

推薦序

Ken Schwaber

充滿智慧的傑出成就。

《*Scrum 精華指南*》是一本非常出色的書。Gunther 以清晰的文字與結構描述了關於 Scrum 的一切，這些描述充滿洞察力、理解力和感知力。你不會感到詫異，而是單純的從中受益，並在之後回想時發現：「這真的非常有用，我找到了我需要知道的東西，沒有任何困難的、輕鬆的理解了我想要的東西。」

我很糾結如何寫這個前言，對我來說，前言應該要像它所介紹的那本書一樣優秀，但以這本書來說，這是很難達成的。閱讀 Gunther 的書，不管只是讀一小部分或是全本書都讀，你都將會感到滿意。

Scrum 是簡單的，但是足夠完善以解決複雜的問題。Gunther 的精華指南是如此完善，足了讓人了解這個可以處理複雜問題的簡單框架：Scrum。

Ken，2013 年 8 月 22 日

作者序

在敏捷的使用持續受到關注的同時，Scrum 成為最為廣泛被採用的敏捷定義。人們普遍對 Scrum 感到有興趣，除了軟體開發領域之外，Scrum 的用途仍在不斷擴大。

將組織的工作方式轉變為 Scrum 是一個巨大的挑戰。Scrum 並不是食譜，能對任何想像得到的情況都提供詳盡、詳細的處理方式。Scrum 是一個關於規則與價值觀的輕量原則框架，在使用 Scrum 的人們中興盛。Scrum 的主要潛力在於發現，並透過實踐、工具和技術來展現，以及針對特定情境下對其進行優化。

Scrum 能實現的好處，起源於消除障礙、跨越區隔的思考並踏上發現之旅的意願。

這個旅程始於了解 Scrum 的規則並開始參與 Scrum，希望這本書能成為你這趟旅程的伴侶。本書展示了 Scrum 如何實現敏捷心態、Scrum 賽局規則是什麼，以及這些規則如何為各種戰略提供了發揮空間。之所以會介紹所有相關面向，是為了使本書對於人員、團隊、經理和變革推動者來說都值得閱讀，不論他們已經在 Scrum 的旅途中或是準備踏上 Scrum 之旅。

我的旅程始於 2003 年，我的敏捷之路開始於極限編程和 Scrum，這無疑是一條不平順地石子路。我曾在許多團隊中使用 Scrum，在不同的規模與組織中執行多項專案和計畫。我與大型和小型企業都合作過，指導對象從個人從業人員、團隊到執行管理階層都有。我與 Scrum 的共同創始人 Ken Schwaber 合作，同時主持了 Scrum.org 的「Professional Scrum」系列培訓課程和評鑑。我很感激一路至今，我能夠以獨立的 Scrum 照護者的身份繼續旅程。

我在 2013 年撰寫了《Scrum 精華指南》的第一版，我認真的在第一版中描述 Scrum 價值觀，在 2016 年 7 月，Scrum 的價值觀被添加到 Scrum Guide 中。我同時闡述了傳統的 3 個提問，這是在 Daily Scrum 中可以被選用的好策略，這些可選性於 2017 年 11 月添加到 Scrum Guide 中，直到 2020 版中刪除了這部分的描述，這些提問其實是可以選擇性採用。

但是，越來越多的挑戰不斷浮出水面。社會的平衡不斷急劇地從工業（通常是體力勞動）向數位化（通常是虛擬）勞動轉移。社會的許多領域，工作的不可預測性不斷增加，工業典範不意外的開始變得無用。對敏捷典範的需求比以往任何時候都大，透過採用有形的 Scrum 框架來幫助人們和組織提高在複雜環境下執行複雜工作的敏捷性。

人們逐漸發現 Scrum 不僅是產出複雜（軟體）產品的方法，同時也是一種「讓人們能夠從複雜的挑戰中獲取價值」的簡單框架。除了在軟體和新產品開發之外，越來越多來自不同領域的人希望能在 Scrum 的旅程中獲得更多的指導和見解，因此需要能從不同角度、以不同方式說明，對 Scrum 規則需要有更通用的描述，這因此促成了本書第二版的變更。當組織構想透過 Scrum 的工作方式重新建立其結構時，他們可以在 Scrum 簡單的規則中找到清晰的見解。

我相信第三版能為人們及其組織提供更多以往沒有的重要基礎見解，無論他們在哪種業務領域，都可以有助於正確地塑造他們的 Scrum。雖然引入一些術語更改，重點仍是放在框架中的規則和角色的意圖和目的。

我感謝 Ken Schwaber 的前言和他對原始版本（2013）的評論，以及其他評論者 Dave Starr，Patricia Kong 和 Ralph Jocham 對第一版的反饋。感謝 Blake McMillan 和 Dominik Maximini 對第二版（2018）的評論。我感謝 Bhuvan Misra 對第三版的讚賞和關鍵的回饋。我感謝所有過去的和未來的翻譯人員，對多種語言傳播所做的不懈努力。我感謝凡赫倫出版社（Van Haren Publishing）的所有人，尤其是 Ivo van Haren，讓我有表達 Scrum 觀點的機會。

祝你閱讀愉快。

持續學習，
持續進步，
持續…Scruming。

Gunther

獨立的 Scrum 照護者

2013 年 6 月，2018 年 8 月，2020 年 11 月

名人推薦

《*Scrum 精華指南*》是一本非常出色的書。它是如此的組織嚴謹、編寫良好，內容也非常出色。對於所有尋求完整清晰的 *Scrum* 概論的人來說，這本書稱的上是標準講義。

(*Ken Schwaber*，*Scrum* 共同創始人，2013 年 8 月)

Gunther 為尋求敏捷性的團隊精心準備的此本正確的、毫無廢話的指南。我不想誇大，但這正是我希望能寫出的 *Scrum* 敏捷性的書。

(*David Starr*，敏捷工匠 *Agile Craftsman*，2013 年 6 月)

在我過去許多 *Scrum* 培訓活動中，我經常被問到：「以 *Scrum* 來說，可以推薦讀哪一本書嗎？」我總是無法給出直接明確的答案，但現在就可以了！《*Scrum 精華指南*》是踏上 *Scrum* 之旅時該閱讀的那一本書，一本有關於 *Scrum* 的簡潔、完整並且充滿熱情的參考書。

(*Ralph Jocham*，敏捷專家，2013 年 6 月)

「*Scrum* 之家是所有人都受到歡迎的溫暖之家。」Gunther 對 *Scrum* 及其參與者的熱情在他的工作和本書的每一章中都清楚展現。他解釋了敏捷典範，列出了 *Scrum* 框架，還討論了「*Scrum* 的未來狀態」，僅用了大約 100 頁就讓讀者深入認識 *Scrum*。

(*Patricia M. Kong*，企業敏捷解決方案，2013 年 6 月)

我建議您在進入 *Scrum* 旅程的初期儘早閱讀《*Scrum* 精華指南》，來幫助您更深入地了解 *Scrum* 如何運作，以及其價值觀和原則如何對團隊的生活以及組織的健康產生積極影響。在旅途的後期閱讀也很棒……除了可能會為了沒有早些閱讀而感到遺憾。

(*Blake McMillan*，首席顧問，2018 年 8 月)

簡單來說，很難找到簡潔扼要的 *Scrum* 文獻，大多數作者都圍繞核心主題討論但不點明，Gunther 選擇打破這種模式，讓我們對 *Scrum* 真正重要的部分得到了了解與啟發。在開始 *Scrum* 之旅時，請確保你隨身攜帶本指南。

(*Dominik Maximini*，敏捷領導教練，2018 年 8 月)

「體積小，價值高。」Gunther 這本關於 *Scrum* 的精華指南是少數我同時擁有的精裝版和電子版的書籍之一，方便我隨身攜帶。這是一本很棒的書，一本絕佳的 *Scrum* 指南，強烈推薦給所有即將踏上 *Scrum* 之旅有抱負的旅行者！

(*Bhuvan Misra*，敏捷石匠，2020 年 11 月)

1

敏捷典範

■ 1.1 轉型還是不轉型

軟體業長久以來受工業的觀點與理念支配，這實際上只是在複製傳統製造業的習慣與理論。這種泰勒學派¹的知識、觀點或實務作法的理論核心認為「工人」沒有完成工作所需具備的智能、自主力與創造力，員工只能做已經被規範好、可立即執行的任務。因此他們的工作都必須由更資深的員工事先準備、設計和計畫。於此同時，更高層的管理者仍需不時監看他們如何執行這些詳細規劃好的任務。並且透過接受好的產出、並打回壞的產出這種方式來確保品質。此外也用金錢獎勵期望中的行為，不樂見的行為則會受到懲罰，也就是傳統的「胡蘿蔔與大棒」的管理策略。

1 Frederick Taylor (1856-1915 年) 是美國工程師，他因研究如何最大限度地提高勞動生產率和效率，同時最小化成本而聞名。他促進了強制性標準化以及系統方法和實踐。控制完全取決於管理層，而員工只需要執行工作就好了。

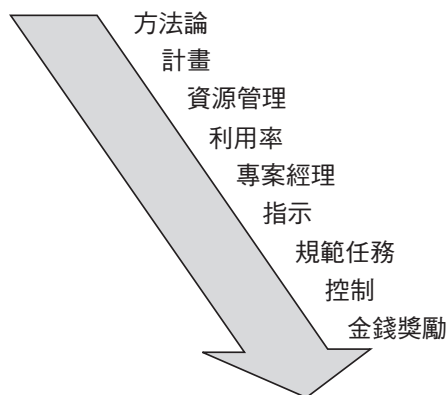


圖 1.1 工業典範

將這種舊式典範套用於軟體產業導致的嚴重缺陷已經有許多明確記載並廣為周知。尤其是 Standish Group 的 Chaos 報告 [Standish, 2011; Standish, 2013] 不斷揭示了使用傳統方法在軟體開發中的低成功率。工業管理規範用於軟體開發所導致眾多缺失與錯誤，早已超出可忍受的合理範圍，不幸的是我們的對策卻是降低期望。在工業管理規範中所認為的「成功」是能在期限與預算內完成規畫好的所有功能，卻使得軟體專案只有 10-20% 的成功率變成可以被接受。依循工業管理典範保證達到的「成功」是充滿爭議的，導致傳統軟體開發業被迫去接受品質不佳，且超過 50% 功能根本沒有被使用的產品 [Standish, 2002; Standish, 2013]。

雖然尚未普遍的被察覺，但工業典範確實讓軟體開發陷入嚴重的危機中。很多人仍嘗試透過加強工業典範的方式去克服這個危機，耗費心力的前期工作持續增加，更多的計劃產生，更多的階段被制定，更多的設計，更多的前期工作，希望能讓真正工作時更有效。但成功率卻沒有提升，而工業典範的思維仍認為是因為指示不夠詳盡。這是因為其核心理念認為工人需要被指導，只是要給他們更詳盡的指示，這導致監督單位不斷的增加並加強，並給出更多指示細節。

然而，這樣的嘗試並沒有什麼改善。我們依然在容忍大量的缺陷、錯誤、與品質低落。

雖然花了一段時間，但在觀察到工業典範下這些明顯的異常後，新的想法和理念必然開始成形。

新世界觀點的種子在 90 年代就出現了。但在 2001 年才形成一個正式的名稱「敏捷」，這是軟體開發歷史的轉折點。新的典範在軟體業誕生，這種典範興盛的基礎在於啟發與創造力，（恢復）對創造工作與對「工作人員」智慧的尊重，並且快速擴散到其它的領域。

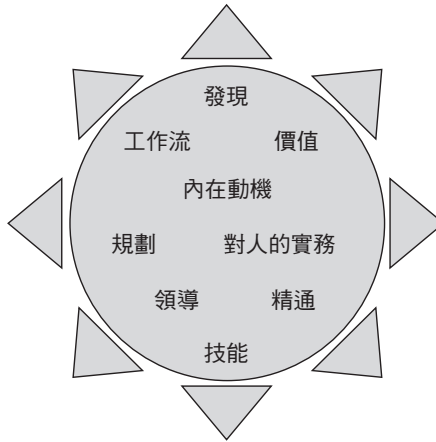


圖 1.2 敏捷典範

軟體業有很好的理由轉向新的典範：眾所皆知現存的已知缺陷相當明顯，與此同時社會中軟體的使用呈指數的增長，這讓新典範在現代世界成為關鍵的一環。然而，從定義來看，典範轉移需要時間，舊典範已經紮根，並有一段相當長的衰退期，至今仍然有人教導與推廣工業方法才是合適的軟體開發方式。

許多人認為敏捷太過於激進，所以他們發展出基於現存傳統框架平緩過渡式的敏捷實拖方法。然而，我們有理由對從舊典範到新典範、從瀑布到敏捷這樣緩慢地演變方式抱持懷疑。

有很高的機率這些平緩的演化永遠都只停留在表面，所有的作為都只是隔靴搔癢。他們會使用新的名稱、新的詞彙和新的實施方法，但根本思維與行為並沒有改變。基本缺陷仍沒有改變；特別是對人的不尊重，以致於持續將這些有創造力、聰明的人當作沒有智能的「工人」，視為一種「資源」來對待。

保留傳統方法的基礎，意味著保持使用既有的資料、指標與標準，並且依據那些舊標準來衡量新典範。但是，不同的典範本質上是由不同的概念與思想組成，而且常常是互相排斥的。簡單來說，在工業典範和敏捷典範之間沒辦法進行有意義的比較。必須誠實才能接受、承認舊方法中存在各種嚴重的缺陷，必須具有領導力、遠見、企業家精神及堅持的人，才能真正放棄舊的思維，接納使用新方法。

實際上，漸進式轉變跟完整保持工業典範的現狀沒有不同。

有大量的證據顯示舊典範不管用，但過去大部分關於敏捷的正面結果的證據仍偏向個人經驗、故事或是相對無足輕重的來源。2011年 Standish Group 的 Chaos 報告是一個轉折點 [Standish, 2011]，這是第一次有清晰的研究證據，並且在其後所有的 Chaos 報告中持續驗證。後來有更多的研究針對用傳統方法與敏捷方法專案的比較。即使是使用傳統的準時、如預算、交付所有功能這些期望來比對，該報告顯示出敏捷方法的效益高出許多。相對於傳統專案，成功的敏捷專案高出了三倍，同時失敗專案也低了三倍。然而在大型專案上，以固定時間、預算、範圍組合這樣的錯誤期望來說，成功率的變化並沒有這麼明顯。相對地在正確的期望下，把注意力放在積極地和客戶協作並頻繁地交付價值，新典範透過垂直切割價值，頻繁交付，來克服大量問題，表現得甚至更好。

說到底，敏捷是一種選擇，不是必須的。但這是改善軟體產業的一種方法，而研究顯示這種方法相當成功。



Scrum 能有幫助。

Scrum 是採用和扎根於敏捷典範的具體方式，透過特有的規則讓人們更容易掌握新的典範。只需要少量的規範指示就能立即展開行動，從而更有效、長期的吸收新典範。使用 Scrum，人們發展出新的工作方式：用探索與基於實驗來學習與協作的新生活型態，進入了敏捷的生活型態，這個過程有助於組織轉型到敏捷這種透過不斷改變、變化、進化與適應的狀態，來釋出時間、人員與精力，而（再次）變得創新。

儘管 Scrum 是極簡主義的，但經驗顯示採用 Scrum 仍然是一個巨大的跳躍。儘管已經證明這些舊的、明確的事物不可靠，但放棄舊的、明確的事物仍會引起不確定性。進行有意義的轉變可能需要時間，也可能需要決心與努力。一次次的經驗顯示，Scrum 很單純，但不容易。

■ 1.2 敏捷的起源

儘管計劃驅動的行業觀點佔據主導地位，但軟體開發採用演進方法並不新鮮。Craig Larman 在他的《敏捷與迭代開發：管理者指南》[Larman, 2004] 一書中廣泛描述了敏捷的歷史前輩。

正式被稱為「敏捷」是在 2001 年二月，當時 17 位軟體開發領導者在猶他州的雪鳥滑雪渡假村聚會。當時他們分享對於用 RUP（Rational Unified Process）這種重量級的實作方式來取代失敗的瀑布式方法的看法。他們發現這種做法並沒有帶來比傳統流程更好的結果。這些領導者各自遵循不同的路徑和方法，每種路徑和方法都在未來成為新的敏捷典範中獨特的表現方式；Scrum、極限編程（eXtreme Programming）、自適應軟體開發（Adaptive Software Development）、Crystal、功能驅動開發（FDD）等。

這次聚會讓這些領導者給了共同的想法、信念、原則和方法一個名稱——「敏捷」。他們將它發布並稱為《敏捷軟體開發宣言》[Beck, et al., 2001]。

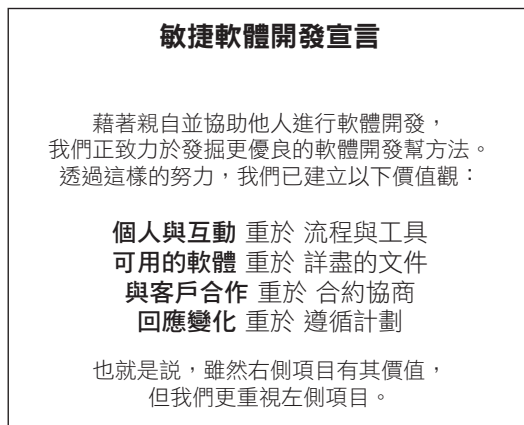


圖 1.3 敏捷宣言

常聽到有人說想要「做敏捷」或「變成敏捷」，然而他們通常期待的是一個神奇的解決方案，一個可以解決所有問題的萬能流程，這讓我常說「敏捷不存在」。敏捷不是一種固定的流程、方法、或實施方式，而是敏捷軟體開發方法共同原則的集合。敏捷指的是符合敏捷軟體開發宣言表達的信念與偏好的一種思維方式。

這個宣言有助於掌握支持敏捷的概念。如果你想用它來加深對敏捷的理解，那麼我強烈建議你看看 4 項價值陳述背後的 12 項原則，請參閱：

<http://agilemanifesto.org/principles.html>

■ 1.3 敏捷的定義

因為敏捷尚未有簡單明確的定義，我偏好使用三項主要特徵來描述「敏捷」。這些是各種敏捷方法的共同特性，也是典型的敏捷工作方式：

- 以人為本；
- 迭代增量流程；
- 以價值衡量成功

1.3.1 以人為本

敏捷不是由預測性計畫驅動，不是在闡述如何實作事先進行過詳盡分析、設計和架構的需求，敏捷承認無法事先預測需求的所有細節。

敏捷也不是在處理不同功能部門獨力完成的各種半成品的流程。

敏捷是由所有必要部門的人不斷的協作所趨動，不論這些人是被稱為業務、資訊、行銷、銷售、人資、客戶服務、維運還是管理人員。敏捷不認同傳統的業務與資訊技術的矛盾，若從創造出可用的、有用的、有價值的軟體產品的角度來看，這兩者缺一不可。

我們對協作、互動與對話愈來愈重視，並需要不同的管理方式才能達到效果，敏捷團隊需要以服務式領導來促成，提供邊界與自主管理的情境給有目標與方向的團隊。透過設定邊界產生不同的控制，而不是透過控制個人、任務或預估。協作與服務取代了指示每個人每天該執行的任務這種傳統的命令與控制機制，以及使用權威進行侵略式控制。

這表達了對人員的創造力、智能與自組織能力的尊重。尊重人有足夠的能力來理解並解決問題，而不需要大量的儀式與官僚主義。過量的儀式性讓人的協作思維、創新與責任被扼殺，並被官僚主義、書面報告、工作移交和行政藉口取代。

人在工作中的時間受到尊重，花在工作中的時間也是基於可持續的強度，讓工作可以有組織的依此節奏永久地持續下去。

1.3.2 迭代增量流程

敏捷流程不是一種隨興的方法，敏捷流程是被定義的且要求大量紀律。

產品是透過一次次對之前版本的新增、改善、刪減與修改逐步建立的（增量）。透過頻繁地檢視各個部件與整體產品來確保整體的完整性（迭代）。

敏捷要求所有參與者明確地關注品質和卓越表現。敏捷取代了以為將內容塞進文件與紙本記錄就會得到一樣的最終結果、可釋出產品或服務的想法。

無論花費多少時間、精力與費用來事先預測和規劃，需求與實作都還是發生改變，這個發現讓我們更加需要迭代 - 增量式流程。舉例來說，市場與競爭者會不斷發展、使用者開始使用後才會知道他們要什麼、企業策略會改變。而改變需要相當高的警覺性與開放的心態。

相對於預測性流程，敏捷流程不排除改變，也不會在開發工作中驅逐改變。敏捷活躍的中心由新的理解、進化的意見與優先序改變等構成。敏捷的興盛基於會持續出現並逐步顯現的需求、計畫、想法、架構與設計。改變不再是一種破壞，而成為常規工作中自然形成的一部份，敏捷甚至鼓勵改變，改變是創新與改善的來源。

1.3.3 以價值衡量成功

在變化逐步顯現的環境中，進度無法像工業典範中以符合預測計畫與里程碑、文件、工作移交、簽名、許可或其它形式上的義務來衡量或保證。因此敏捷採用新的方式來衡量進度與成功。

敏捷明確指出，進度和成功，是透過頻繁檢視可用版本產品（而不是中間相關的文件），及產品能提供的實際價值來判斷。

使用者只有在實際使用時才能評估它的可用性與實用性，這在產品開發中是一件很自然的事，沒有文件或虛擬程序能夠取代。迭代 - 增量流程符合這樣的情境，定期的取得使用者的回饋來衡量影響力與滿意度，作為產品演進的重要資訊來源。

■ 1.4 持續迭代增量

敏捷方法將時間切割為有限時的迭代，一個迭代是有固定開始與結束日期的時間區間。限時（time-box）這種技巧有許多的好處，其中最重要的是專注。這種時間管理技巧有助於改善工作轉換與分散的狀況，也確保能規律地檢查並持續地學習改善。每個迭代的核心目的是至少在迭代結束前產出具有價值且可用產品版本，以利取得回饋並實現早期學習。

敏捷重新組織了所有開發工作，用來最大化回應與利用商業機會的能力。

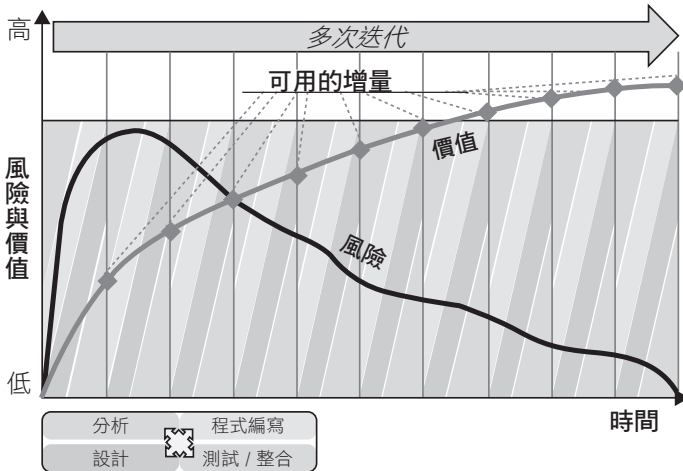


圖 1.4 敏捷價值交付

「價值」對使用者、市場、商務需求等面向，是衡量整體進度和成功的答案。在工作釋出到市場之前，價值都只是一種組織內部的假設，將產品釋出到市場是唯一能有效驗證價值假設的方式。定期向市場釋出產品，是唯一可以用來收集並合併市場回饋與滿意度（或是市場缺乏反應）的方法。這是在產品的後續演變中完成，價值會在迭代過程中不斷的最大化。透過明確的開發標準，連續地產生可用的增量，因此風險也能夠被控制。

「風險」也包含業務觀點，注意在現在的 IT 情境中，風險常被認定成是技術議題（這個系統可以運作嗎？這個系統可以因應大量使用嗎？），這涵蓋了產品可用性的觀點（技術上可處理嗎？不會壞嗎？），但是僅關注技術或開發面向的風險，而忽略了敏捷開發的最終目標是讓使用者和客戶更加滿意以確保產品的實用性，技術上可用的產品僅僅是一個開始。產品必須同時具有可用性與實用性，以確保成功地實現價值交付。

任何現代開發流程，都應該解決包括無法利用未預期的市場機會、無法夠快的釋出產品、引起客戶不滿（例如釋出未測試的成果、釋出使用者不想要的功能、釋出落後於競爭者的產品）等風險。

敏捷開發以最大程度降低（業務）風險的方式運作：高價值需求最早被回應、產品快速且頻繁地釋出新版本與更新。滿足現存需求的同時，也提供超乎預期的創新功能，讓使用者願意為產品付費並讓利害關係人的回報最佳化。追求高品質來最大程度地降低維護與支援，讓總體持有成本（TCO）達到最佳化。敏捷開發的方式，是讓從事核心工作的人員（再次）開始參與。

敏捷認同「一般」IT 活動（在圖 1.4 中簡單的表示為分析、設計、程式編寫、測試或整合）的核心目的，但拆散這些活動的順序。這些活動的結構被重新組織，用來產生適當的動力釋出產品版本，並盡早實現效益，目標是提高彈性與速度，而不是形成阻礙。在敏捷中這些素養全部必須以不連續、增量式的方式，每天在跨技能團隊持續協作與討論新想法、技術與實務作法中同步地進行。

這種整合式、跨功能方法的目標是為了在過程中提高品質並降低缺陷，而不是用生產後再來抓程式臭蟲的方式來達到品質要求。從想要定期釋出演變為實際有能力可以定期釋出是非常重要的，品質無法在產品完成後才增加上去，在開發程序之後才發現品質的缺乏，會導致失控的延遲與預算超支。

要實現敏捷開發實際和持久的利益，就需要超越 IT（或類似的技術）部門的邊界。敏捷不但擁抱變化、與變化共存，甚至是鼓勵變化，採用的方式在組織中也會是一大挑戰，但也是一個機會。整個組織會因為採用短週期、頻繁得到結果、演進式調適的敏捷方法而興盛。敏捷觀點與方法最終會讓大部份的組織部門不再試著去預測那些無法預測的事。敏捷善於利用交付產品時出現的答案、解決方案與有競爭力的想法。

可能需要一些時間才能體驗到敏捷帶來的持續學習，能夠對動盪的企業、業務與市場環境提高了控制。可能會需要一些時間，管理者才能轉移焦點不再固守過往判斷的方式，例如轉為透過事實與時間的記錄來了解所執行工作的影響和成果。也可能會需要一些時間，才能提高對於以敏捷開發流程增量成果取得回饋來優化價值的信心。

可能需要一些時間，來接受敏捷需要時間來改變。可能需要一些時間，來接受敏捷不需要事先分析、設計、計畫。

■ 1.5 敏捷無法由計畫產生

敏捷是一種透過敏捷工作方式來實現的狀態。敏捷性是一種講求持續變化、高響應性、快速和高適應性的狀態，是一個需要應對不可預測性的狀態。而這種不可預測性，則是產品開發中的創造性工作，以及組織所處的不斷變化的市場的共同特性。

如果這些響應能力、速度和適應性的變化，沒有延伸到組織與市場、社群和消費者之間的關係，則敏捷性沒有任何意義。應用敏捷流程是為實現企業敏捷的重要基礎，隨著敏捷的採用，出現了新的流程以及新的學習、改善和不斷適應並恢復對人的尊重的組織文化。在整個應用過程中，這些重要的學習會在組織中紮根，並注入到組織的DNA中。

要達成更高的敏捷性，理解一些基本事實是至關重要的。若不接受這些基本事實卻想引入敏捷，反而會將提高敏捷性的大門關閉，而不是讓其成為機會之門：

- 敏捷無法透過計畫得到；
- 敏捷無法透過要求得到；
- 敏捷無法透過複製得到；
- 敏捷沒有結束狀態。

以時程計劃來導入敏捷反而會帶來怪異且令人失望的結果。敏捷代表一種新的典範，轉變至新典範將引起顯著的組織性動盪，現有的程序、部門和功能都會受到影響，這種變動相當複雜，因此難以被預測，沒有方法能預先計畫和控制下一步，無法預測什麼時間點將需要什麼變化、如何應對這些需求、新的工作方式將如何生根以及會得到什麼明確的結果。我們無法預測這些轉變傳播和紮根的速度。

敏捷本身並不只是遵循新的流程，而是行為、文化轉變。決定轉向敏捷就是決定放棄舊有的（工業）方式。我們不僅是接受敏捷，甚至是

將敏捷視為一門實踐可能性的藝術。需要具有勇氣、誠實和決心，要能立即採取行動，面對並處理透過迭代增量的流程取得的真實資訊。敏捷就是要在可能的每時每刻都盡力而為，接受我們所擁有的方法與手段的限制，並且要面對所出現的侷限。時程計劃的敏捷轉型忽略了敏捷的本質是透過有目的的實驗和學習等步驟來處理複雜問題的方法。計劃時程和嘗試針對時間限制計劃敏捷只是在延伸舊思維，計劃會產生嚴重延遲和等待時間，實際上會適得其反，減慢轉型過程。

時間計劃營造了有最後期限和最終狀態的錯覺，但敏捷是沒有終點的。敏捷是一種不斷改善的狀態，認同並接受現狀會不斷的被我們自己的意志或外部動盪挑戰的狀態。

敏捷性是一個獨特且不斷發展的狀態，它反映了組織曾經歷和正在經歷的各種教訓和學習，在這個過程克服了特定困擾和障礙，也不可避免的進行了許多檢視和調適。敏捷性是一個獨特的簽名，在組織內部的人員、關係、互動、工具、流程、實務方法、構造和跨組織的其他生態系統等方面都留下了印記，沒有模型可以預測、概述或截取這種獨特的簽名。敏捷是一條需要遠見、信念、毅力以及努力的道路。敏捷是一種高度調適的狀態，透過定期依據實際工作與可見結果來進行調適，今天有效方法的可能明天就行不通了，團隊、技術和業務的有效工作組合有可能在另一種組合就無效了。在複雜、創新、激烈的競爭和不可預測的世界中，沒有調適的檢視是沒有意義的，另一方面來說，沒有觀察和檢視的調適是沒有方向的。

面對不可預測的產出結果，實踐可能性的藝術，可以使人們參與其中並加速轉變，在未定的未來以及可能帶來的成就上興盛，成為塑造未來的動力。對於那些具有遠見、決心和足夠投入的組織而言，他們有擺脫遵循計劃或複製模型的勇氣，將迎來一個光明的未來。

這些基本真理必須存在於每一個使用敏捷思維去管理、指引、促進、生活或領導轉型的人心中。即便如此，參與在轉型中的人仍需要時間去適應敏捷。畢竟，這幾十年來甚至更長的時間，人們都受到工業典範的「錯誤」行為的指導。

■ 1.6 結合敏捷與精實

精實（Lean）就如同敏捷一樣，重要的是要意識到它是一套思維工具，一系列相互交織的原則，這些原則可以教育、激勵、重視和指導人們不斷優化他們的工作和處理工作方式。精實原則是用有系統的方法，讓人用可持續和相互尊重的方式更快地產出更好的產品。

精實沒有一個正確標準、完全成熟、一體適用或統一的流程，被用來提供定義好與制定好的階段、角色、定義、工件、可交付成果等。對於產品開發或製造，精實流程應是根據其基本原理和思想進行設計，強調調適性，不斷調整以適應實際情況。*Bas Vodde* 和 *Craig Larman* 的《精實入門》（*Lean Primer*）線上文件 [*Larman & Vodde, 2009*] 在介紹精實的起源、原理和思想方面做得非常出色。

1.6.1 精實的主要面向

人

任何聲稱是精實的系統，基石都是人。「人」指的是精實產品開發 / 建置系統的整個生態系統中，不論內部或外部，每個可能的參與者：包括客戶、工作者、團隊、供應商和管理者。

所有人都以自己的方式為製造產品做出貢獻。他們透過協作，來減少階段交接、延遲和等待時間。他們自主地做出決定，並專注在知識的取得以及持續學習。管理者像是老師，身處於工作現場鼓勵、促進精實思維系統；幫助人員了解如何反思自己的工作、工作成果以及如何製造更好的產品。整個系統體現了「改善」（*Kaizen*）的精神：一種不斷思考流程、產品和可能的改善的態度。發生問題時，系統中的每個成員都可以「停止作業」² 來查明問題的根源，以便提出或採取對策。

2 這是指豐田（Toyota）起源於精實的生產線流程，每一個在生產線上的人都有權利在發現問題、缺陷或品質不佳時，叫停生產。

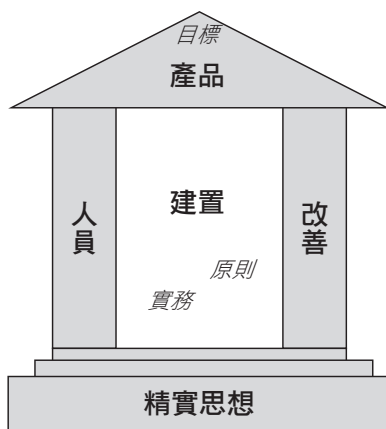


圖 1.5 精實殿堂

價值鏈中的每個人都以整體的方式工作。與供應商和外部合作夥伴的關係不是基於大量採購、重大談判和相互施壓的傳統方法，而是在相互分享利潤（和風險）的基礎上建立關係。精實合作關係相互的成長。

浪費

在討論到浪費這個主題時，重要的是**避免浪費**，而首選方法是透過持續改進和小幅度優化。另外，「浪費」指的是過程流程，而不是用來減少人員的藉口。

顯然無論如何注意避免浪費，浪費都會發生。改善精神驅使所有人對日常工作保持承擔、警覺與挑剔的精神，當檢測到浪費時，採取行動消除和預防浪費成為一種自然反應。

價值流對應圖（*Value Stream Mapping*）是一種辨識結構性浪費的方式。從「概念」到「現金」的過程中的所有步驟和階段都在時間表上列出，活動可能被標記為「有增值」或「無增值」，沒有直接增值功能的活動，也可以被標記為必要。價值比率是用投入增值活動與浪費活動的時間比例來計算，得出的數字可以作為衡量改進的基準。但是，與所有改進活動一樣，沒有明確的最終目標，也沒有最終狀態，改進本身就是目標。

庫存、在製品（WIP）與工作流（Flow）

精實追求好的連續性和工作流。生產過剩和庫存過多會破壞工作流，並可能會延遲發現和解決品質問題。同時這也是一種不尊重，因為它迫使人們去做實際上可能永遠不會被使用的工作。庫存的成本昂貴，而且讓組織容易遭受損失。

精實透過「即時生產（Just in Time）」模式，只生產即將要使用的材料——當流程中的後續步驟發出拉（*pull*）信號時——來限制「在製品」（和昂貴的庫存）的數量。看板（*Kanban*）的作用是在製造系統中作為實體信號卡，看板應連接到庫存清單，反映庫存水平，僅當材料被使用的夠多並且信號產生時才生產新零件。

1.6.2 實現精實

就像敏捷一樣，許多組織對實施精實感到困難。除此之外，組織也在實施敏捷和精實的結合中苦苦掙扎。

一般來說，公司在表達對「精實」的渴望時會提到管理或組織問題，另一方面，如果他們只是想「採用敏捷」，則大多是指產品開發中的問題。但是，不論敏捷或精實都沒有提供一種神奇的現成解決方案（銀彈）。

不幸的是，精實太常侷限於在用來消除浪費。僅從工具箱中選擇一個元素來使用，是對單一面向過度關注。如果違反了原則本身，並且將「消除」應用於（開除）人員，而不是作為改善流程和結構的手段，只會讓情況變得更加糟糕。廣受歡迎的「削減成本」管理運動傾向於將這種重要的精實實務扭曲為將人們的工作認為是經常性的開支，而不是價值。潛在的信息就是從事這項工作的人是浪費的…而且是可拋棄的。

因為這種普遍的誤解及其對精實的片面觀點，要建立起正確的認識將是一段漫長的旅程，精實主軸在於尊重人員以優化價值和品質。精實更多地是關於人員為了產生效益而共同努力的情境，而不是持續地過

分強調結果和績效。它促使去中心化的決策機制，代替「命令與控制」、大老闆行為、微管理、超額分配和細節指派的艱鉅工作。

還有很長一段路要走，才能從誤解到對理解精實不只是一些實務作法，理解精實是一種思想，在沒有明確標準的最終狀態的情境，人們不斷對日常工作進行反思和自我改善。

敏捷能有幫助。

敏捷和精實之間有非常多的相似處值得探討。某些管理或治理哲學不應混為一談，因為這將產生含糊的混合物，而其各自的獨特風味以及益處也會消失。但是，就敏捷和精實而言，我相信不僅可以將精實和敏捷結合起來，而且將產生更強大的結果。

精實和敏捷是真正的調合哲學，精實興盛於一種有力但典型思維，而敏捷則具有獨特的觀點，這些觀點不僅非常符合精實的主要原則，並且在針對產品交付的目的進行了非常具體的實施。

1.6.3 敏捷和精實的調和哲學

《精實敏捷的調合哲學 (The Blending Philosophies of Lean and Agile)》同時也是我針對該主題發表的一篇更詳細的論文的標題 [Verheyen, 2011]。在這裡，我僅介紹一些敏捷與精實一致的明確策略：

- **潛在無用的庫存**：在軟體開發和其他形式的複雜工作中，產出詳細需求、完整的細節計劃、設計等被視為一種責任，但他們其實不是資產，而是可能不會被使用到的工作。敏捷避免預先設想出任何可能的細節，而是替換成逐漸顯現與立即需要的工作。如果一項工作在一段時間後才可能會被實現，就表示這工作很有可能不會被執行，就算被執行了，在此期間期望的細節可能已發生變化。透過實作和釋出後獲得的經驗，也可能因此發現了另一個更好的處理方法。我們只詳細規劃即將到來、順序最高的工作，因為這是即將在下一步進行的工作。但即便如此，團隊也只會投入

規劃他們認為一個迭代可完成的工作量，並在實際執行時，最高以每天為單位，採用漸進式學習和持續改進的方式進行。

- **部分完成的工作**：尚未完全完成的工作，「馬上就好了，我只需要再多一點點時間」的這種工作，是一種重要的已知浪費類型。在敏捷過程中，每次迭代的目標是生產一個可使用的產品。看得到的結果中不包含未完成的工作。限時（Time-boxing）是一種時間管理技術，用以協助團隊專注於完成當前的工作。
- **功能的使用**：研究表明，用傳統方式建置的（軟體）產品中幾乎只有 20% 的功能會被定期使用 [Standish, 2002; Standish 2013]。無論是對開發方或是維護方，未使用或未被充分使用的功能都代表浪費了大量的精力和預算。透過了解並代表客戶和用戶的人員積極協作可防止產生不必要或無價值的需求，並幫助團隊專注於可能真的會被重視的最小功能集。關注「想要的」需求，不僅節省了開發預算，而且還可以保持較低的未來維護和維運成本，並且迭代增量流程可以根據對交付價值觀察到的有效的回饋（或沒有回饋）以及新的價值機會，來定期調整產品。

敏捷對於持續改善有著明確的策略，因此利用改善的精神：

- 敏捷團隊的工作計畫每天會檢查並更新。
- 在迭代結束時的產品版本將會被驗證（無論是已釋出還是處於可釋出狀態），以便收集回饋、評論、改善和加強。
- 團隊的工作、協作、溝通和執行的過程，都定期透過迭代回顧被驗證。

敏捷是對整體的優化，透過要求客戶或客戶代理人表達和排序工作，積極參與開發過程，釐清功能和功能取捨，在實作中也如此。團隊擁有的所有技能，是為了在一次迭代之內將最多的概念、選項與需求實作出來。

敏捷透過避免傳統的階段交接和外部決策等等待活動來優化價值流，從而縮短週期時間。沒有宏觀的階段交接，即跨部門和組織的交接，這通常發生在將工作階段以不同的專業切割的循序型工作組織。同時團隊承擔共同責任，也沒有微觀的階段交接，即團隊內個人之間的責任移交。

總的來說，敏捷的策略和原則與所有主要的精實原則都一致，甚至可以加以相互利用，如下表所示：

| 精實 | 敏捷 |
|-----------|-------------------|
| 對人的尊重 | 自組織團隊 |
| 改善 | 檢視與調適，縮短回饋週期 |
| 避免 / 消除浪費 | 拒絕沒用到的規格、架構、或基礎建設 |
| 拉式庫存（看板） | 依團隊產能規劃工作 |
| 視覺化管理 | 資訊輻射站 |
| 內建品質 | 完成的定義、開發標準 |
| 客戶價值 | 商業面的主動協作 |
| 整體最佳化 | 完整團隊合作（含利害關係人） |
| 快速交付 | 具有工作增量的限時迭代 |
| 導師式管理者 | 服務式領導 |

圖 1.6 精實與敏捷原則的一致性