選擇目標領域

你的團隊應該追求哪些熟練度領域？這取決於組織可以支持哪些領域。在不考慮任何因素情況下，專注、交付和優化全都是你最佳的選擇。這三個領域的結合能夠帶來敏捷最好的成果與最貼近的實現方法。

但同時選擇三個領域也代表需要最多的投資。如果無法向利害關係人證明這些投資的合理性，你可能很難取得需要的支持。在沒有足夠的投資情況下，你的團隊將難以達到熟練，而你會白白承擔學習成本而一無所得，甚至會看到比目前工作方式更糟糕的結果。

換句話說，只選擇公司既需要又願意投資的領域。

所以，你應該選擇哪個領域？

- 每個敏捷團隊都需要專注熟練度，因為這是一切的基礎。如果你的公司對於專注熟練度都無法給予投資，那麼對於你的公司來說，即便你可以透過交付熟練度來逐步實現敏捷，它可能都不是一個好選擇。
- 交付熟練度可以降低成本並提高開發速度。沒有它，你的程式碼最終只會被技術債壓垮。這樣的現實使得交付熟練度對於大多數的團隊來說是一件輕而易舉能被接受的事情。也就是說，即便一些組織還沒有準備好為交付熟練度領域所需的學習和程式碼品質，進行大量投資。他們也可以先從專注熟練度開始，接著展示成功，然後用交付熟練度來證明進一步投資的理由。
- 優化熟練度是敏捷最閃耀的地方，同時也是最多人詢問之處。對於大多數組織來說，最好先透過展現專注和交付的熟練度來建立信任感，然後逐步地承擔更多的責任，再來思考這個領域。不過，如果你的組織已經具備授權決策給跨職能團隊（就像在新創公司中經常看到的那樣）的文化時，優化熟練度將會帶給你最好的成果。

每個領域的詳細資訊及其好處，請參見本書的第二部分、第三部分和第四部分的介紹。有關所需投資的詳細摘要，請參閱第 26 頁的「投資項目彙總」。如果你不確定要選擇哪些領域，請從專注和交付開始。

無論你選擇哪個領域，都要同時投資學習它們的所有實務做法。後面領域的實務做法能夠讓前面領域的實務做法運作地更好，因此你最好一起採用它們，而不是一次採用一個。不過如果你不能在想要的每個領域進行投資，那也沒關係！雖然需要花更長的時間，但隨著時間的推移，你將能夠進展到其他領域。

一旦你知道想要哪些領域，就該更詳細地考慮組織的投資了。我們將在下一章中研究它們。