

前言

我很高興能有這個機會，為你帶來這 97 篇關於 Scrum 的文章。這不是一本學習 Scrum 的書；相反地，這本書對 Scrum 框架的規則、角色及其目的提出了一些見解。此外，這本書也提供與 Scrum 搭配運用的戰術、策略和模式，以及一些實戰故事和超越 Scrum 的觀點。

儘管 Scrum（於 1995 年問世）逐漸成為最常被用來定義敏捷方法（於 2001 年問世）的框架，但對於我們 Scrum 實踐者而言，仍然存在許多重大挑戰。我們不僅在持續釐清許多對於 Scrum 的誤解，也一直在尋找最佳的方式和管道，以提供資訊給 Scrum 新手及想更了解 Scrum 框架的人。

由於 Scrum 的廣泛採用，全世界已有數百萬的實踐者，可以提供豐富的知識和現實生活中的專業經驗。於此同時，有不計其數的 Scrum 新成員和組織正在尋求見解、建議和故事；也有實踐者渴望獲得更深的洞見。透過這本書，我們希望能將經驗豐富的實踐者與知識尋求者連結在一起。

Scrum 的單純是眾所皆知的，但它並不容易。Scrum 是一個旨在解決錯綜複雜挑戰的單純框架，而錯綜複雜的產品開發是此類挑戰的重要子議題。Scrum 給各種實踐和策略應用留了很大的空間，並且可以涵納許多此類實踐和策略。要描述 Scrum 可以採取的多樣形式，還有什麼比從頂尖 Scrum 實踐者那裡，收集和編寫可用故事更合適的呢？

在這本書裡，你將了解經驗豐富的實踐者如何應對問題和挑戰，以及他們認為你該知道的關鍵要點。

站在他們的肩膀上，以他們的見解為基礎來提升你對 Scrum 的了解。透過應用和調整這些實踐者的方法來自我啟發，找到屬於自己的 Scrum。

這 97 篇論文所展現的集體智慧，代表著各式各樣的想法。你可能不喜歡每篇文章，也可能不會同意所有觀點；你可能會看到矛盾、反常觀點和一些模糊形式。歡迎來到 Scrum 幫助我們應對的錯綜複雜世界：一個難以界定黑白的世界。希望你能在這裡找到許多價值，以更好地應對特定情況的錯綜複雜性。

本書的編排方式

除了追著人們跑、收集和閱讀文章，以及為作者提供編輯建議外，我還喜歡尋找共同點並分享延伸主題。在完成對文章進行排序的工作後，我最終將這 97 件事歸納為 10 個主題。儘管在書中創造閱讀流程是一種方式，但幸運的是，每一篇文章都不侷限於一個主題，並且可以獨立閱讀：

第一部分：開始、採用、重複

在第一部中，你會學習到在採用 Scrum 的過程中，值得考慮和反思的 11 個重要小細節。因為採用 Scrum 不僅僅是引入 Scrum 的一次性嘗試，也是思考、重新思考和發現的持續練習。

第二部分：產品創造價值

在第二部中，你將學到的 11 件事，強調了 Scrum 在產品和服務上的重點，以及值大於量的原因。因為在需求不穩定和技術不斷發展的錯綜複雜世界中，「產品」提供了最小形式的穩定性，使你得以運用 Scrum 來組織工作。

第三部分：協作是關鍵

在第三部中，你將學到的 10 件事，解釋了為何 Scrum 的關鍵在於團隊協作，且溝通和互動在其中至關重要。因為在錯綜複雜又多變的環境中創造、維持和發展錯綜複雜的產品和服務，需要集體智慧、技能和專業知識。

第四部分：開發是多面向的工作

在第四部中，你將學到的 12 件事，舉例說明了 Scrum 中的「開發」如何包含所有工作，以在衝刺結束之前創造出有形、可展示的工作成果。因為錯綜複雜產品的開發（在通常錯綜複雜的情況下），需要的不只是技術生產工作（例如只寫程式）。

第五部分：事件與會議不同

在第五部中，你將學到 10 件事，以幫助你掌握 Scrum 事件的協作目的和性質，並了解它們的前瞻性是如何與過去、報告或狀態更新有所不同。因為俗稱的 *Scrum* 會議，實際上是為檢視和調適提供特定機會的事件。

第六部分：精熟度至關重要

在第六部中，你將學到的 12 件事，說明了精通 Scrum 須掌握的關鍵，以及必須尋找和注意的事項。儘管 *Scrum Master* 作為流程管理者相當重要，但精熟度對任何角色都是個關鍵。

第七部分：人類都太人性化了

在第七部中，你將學到的 8 件事，側重在 Scrum 中人類工作的一面，甚至會涉及到人類學和神經科學。人不是資源（機器人、齒輪或可替換的機械零件）。開發是由人來完成的，因此往往會創造出人類的工作。而人類畢竟是……人。

第八部分：價值驅動行為

在第八部中，你將學到的 6 件事都根基於一種理解：Scrum 所牽涉的遠不止於規則。根據設計，Scrum 是一個不完整的框架，而該框架邀請人們參與、互動和協作。因為 *Scrum* 是規則、原則和價值的框架，而價值會驅動行為。

第九部分：組織化設計

在第九部中，你將學到的 9 件事，包括觀點、實際案例，以及 Scrum 對組織結構影響力的教訓。儘管這不一定與規模相關，但是在大型組織中，組織的影響更為明顯。因為若要引入 *Scrum*，將無可避免地影響到組織和現有的組織結構。

第十部分：Scrum 的延伸

在第十部中，你將學到的 8 件事，包含 Scrum 本身、其用法，以及在 Scrum 相關和無關領域中的潛在使用方式；其中還包括一些可以追溯到 Scrum 誕生前的草根故事。因為 *Scrum* 實踐者若要幫助塑造 *Scrum* 的未來，需要將想像力與歷史脈絡相結合。

在本書的最後有一個 Scrum 詞彙表，它以最單純的方式條列並解釋了本書中所使用的術語。

在整本書中，你會看到許多從《Scrum 指南》(Scrum Guide, <https://www.scrumguides.org>) 摘錄的內容，其均取自 2017 年 11 月版。

特別感謝

我很幸運能擁有一個很棒的家庭，啟發並鼓勵我去做我想做的事，無論我最終會成就什麼。謝謝我的妻子和我們的三個孩子；因為有他們，我才會有今天的成就。

我很感謝 Martine Devos 幫助我與一些傑出的貢獻者取得聯繫；此外，我更感激她默默的信任和鼓勵，這對我來說意義重大。因為 Martine 一直是 Scrum 界的先驅者，且持續努力不懈地推廣 Scrum。

在使用 Scrum 的那些年裡，我與許多人和組織一起工作。人們在這過程中來來去去；在這些人之中，我要特別感謝傑出的 Hans Lindhout。他讓我體驗到了真正的合作互惠，並使我的工作動機得以超越商業目的。希望這個世界能有更多像這樣的人。

本書有超過 68 位實踐者花費了許多心血，撰寫一篇或多篇與 Scrum 相關的文章。我們並不是因為他們的頭銜、階級或職位才邀請他們的，之所以邀請他們，是因為他們有寶貴的洞見，可以與像你這樣的實踐者分享。我衷心感謝他們每一個人。

感謝購買本書並閱讀到這裡的你。如果你能運用 Scrum，並分享和散播你如何利用 Scrum 來克服特定挑戰，我會更加感激。請持續啟發其他的 Scrum 實踐者。

多虧有 O'Reilly Media 的強大團隊，這本書才得以存在；非常感謝所有相關人員。我尤其要感謝 Chris Guzikowski 和 Ryan Shaw 發起這項辛苦的計畫，並感謝 Corbin Collins 在過程中堅持了下來。

祝你閱讀愉快，還有……請持續使用 Scrum。

—Gunther Verheyen，獨立 Scrum 看守者
寫於比利時安特衛普

沒人告訴你的 Scrum 五件事

Marc Loeffler



如果你正在考慮使用 Scrum 來改變你的工作樣貌，以下是一些沒人告訴過你關於 Scrum 的事。

1. Scrum 無法解決你的問題

有些人希望 Scrum 能夠神奇地解決問題，就好像它是什麼靈丹妙藥一樣。但是像 Scrum 這樣的敏捷流程，其實更像是無時無刻在背後觀察你的婆婆——她指出了你所有的問題和錯誤，將它們全部攤在陽光底下。最後你還是必須靠自己來解決它們，並完成艱苦的工作。

2. 如果你只是遵循流程，Scrum 並不會給你帶來什麼好處

我見過完全按表操課的 Scrum 團隊。他們每天舉行立會、衝刺規劃會議、衝刺審查、回顧，甚至包括產品待辦清單精煉會議。但他們從 Scrum 那裡獲得的好處卻微不足道；因為他們沒有認知到，敏捷的重點是在思維而非作為。要實踐敏捷，就需要改變思維，以適應過程中的變化。只有擁抱敏捷宣言和 Scrum 價值觀背後的 12 條原則，才能將 Scrum 的潛力完整發揮出來。

3. 沒有「Scrum 開關」這種東西

將所有員工送去接受為期兩天的 Scrum 認證培訓，並不足以成就一個 Scrum 公司。並沒有這種「Scrum 開關」，能夠讓你的公司在一夕之間切換到敏捷模式。Scrum 的流程說明看起來很容易，但事實卻恰恰相反——採用 Scrum 很難。有很多東西要學，包括：最大化未完成的工作、減少現有會議、在每個衝刺結束時交付產品的生產就緒版本、解決棘手的組織問

題等等。是的，你可以「切換」到 Scrum 模式，但這將是一個循序漸進的過程。你可能需要花費數個月到甚至數年，並且永遠不會有結束的一天。

4. 轉型為 Scrum 也意味著組織轉型

大多數的敏捷轉型計畫都始於產品開發，但不應只停留在那裡。如果組織中的其他成員對於該計畫毫不在意，那就像在沙漠中種一朵花——它會慢慢死去。引入 Scrum 一定會帶來整體或部分的組織轉型。如果你還沒有準備好面對這一點，那就代表你還沒有準備好使用 Scrum。

5. Scrum 不會比較快

即使「衝刺」一詞暗示著 Scrum 很快，但先不要抱太高的期待，至少在最初先保守點。如果要在每次迭代結束時發佈生產就緒版本，你將無法在相同的時間內交付更多功能。Scrum 確實能幫助你縮短產品上市時間，但是透過以下不同的方式：

- Scrum 會逐漸增加未完成的工作。關注客戶真正的需求，而不是打造沒人知道如何使用、許多功能也不被需要的混亂產品。當然，這將造就更少（且更清晰）的產品待辦清單和更快的交付，從而最終縮短產品上市時間。
- Scrum 需要從一開始就重視品質。這將為你爭取空間，讓你能專注於打造創新功能，而不是浪費時間和精力在漫長的臭蟲修復上。同時，你需要維護的時間也將大幅減少。

思維遠比實踐重要

Gil Broza



你的每日 Scrum 像是狀態更新會議？

管理者與利害關係人把產品待辦清單視為專案計畫的一部分？

你的 Scrum Master 把大部分重點放在管理任務，並確保流程合乎規範？

如果你對這些問題的回答是肯定的，那麼讓我來猜猜你 Scrum 的其他症狀。團隊回顧不會帶來很大的改善；成員之間的協作很少；對「完成」的定義也沒有明確的共識。如果你身處的是軟體開發團隊，則你們的自動化測試覆蓋率很低。你們只是在表面上（且很少）重構程式碼，而生產部署的過程總是很謹慎，且需要大量的計畫。你們雖然似乎都有遵循 Scrum 的流程，但卻可能感受不到太多敏捷帶來的優點。怎麼會這樣？

要了解狀況，我們需要先知道 Scrum 團隊為何遵循特定的流程、會議、角色和產出物。首先，讓我們來看看當戰略被有效運用時，背後所遵循的原則。

想一想每日 Scrum。從敏捷的角度來看，這是最有效的方法，使團隊成員可以相互協作和自我組織以完成重要的工作。他們保持透明，充分且真誠地參與專案，並且在心理上是感覺安全的。

產品待辦清單是一種潛在重要工作的有序主列表。當正確使用時，這個單純的產出物能幫助團隊將工作專注在有意義的成果上。他們可以輕易確定自己的做法是否有效，並能將許多與工作項目相關的決策（例如細節、拆分和驗收測試）推遲到最後須兌現的時刻。

Scrum Master 是一個僕人式領導（*servant-leader*），目標在幫助團隊成長為一個堅實的敏捷團隊。這個角色會透過創造安全、信任和尊重的環境，並促進溝通、協作和透明性，來達成此目標。

Scrum 實現了上述原則（還有一些其他的），因為它們支持四個敏捷價值：調適性、頻繁的價值傳遞、客戶協作，以及以人為本。當人們相信自己實現目標的方式時，他們就會接受這些價值（「為他們優化」，或者讓他們自行引導所作所為）。價值、原則及信念是思維的三個要素。

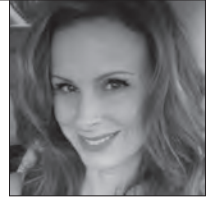
組織在採用 Scrum 時，通常已存在一些既定思維。儘管每個組織的起點不同，但它們的思維卻有一些相似之處，我們可以將其稱呼為「傳統」。它們的價值通常包括：第一次就實現可交付成果、儘早承諾、按時按預算交付，以及遵循標準。它們依照以下原則來對這些價值採取行動：計畫工作並實現計畫、限制變更以及要求簽核。他們由少數人決定工作，讓「資源」（人）相互交接工作，並最大化他們的使用率。在 Scrum 之前，他們透過諸如專案計畫、完成度、頻繁召開狀態會議，和凍結程式碼之類的戰術來實現這些原則。當這樣的組織用 Scrum 策略替換其現有策略，而又不改變其思維方式時，會發生什麼事？

計畫工作、限制變更以及最大化使用率，使每日 Scrum 變成了一個用來確保每個人工作量的狀態會議；由少數人決定工作，使每日 Scrum 變成了一連串與 Scrum Master 的一對一交流。另外，由於安全性和透明性並不會在每日立會時神奇地提升，所以人們會忙於長篇大論地回答「我從昨天到現在完成了什麼？」而不願回答「我遇到什麼困難？」也不願說「我被困住了，需要幫助。」

同樣地，組織的思維將改變其對任何 Scrum 實踐、角色、會議和產出物的使用方式。思維離 Scrum 背後的敏捷價值愈遠，Scrum 戰術就愈被削弱，團隊成員容易感到困惑和迷惑，實際敏捷性也會降低。要實現真正的敏捷，請改變思維，這遠比實踐重要。

事實上， Scrum 本身不是重點

Stacia Viscardi



Scrum 是一個看起來很單純的方法；它告訴我們如何跑衝刺流程、繪製燃盡圖，以及定義「完成」等。但你或許也在某個地方看到有人這麼評論 Scrum：「Scrum 表面上很容易，但實作起來很難。」這是因為從表面上看，Scrum 的構造很單純，就像排列整齊的磚塊。但稍微深入一點，你會看到管理錯綜複雜且混亂情況的盤根錯節、人與動機的基石，以及殘留的精實精神地基；它們就像在現代街道下奔流的古老溝渠。Scrum 最終是一個讓我們得以檢視現實的陽台，而無論那個現實是什麼；人們接著可以根據該現實的新改善版本，作出更高品質的決策。

不久前，我在推特上說：「Scrum 是一種系統整合測試，目的在確認組織是否能在衝刺中實現價值。該測試會揭露系統的錯誤或障礙，然後我們必須進行修復。」這些「臭蟲」（路上的顛簸、障礙）是阻礙組織交付真正重要內容（使用者、客戶，和 / 或市場滿意的功能）之關鍵。

如果你在一夕之間開始實踐 Scrum，去揭露現實，去經歷這項巨大的考驗，會遇到一些反彈也是正常的。就像瀑布式專案一樣，在可怕的「程式碼凍結」里程碑之後，開發人員也不願將程式碼交給測試人員。團隊和組織抗拒去面對，當他們將揭露出來的現實交給這個名為衝刺的透明性測試後會發現什麼。

Scrum 對組織決策結構的影響也使實踐 Scrum 變得困難。現在，產品負責人負責產品內容與目的，開發團隊負責交付方式與速度，而 Scrum Master 是促進者和「障礙揭露者」。透過建立這些角色和單純的機制，我們得以將錯綜複雜的專案拆解成多個衝刺並逐一克服。我們也得以「使軟體開發再次成為一種專業」；從中人們可以得到信任和尊重、探索解決

方案，並最終為他們的工作感到自豪。**Scrum** 是一個以人為本的框架，這是其固有的五大價值之一。此價值對於思想傳統的管理者來說通常較難掌握，因為他們錯誤地將方法論視為一種「解放之法」。

這就是為什麼我會說 **Scrum** 本身不是重點；重點在於使用框架來發現從前未知的事物。因此，現在就開始面對揭露出來的事物並從中調適吧！利用優先排序、時限和角色等界限，來幫助團隊和組織了解所發現的事物。人們對失敗和未知的恐懼，有時會阻礙他們做正確的事；因此一個組織通常不會輕易接受失敗。但我希望隨著時間的推移，透過使用 **Scrum** 或任何經驗主義方法，他們會將面對失敗視為探索之旅的重要過程，並不斷從中學習改進。

Scrum 很單純， 直接用就對了！

Ken Schwaber



Scrum 是一種思維、一種將錯綜複雜又混亂的問題轉變為可處理狀態的方法。我和 Jeff Sutherland 認為 Scrum 根基於以下支柱：

1. 自我組織的小型團隊
2. 精實原則
3. 經驗主義——透過頻繁的檢視和調適來指導團隊工作，以儘可能地取得最大成功結果

《Scrum 指南》是一門知識體系，明確定義出什麼是 Scrum（以及在預設情況下什麼不是）。《Scrum 指南》不會告訴你如何使用 Scrum、如何實行 Scrum，或如何使用 Scrum 來打造產品。

人們透過參加課程和會議、閱讀書籍和部落格等方式，來了解 Scrum 的含義以及如何使用它。但人們主要是運用對 Scrum 的理解，嘗試根據願景、概念和期望來創造出有用的東西。隨著他們的努力，Scrum 開始變得有意義；Scrum 幫助他們管理成果。但當人們嘗試使用 Scrum 時，他們了解到 Scrum 的困難在於擁有共識；這些共識包括期望、可行計畫，以及哪些具備的技能可使團隊創造、合作並達到最佳狀態。

在 2009 年，我意識到我們已經打破了瀑布式模型。人們（大體上）理解我們的「敏捷、輕量」方法可行，並且適用於世界上新興的錯綜複雜性。但是就像在玩電話捉迷藏^{譯註}一樣，人們對 Scrum 有很多不同的解釋。有

^{譯註} 電話捉迷藏 (telephone-tag game)，意指雙方打電話給彼此，對方卻總是剛好不在，因此只能不斷留言請對方回電的狀態。
在此用於形容大眾對於 Scrum 的解釋眾說紛紜而沒有共識。

時，這是由於溝通不順暢、指導不適當或其他商業原因所致。那些認為 Scrum 會告訴他們如何打造產品來滿足其需求的人，會覺得 Scrum 太弱了；因為它沒有明確說明該如何做到。

沒錯。正如我經常說的：Scrum 很簡單，但使用 Scrum 來解決問題非常困難。

因此，當我在 2009 年創建 Scrum.org 時，我撰寫了 Scrum 的定義。內容雖然簡短，但保留了我和 Jeff 的重要想法和知識。我確保定義中保留了 Scrum 作為框架的角色，並避免任何包含意見、依賴情境的實踐，以及將其限制為僅適用於某些應用的內容。Scrum 是一個框架，而不是方法。

這是第一份《Scrum 指南》，是具權威性的知識體系。指南中未包含、或與指南相反的任何內容均不是 Scrum。Jeff Sutherland 後來和我一起修訂和維護這份指南。

《Scrum 指南》是根據我和 Jeff，以及其他所有嘗試使用 Scrum 者的工作經驗撰寫而成的。從那時起，我們就對其進行了許多修訂。《Scrum 指南》除了提供對 Scrum 是什麼的石蕊測試之外，沒有任何商業目的。我和 Jeff 深深感激幫助我們翻譯以及維護指南的人們。

請記得：Scrum 很單純。不要再煩惱該如何精進 Scrum 來使其更完美，因為永遠不會有那一天。無論如何，世界上有太多錯綜複雜又混亂的情況，是你能發揮技能幫助其他人解決的。我們不必浪費時間自我設限。

在你的 Scrum 中， 從問為什麼開始

*Peter Goetz
& Uwe Schirmer*



「這樣做符合 Scrum 準則嗎？」是我們經常被問到的一個問題。在這個問題之前，詢問者通常會先具體地描述團隊的實踐方式；他們想知道自己是否有正確地運用 Scrum（是否有遵循《Scrum 指南》）。

儘管這個問題很合理，但我們會先試圖理解為什麼詢問者想問這個問題。團隊和組織經常專注於 Scrum 的機制，而忘記了其真實目的——通常是為客戶、利害關係人和自己創造並增加價值。

不要誤會我們的意思：《Scrum 指南》對我們很重要。它描述了解決錯綜複雜調適問題的核心要素，以及它們如何共同發揮作用。我們很看重這份指南，因為它可以幫助我們專注於協作和創意性地解決問題，而不僅僅是閱讀和應用《Scrum 指南》。

Scrum 存在的其中一個原因在於：「在每個衝刺結束時，交付潛在可發佈的『完成』產品增量」（<https://www.scrumguides.org>）。Scrum 中的所有要素都支持這一點。角色將工作人員劃分並使其共同當責（accountability）；產出物建立了迭代和逐步交付可發佈增量所需的最小透明性。而根據從產出物中收集到的訊息，事件提供了檢視和調適的機會，並建立出一種定期獲取各方利害關係人回饋的節奏；這種節奏改善了 Scrum 團隊的工作流程。Scrum 價值觀描述了一種共同思維，這在持續審查和改善合作方式時非常有幫助。

你可以機械式地遵循《Scrum 指南》中所述的規則，任命產品負責人、Scrum Master 和開發團隊；遵循衝刺流程並邀請團隊（有時包含利害關係人）參加事件；設定產品待辦清單和衝刺待辦清單管理工具，並請 Scrum 團隊開始使用。但如果使用 Scrum 背後的原因沒有被理解和接受，那麼

真正的改善就不會發生。**Scrum** 團隊甚至可能會對這些新規則感到惱火，因為他們看不到規則背後的意義。最壞的情況是，相較於舊的工作方式，這些新規則所產生的壓力反而使他們精疲力盡。

因此，我們建議採用另一種方法。我們鼓勵組織先談論他們面臨的挑戰，以及他們目前解決這些挑戰的策略為何沒有幫助，或有哪些需要改進的地方。我們接著可以思考克服這些限制的方法，並探索 **Scrum** 可能提供什麼幫助。只有這樣，引入 **Scrum** 才會有意義。

在運行 **Scrum** 的旅程中，有兩個問題應該作為方向的指引：

1. 我們能否在每段衝刺結束時，提供有價值且可發佈的產品增量？
2. 我們如何改善我們的合作方式？

這些問題讓我們不再只將焦點放在 **Scrum** 機制上；我們學習如何將 **Scrum** 用於更明確的目標和問題上。我們將持續改善我們的技能和工具，以逐步實現這些目標。這將增加我們的動力，並帶來更多快樂和更人性化的工作環境。

最後很可能地，我們的工作方式仍將遵循《**Scrum** 指南》。但我們這麼做不是因為《**Scrum** 指南》上這麼寫，而是因為這麼做合乎道理。

調適前先採用

Steve Berczuk



Scrum 是一個相當小的流程框架，能夠最好地調適團隊工作。採用 Scrum 的團隊有時會嘗試將調適性的概念延伸來「調適」Scrum 流程，使其僅成為現有（通常是傳統）流程之上的裝飾表層。團隊須將 Scrum 流程的原則和價值觀內化，才能發揮調適性的最大效果；而要理解不同的工作方式則需要實踐和經驗。

在採用 Scrum 期間不成熟調適的一個常見例子，是團隊會看著 Scrum 事件並決定不照單全收。他們將衝刺規劃會議縮短或只邀請部分團隊參與，導致衝刺待辦清單的工作量超過了團隊過往經驗中所能處理的。每日 Scrum 變成狀態會議；或者更糟的是，其他狀態會議仍保留在行程表中。衝刺審查和回顧會議被草率舉行，甚至跳過。簡而言之，你的事件可能只是虛有其表，其中並不包含任何 Scrum 價值觀。這是一個很糟的做法，因為它最好的結果是團隊無法從 Scrum 中獲得任何價值，而最壞的結果是這種偽裝或混合的方法反而會增加不必要的成本，導致效能下降。

發生這種情況的原因因組織而異，但是有兩個因素幾乎總是扮演要角。首先，任何改變都是困難的、是需要努力和學習的，但人們往往在壓力大時容易回到熟悉的舊模式。另外，Scrum 學起來很容易，但要掌握很難；Scrum 的奧妙之處不僅僅是舉行一些 Scrum 事件而已。團隊中的每個人都必須接受訓練，以確保他們能盡其所能地幫助流程順利進行。

在調適 Scrum 之前，請先採用 Scrum 的實踐方法；這些方法將幫助你理解並內化新的工作方式。抵抗改變 Scrum 規則的誘惑，直到團隊對這些方法如何增加價值與支持 Scrum 原則有所了解。請「照本宣科」地遵循 Scrum 流程至少三到五個星期（你可以參考《Scrum 指南》，或其他任何優秀的 Scrum 書籍），並確保留下時間進行衝刺回顧。在回顧中探索流程如何影響協作和交付，這使得人們在看到透過 Scrum 框架實現的行為後，有一個「啊哈」的頓悟機會。

當你真的相信自己已了解 **Scrum** 價值觀，你就準備好調適 **Scrum** 了。由於 **Scrum** 與大多數組織習慣的工作方式不同，因此大多數團隊需要實踐方法以了解其價值。儘管對於成功採用而言不是必需的，但擁有一個良好的基準來調適會是最好的。

定期回歸到可能有用的 最單純方法

Todd Miller



簡化將產生更豐富的結果。

— Yvon Chouinard¹

許多年前，我辭掉一份 Scrum Master 的工作，成為另一家公司的 Scrum Master。我當時要離開的 Scrum 團隊非常傑出；他們合作順暢，逐步導入 Scrum 有成，實踐也頗具成效，並且正在開發出色的產品。但是我已經準備好迎接新的挑戰。

我成為新 Scrum 團隊的一員，他們正準備開發新產品。在加入團隊的最初幾天，我們花了很多時間討論 Scrum。我們在討論的過程中建立共識、使用相同的語言，同時逐漸了解彼此。每個人都熱切地參與並分享他們對於利用 Scrum 來開發新錯綜複雜產品的期待。

同時，我們的產品待辦清單準備充分，足以使我們放心開始第一個衝刺。在第一個衝刺之前，我們舉辦了為期一天的團隊啟動日。我精心策劃了當天的議程，包括分享和推廣我堅信有用的實踐方法。我想將我從以前 Scrum 團隊中獲得的洞見和經驗傳承給這個團隊；我竭盡全力地打造並催化一整天的緊密協作。

最初幾個議程進行得很順利。Scrum 團隊討論了當前的產品待辦清單，以確保對前幾個衝刺的工作排序有充分的了解，並針對「完成」的定義有了精采的結論。但是，當我介紹下一個主題「建立團隊工作協議」時，事情

¹ Yvon Chouinard, *Let My People Go Surfing: The Education of a Reluctant Businessman* (New York: Penguin Books, 2016).

開始發生奇怪的變化。團隊成員開始質疑這個做法的必要性，畢竟我們合作的第一週進展順利。當我感覺到房間中的緊張情緒迅速提升時，我放棄了這個主題，繼續進行下一個議程項目。但很意外地，我還是感受到相似的抗拒；更糟糕的是，隨著我嘗試推進每個主題，這股抗拒都在不斷提升。

然後，突然有人問了一個出乎我意料的問題：「你為什麼要把事情變得如此複雜？」我唯一的回答是：「這些實踐方法對我的上一份工作很有效。」這個回答不僅難以說服團隊，連我自己都不滿意。現場充滿了不安的氛圍；但我決定放棄我對啟動日的雄心計畫，宣布我們已準備好開始第一個衝刺。

在我最初感到困惑時，當時的情況實際上對我來說是一次很好的學習。在錯綜複雜的工作領域中，你無法保證在一種情況下有效的方法，在另一種情況下依然有效。我的新 Scrum 團隊後來順利地完成許多任務，並在過程中探索並修改實踐方法。

這次經驗至今仍提醒著我，在錯綜複雜工作中最重要的精神是經驗主義——經常檢視和調適透明性的過程。Scrum 中的經驗主義不僅限於探索產品功能，還包括在 Scrum 中應用的實踐方法。這不僅適用於新的 Scrum 團隊；舊有的 Scrum 團隊也經常發現自己被眾多的補充實踐方法所包圍，從而對他們向內看（檢視性）、改變方向（調適性）或保持公開（透明性）的能力產生負面影響。

請記得要經常檢驗你在 Scrum 中所使用的實踐方法是否合理，且其存在能優化經驗主義行為，而不是阻礙它。

Scrum 適用於 多據點的開發模式嗎？

Pete Deemer



（劇透注意）是的！**Scrum** 非常適合多據點開發！但是，**Scrum** 並不是能使多據點開發神奇運作的靈丹。

Scrum 是一組非常單純的實踐方法；當這些方法一同發揮效用時會產生高透明性，讓我們能更加清晰地看見實際情況。**Scrum** 將我們在一到四個星期內可以完成的事情透明化；**Scrum** 使我們能夠清楚地看到產出的品質和正確性，以及能夠完成的工作量；**Scrum** 使我們面對實踐方法的有效性，包括功能障礙模式和我們正在犯的錯誤。最重要的是，**Scrum** 提供了一個機會，讓我們能善用這些洞見，並在下一個衝刺中改善。因此，「**Scrum** 非常適合多據點開發！」這句話的意思是，**Scrum** 能夠將我們遇到的所有功能障礙和效率低落問題，清楚地展示在我們面前。而分散式專案往往會面臨很多這種問題！

軟體開發從本質上來說是一種活動，它吸收了人類思想和邏輯推理的產物，並使其以程式碼的形式不斷重複。對於開發軟體的團隊來說，產出的關鍵驅動力在於他們透過共識、協作和溝通來表達思想的能力。在多據點專案中，這項活動因為面臨多種分隔形式（物理分隔、時區分隔、文化和語言分隔，以及語言和文化障礙帶來的問題）而變得特別困難。多據點團隊完成的第一個衝刺經常像是功能失調的豐富馬戲團，而可能的結論往往是「這根本行不通」；但這個認知是邁向成功的關鍵第一步。下一步是列舉所有運作不順的項目，並制定措施以在下一個衝刺中有更好的成效。

舉例來說，產品增量可能在第一個衝刺中沒有達到「完成」的定義標準。在衝刺回顧中，**Scrum** 團隊找出主要的根本原因，例如：缺乏對衝刺目標的清晰共識、團隊內部缺乏溝通導致缺乏協調，或者團隊成員之間缺乏信任。接著在第二個衝刺中，團隊對每個問題都採取了相對應的措施（有

關措施建議，請參閱《分散式 Scrum 入門》(Distributed Scrum Primer，<https://oreil.ly/iOKbL>)，而這些問題在該衝刺中有了些微的改善。常見的情況是，在接下來的衝刺中仍會有其他問題顯現出來，並且每次都會發現全新的改善方式。Scrum 團隊會在每個衝刺中逐漸成長，發展出愈來愈好的工作方式。

Scrum 會「神奇地」使每個分散式 Scrum 團隊發揮效用嗎？不會。儘管有些方法這麼保證，但事實上沒有一種方法具有這種能力。Scrum 的突破性創新在於它否定基於方法論來獲得成功的想法。相反地，Scrum 是基於檢視性和調適性而構建的單純框架，可在創造改善機會的同時，根據給定情況的獨特環境和特徵來創造持續的透明性。唯一的問題是：團隊願意且能夠接受多大膽的行動！