

Ken Schwaber & Jeff Sutherland

# Scrum 指南

Scrum 權威指南: 遊戲規則

2020 年 11 月

繁體中文

## Scrum 指南的目的

我們在 1990 年代初期開發出 Scrum。為了協助全世界的人們了解 Scrum，我們於 2010 年間撰寫了 Scrum 指南的首版。自那時起，我們透過小的功能更新對 Scrum 指南進行了演進。我們是 Scrum 指南的共同後盾。

Scrum 指南包含了 Scrum 的定義。為了實現 Scrum 中的全部價值和成果，框架內的各種元素都具有其特定用途。改變 Scrum 的核心設計或 Scrum 的各種理念，遺漏其中任何元素，或是不遵照 Scrum 的規則，是在掩蓋問題，並限制了 Scrum 的各種好處，甚至可能使其變得毫無用處。

我們關注到 Scrum 在錯綜複雜（complex）的世界中應用越來越廣泛。我們很榮幸地看到 Scrum 在許多本質上錯綜複雜（complex）的工作領域中被採用，超越了原本 Scrum 的起源領域 -- 軟體產品開發。隨著 Scrum 的應用範圍逐漸擴大，developers、研究人員、分析師、科學家與其他專家都能以其工作。我們在 Scrum 中使用「developers」一詞不是為了排除其他使用者，而是為了簡化統稱。如果您從 Scrum 當中獲得價值，那麼您可以將自己視為其中一員。

在使用 Scrum 時，可能可以找到、應用和設計符合本文所描述的 Scrum 框架內的模式、流程、見解等。它們的描述超出了本指南的宗旨，因為其中的差異跟您的 Scrum 使用環境有關。這些使用 Scrum 框架內的戰術技巧有很大的變化，因此不在此描述。

Ken Schwaber & Jeff Sutherland 2020 年 11 月

© 2020 Ken Schwaber and Jeff Sutherland

This publication is offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Scrum Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

使用本 Scrum 指南，您認可以及同意以創作共用的署名相同方式共享的條約。

Scrum 指南的目的	1
Scrum 的定義	4
Scrum 的理論	4
透明性 (Transparency)	4
檢視性 (Inspection)	5
調適性 (Adaptation)	5
Scrum 的價值觀	5
Scrum Team	5
Developers	6
Product Owner	6
Scrum Master	7
Scrum 事件	8
The Sprint	8
Sprint Planning	9
Daily Scrum	9
Sprint Review	10
Sprint Retrospective	10
Scrum Artifacts	11
Product Backlog	11
承諾：Product Goal	11
Sprint Backlog	12
承諾：Sprint Goal	12
Increment	12
承諾：完成之定義	12
結語	14
致謝	14
人員	14
Scrum指南的歷史	14
致謝繁體中文譯者	14
2020 年 Scrum 指南版本更新說明	15

甚至更少的規定指示	15
一個團隊，專注於一個產品	15
加入 <b>Product Goal</b>	15
<b>Sprint Goal</b> 、完成之定義和 <b>Product Goal</b> 有了歸宿	15
自我管理（ <b>Self-Managing</b> ）超越自我組織（ <b>Self-Organizing</b> ）	15
三個 <b>Sprint Planning</b> 的主題	15
擴大讀者範圍，全面簡化用語	15

## Scrum 的定義

Scrum 是一個輕量化的框架，透過提供針對錯綜複雜（complex）問題的調適性解決方案，來幫助人們、團隊與組織產生價值。

簡而言之，Scrum 需要一位 Scrum Master 來營造一個環境：

1. 由一位 Product Owner 將錯綜複雜（complex）問題所需之工作事項整理排列成為一份 Product Backlog；
2. Scrum Team 在一個 Sprint 中將所挑選的工作轉化為價值的 Increment；
3. Scrum Team 和利害關係人（stakeholders）檢視結果，並為下一個 Sprint 進行調整；
4. 重複。

Scrum 是易於理解的，試著原封不動地應用它，然後看看 Scrum 的哲學、理論與架構是否可以幫助你達到目標與創造價值。我們刻意的讓 Scrum 框架留白、不完整，而只定義了實踐 Scrum 理論時所需要的部分，Scrum 是建立在 Scrum 使用者的群體智慧之上的，Scrum 指南的規則提供的是對這些人之間各種關係與互動的指引，而不是給予詳細的使用說明。

在 Scrum 框架中可以使用各種不同的流程、技術，與方法。Scrum 將現行的實務做法包裝進來，或是使其變成非必要的。Scrum 凸顯當下關於管理、環境和各種工作技術的相對成效，以便可以對其進行改善。

## Scrum 的理論

Scrum 建立在經驗主義（empiricism）和精實思維（lean thinking）之上。經驗主義（empiricism）堅信，知識來自經驗，以及根據觀察到的事物做出決策。精實思維（lean thinking）減少浪費，專注於根本。

Scrum 採行迭代（iterative）和增量（incremental）方法來優化對未來的預測性並控制風險。Scrum 讓一群共同擁有所有技能和專長的人員參與來完成工作，並根據需要來分享或獲取所需技能。

Scrum 有四個正式檢視與調適的事件，並且用另外一個事件：Sprint 來包含這四個事件。這些事件起作用的原因是它們實現了以經驗為導向的 Scrum 支柱：透明性、檢視性與調適性。

### 透明性（Transparency）

湧現的（Emergent）流程和工作必須對執行工作和接受工作的人員都是可見的。在 Scrum 裡，重要的決策是基於對其三個正式 artifacts 的感知狀態。透明性低的 artifacts 會導致做出讓價值減少且風險增加的決策。

透明性促成檢視性。沒有透明性的檢視會產生誤導和浪費。

## 檢視性（Inspection）

必須要經常用心地檢視 Scrum 的 artifacts 與一致同意的目標進度，以便發現潛在的誤差和問題。為了有助於檢視性，Scrum 以五個事件的形式提供了穩定的節奏。

檢視性促成調適性。沒有調適性的檢視是沒有意義的。Scrum 的事件旨在激發改變。

## 調適性（Adaptation）

若是流程的任何方面已偏離且超出可接受的範圍，或是所得的產品無法被接受，則必須將所採用的流程或其打造出來的材料進行調整。調整工作必須盡快執行，以減少未來更多的偏差。

當相關人員未獲得授權，或無法自我管理（self-managing）時，調適性將變得更加艱難。一個 Scrum Team 透過檢視而學習到的新事物之當下，應當立刻調適。

## Scrum 的價值觀

Scrum 的成功應用取決於人們更完善地活出這五項價值觀：

***承諾、專注、開放、尊重、與勇氣***

Scrum Team 致力於達成其目標並彼此相互支援。他們主要專注在 Sprint 的工作，盡可能地朝向目標前進，以獲取最好的進展。Scrum Team 與其利害關係人（stakeholders）對於工作及挑戰抱持著開放的態度。Scrum Team 的成員們互相尊重對方是個有能力和獨立的人，同時也受其他同事同等的尊重。Scrum Team 的成員們有勇氣去做對的事情、並處理棘手的問題。

這些價值觀指引了 Scrum Team 的工作、行動與行為。所做出的決定、所採取的步驟與如何使用 Scrum 都應該反覆的加強這些價值觀，而不是削弱或破壞這些價值觀。Scrum Team 成員們依照 Scrum 的事件和 artifacts 來學習並探索這些價值觀。當 Scrum Team 以及與工作的同事們展現了這些價值觀，就能活現出 Scrum 經驗的支柱：透明性、檢視性與調適性，並在每個人之間建立信任。

## Scrum Team

Scrum 以小團隊作為基本單位，稱為 Scrum Team。Scrum Team 由一名 Scrum Master、一名 Product Owner 與 Developers 所組成。在 Scrum Team 內沒有子團隊或是階級架構，是由一群具有凝聚力的專業人士，一次只專注在一個目標上：Product Goal。

Scrum Teams 是跨職能（cross-functional）的，也就是說，成員們擁有在 Sprint 內創造價值所需的所有技能。他們也是自我管理（self-managing）的，也就是說，他們在團隊內部決定誰做什麼、何時做以及如何做。

Scrum Team 規模足夠小以保持靈活性，同時足夠大以便可以在 Sprint 內完成重要的工作，通常人數在 10 人以下。一般而言，我們發現越小的團隊溝通越好，生產力更高。如果 Scrum Teams 變得太大，他們應該考慮重新組織為多個有凝聚力的 Scrum Team，每個團隊還是專注在同一個產品，因此，他們應該有著相同的 Product Goal、Product Backlog 和 Product Owner。

Scrum Team 負責所有與產品相關的活動，像是與利害關係人（stakeholder）的協同合作、驗證、維護、營運、實驗、研究、和開發以及任何其他可能需要的工作。組織成立 Scrum Team 並授權他們自行管理工作。在 Sprint 中維持可持續的步調工作，可以加強 Scrum Team 的專注與一致性。

整個 Scrum Team 都有責任在每個 Sprint 中創造出有價值的、有用的 Increment。Scrum 在 Scrum Team 中定義了三種特定的當責（accountabilities）：Developers、Product Owner 和 Scrum Master。

## Developers

Scrum Team 當中的 Developers 是指在每個 Sprint 致力於打造任何可用 Increment 的成員。

Developers 所須具備的特定技能通常很廣泛，並且會隨著工作領域不同而有別。然而，Developers 總是對以下事項負有責任：

- 打造一份 Sprint 的計畫，也就是 Sprint Backlog；
- 藉由遵循完成之定義，以灌輸品質；
- 每天調適其邁向 Sprint Goal 的計畫；和，
- 作為專業人士對彼此負責。

## Product Owner

Product Owner 負責將把 Scrum Team 的工作所打造出來的產品價值最大化。如何做到這一點可能會依組織、Scrum Teams 和個人的不同而有極大的差異。

Product Owner 也負責對 Product Backlog 進行有效的管理，包括：

- 開發並明確的描述溝通 Product Goal；
- 創造並清楚的描述溝通 Product Backlog items；
- 對 Product Backlog items 進行排序；和，
- 確保 Product Backlog 是透明的、可見的與可理解的。

Product Owner 可以自己做上述工作，或者也可以將職責委託他人，然而，Product Owner 仍肩負最終責任。

為了讓 Product Owners 成功，整個組織必須尊重他們的決定，這些決定在 Product Backlog 內容和順序中可以被看見，並且在 Sprint Review 時透過可檢視的 Increment 展現出來。

Product Owner 是一個人，而不是一個委員會。在 Product Backlog 中，Product Owner 可能代表了許多利害關係人（stakeholders）的需求，想要改變 Product Backlog 的人可以試著去說服 Product Owner。

## Scrum Master

Scrum Master 負責按照 Scrum 指南來建立 Scrum，方式是透過幫助 Scrum Team 內與組織內部的每個人了解 Scrum 的理論與實作。

Scrum Master 對 Scrum Team 的效能負責。他們透過讓 Scrum Team 在 Scrum 框架內改善其實務作法來做到這一點。

Scrum Masters 是真正的領導者，服務對象是 Scrum Team 和更大範圍的組織。

Scrum Master 以多種方式服務 Scrum Team，其中包括：

- 以教練方式提升團隊成員的自我管理（self-management）與跨職能能力（cross-functionality）；
- 協助 Scrum Team 專注於打造出滿足完成之定義且具備高價值的 Increments；
- 促使 Scrum Team 的阻礙被移除掉；
- 確保所有的 Scrum 事件都舉行，有建設性、有成效的並且保持在時間盒（timebox）內進行。

Scrum Master 以多種方式服務 Product Owner，其中包括：

- 幫助找到有效定義 Product Goal 與管理 Product backlog 的技巧；
- 幫助 Scrum Team 理解為何需要清楚且簡明的 Product Backlog items；
- 幫助在錯綜複雜（complex）的環境下，建立以經驗為導向的產品計畫；和，
- 當被要求或需要時，引導利害關係人（stakeholder）的協同運作。

Scrum Master 以多種方式服務組織，其中包括：

- 在組織採用 Scrum 的過程中，以領導、訓練和教練方式帶領組織；
- 在組織內，規劃並指導 Scrum 的執行運作；



- 幫助員工和利害關係人（stakeholders）理解與制定以經驗為導向的方法來處理錯綜複雜（complex）的工作；
- 移除利害關係人（stakeholders）與 Scrum Teams 之間的障礙。

## Scrum 事件

Sprint 是所有其他事件的容器。Scrum 中的每個事件都是用來檢視與調適 Scrum artifacts 的正式機會。這些事件都是為實現所需要的透明性而特別設計的。未能按規定運作任何事件將導致失去檢視和調適的機會。Scrum 使用事件來創造規律性，並以此減少 Scrum 中未定義的會議的需要。最理想的是，所有事件都在同一時間同一地點舉行，以減少複雜性。

### The Sprint

Sprint 是 Scrum 的核心，在這裡能將創意轉化為價值。

Sprint 是固定時間長度的事件，為期一個月或更短，以保持一致性。前一個 Sprint 結束，下個新 Sprint 就緊接著開始。

所有為了達成 Product Goal 的工作都發生在各個 Sprints 內，包括 Sprint Planning、Daily Scrums、Sprint Review、和 Sprint Retrospective。

在 Sprint 期間：

- 不得做出危及 Sprint Goal 的改變；
- 不得降低品質；
- 需要時將 Product Backlog 精煉；和；
- 隨著學到更多，可以與 Product Owner 針對範疇作進一步澄清與重新協商。

藉由至少每個月一次對邁向 Product Goal 的進度來進行檢視與調適，Sprint 促成了可預測性。當 Sprint 的時間太長，可能導致 Sprint Goal 失效，複雜性的程度可能會上升，以及風險可能會增高。採用時間較短的 Sprint，可以建立更多學習周期，並將成本與氣力的風險控制在一個較短的時間內。每一個 Sprint 都可以視為一個短期專案。

用來預測進度的實務做法有很多種，像是燃盡圖、燃起圖，或是累積流量圖。雖然這些做法是有用處的，但不能取代經驗主義（empiricism）的重要性。在錯綜複雜（complex）的環境中，接下來會發生什麼是不可知的。只有已經發生的事物才能用來做前瞻性的決策基礎。

若 Sprint Goal 已經不合時宜，可以取消 Sprint。只有 Product Owner 有取消 Sprint 的權限。

## Sprint Planning

Sprint Planning 透過安排在 Sprint 中要執行的工作來啟動 Sprint。計劃是由整個 Scrum Team 協同合作所產出的。

Product Owner 確保與會者準備好討論最重要的 Product Backlog items，以及它們如何對應到 Product Goal。Scrum Team 也可以邀請其他人參與 Sprint Planning 以提供建議。

Sprint Planning 處理以下主題：

主題一：為什麼這次 Sprint 有價值？

Product Owner 提議產品如何在這次的 Sprint 中增加其價值和實用性。然後整個 Scrum Team 協同合作定義出 Sprint Goal，並與利害關係人（stakeholders）溝通為什麼這個 Sprint 是有價值的。Sprint Goal 必須在 Sprint Planning 結束前被確定下來。

主題二：這次 Sprint 能完成（Done）什麼？

透過與 Product Owner 的討論，Developers 從 Product Backlog 內選擇一些項目，並放入目前的 Sprint 中。Scrum Team 可以在這個過程中精煉這些項目，從而增加理解與信心。

選擇在 Sprint 中可以完成多少項目可能會有挑戰，但是，Developers 越知道他們以往的表現、他們接下來的產能、與他們對完成之定義了解得越多，他們對 Sprint 的預測就越有信心。

主題三：如何完成所挑選的工作？

對於每個選定的 Product Backlog item，Developers 會計畫必要的工作，以便創造符合完成之定義的 Increment。這個過程通常會把 Product Backlog items 拆解成等於或小於一天的較小工作。但要如何拆解是完全由 Developers 來決定。沒有人告訴他們要如何把 Product Backlog items 轉化成具有價值的 Increments。

所謂的 Sprint Backlog 是由 Sprint Goal、該 Sprint 所選的 Product Backlog items，以及如何交付的計畫所組成。

Sprint Planning 是有時間盒限定（timeboxed）的，一個月的 Sprint 最多為八個小時；而較短的 Sprint，這個事件所需時間通常會更短。

## Daily Scrum

Daily Scrum 的用途是檢視目前 Sprint Goal 的進度，並根據需要調適 Sprint Backlog，以調整即將到來的計畫工作。

Daily Scrum 是一個屬於 Scrum Team 當中的 Developers 的 15 分鐘事件。為了降低複雜性，它在 Sprint 內的每個工作天會在同樣時間、同樣地點舉行。如果 Product Owner 或 Scrum Master 積極地打造 Sprint Backlog 上的工作項目，他們會以 Developers 的身份參與。

Developers 可以選擇他們想要的任何 Daily Scrum 的結構和技術，只要他們的 Daily Scrum 專注於實現 Sprint Goal 的進展，並且產生下一個工作天可執行的計畫。這樣可以更專注並改進自我管理（self-management）。

Daily Scrum 增加溝通、點出障礙、促進快速決策，從而消除其他會議的需要。

Daily Scrum 並不是 Developers 唯一一次允許調整計劃的時間，在一整天的工作中，他們常常會見面詳細討論要如何做出調適，或是重新計畫 Sprint 剩下的工作。

## Sprint Review

Sprint Review 的用途是檢視此 Sprint 的成果和決定未來的調適方向。Scrum Team 向利害關係人（stakeholders）展示他們的工作結果，並討論 Product Goal 的進展情況。

在這個事件中，Scrum Team 與利害關係人（stakeholders）回顧在 Sprint 中完成的成果，以及環境發生了什麼變化，基於這些資訊，與會者對接下來要做什麼進行協同合作，也可調整 Product Backlog，藉此來掌握新的機會。Sprint Review 是一個工作會議，Scrum Team 應避免將其限於投影片展示。

Sprint Review 是每個 Sprint 最後倒數的第二個事件，是有時間盒限定（timeboxed）的：一個月的 Sprint 最多為四個小時；而較短的 Sprint，這個事件所需時間通常會更短。

## Sprint Retrospective

Sprint Retrospective 的用途是規劃出能提升品質與效能的方法。

Scrum Team 檢視上個 Sprint 中有關人員、互動、流程、工具以及他們的完成之定義的情況。被檢視的元素通常隨工作領域而不同。團隊會辨識出他們迷失方向的假設，並探究這些假設的起源。Scrum Team 討論此次 Sprint 中，什麼進展順利，遇到哪些問題，以及如何（或為何無法）解決這些問題。

Scrum Team 辨識出最有用的改變以提升其效能。最具衝擊力的改善行動將儘速執行。甚至可以納入到下一個 Sprint 的 Sprint Backlog 中。

Sprint Retrospective 是總結 Sprint 的事件。是有時間盒限定 (timeboxed) 的，以一個月的 Sprint 來說，最多為 3 個小時；而較短的 Sprint，這個事件所需時間通常會更短。

## Scrum Artifacts

Scrum 的 artifacts 所代表的是工作或價值。Artifacts 的設計是為了使關鍵資訊之透明性極大化。因此，每個檢視這些 artifacts 的人，對於調適，都會有相同的基礎。

每個 artifact 都包含一個承諾，以確保它提供可增強透明性和專注度，給以下可以進度量測的事物：

- 對於 Product Backlog 而言，它是 Product Goal；
- 對於 Sprint Backlog 而言，它是 Sprint Goal；
- 對於 Increment 而言，它是完成之定義。

這些承諾的存在是為了強化經驗主義 (empiricism) 和 Scrum Team 及其利害關係人 (stakeholders) 的 Scrum 價值觀。

## Product Backlog

Product Backlog 是一份湧現的 (Emergent) 和有順序的清單，它列出產品需要被改善的地方，它是 Scrum Team 工作事項之唯一來源。

凡是 Scrum Team 能夠在一個 Sprint 中完成 (Done) 的 Product Backlog items，就可視為是備妥的，而可在 Sprint Planning 中被挑選。通常在精煉活動後才可達到這樣的透明性。Product Backlog 的精煉是將 Product Backlog items 拆解並進一步定義，使其變得更小、更精確的活動。這是一項持續進行的活動，加入更多描述、順序與大小之類的細節。這些屬性通常隨工作領域而不同。

實際從事工作的 Developers 要負責其適當的大小。Product Owner 可以藉由幫助 Developers 理解和權衡取捨來影響他們。

### 承諾：Product Goal

Product Goal 描述了產品的未來狀態，可以作為 Scrum Team 制定計劃的目標。Product Goal 在 Product Backlog 中。Product Backlog 的其餘部分會湧現出來以定義「做哪些事情」可以實現 Product Goal。

*產品是交付價值的載具，它具有明確的邊界、已知的利害關係人 (stakeholders)、定義明確的使用者或客戶。產品可以是一種服務、一個實體的產品，或是更抽象的東西。*

Product Goal 是 Scrum Team 的長期目標，他們必須完成 (或放棄) 一個目標才能再開始下一個。

## Sprint Backlog

Sprint Backlog 是由 Sprint Goal（為什麼做）、該 Sprint 所選擇的 Product Backlog items（做些什麼）以及用來交付 Increment 的執行計畫（如何做到）。

Sprint Backlog 是 Developers 所製定並專屬的計畫。它是 Developers 在 Sprint 期間為實現 Sprint Goal 而規劃要完成的工作，是一個工作高度可視且即時的工作畫面。因此，隨著學到更多，Sprint Backlog 會在整個 Sprint 期間進行更新。它應該有足夠的細節，以便可以在 Daily Scrum 中檢視其進度。

### 承諾：Sprint Goal

Sprint Goal 是 Sprint 的單一目標，儘管 Sprint Goal 是由 Developers 所做出的承諾，但它為實現該目標所需的確切工作提供了彈性。Sprint Goal 還創造了連貫性和專注性，鼓勵 Scrum Team 一起工作而不是分開行事。

Sprint Goal 是在 Sprint Planning 事件中被創造出來，然後添加到 Sprint Backlog 裡。當 Developers 在 Sprint 期間工作時，他們將 Sprint Goal 銘記在心。如果需要做的工作跟他們原本預期的不一樣，他們會與 Product Owner 協同合作，在不影響 Sprint Goal 的情況下，來協商 Sprint Backlog 的範圍。

## Increment

一個 Increment 是邁向 Product Goal 的一塊堅實踏腳石。每個 Increment 都是之前所有的 Increments 累加起來的，並經過了徹底地驗證，以確保所有整合在一起的 Increment 都是可用的。為了提供價值，Increment 必須是可用的。

一個 Sprint 內可以產生多個 Increments。在 Sprint Review 時，會把所有 Increments 展示出來，從而支持經驗主義（empiricism）。但是，Increment 可以在 Sprint 結束之前交付給利害關係人（stakeholders）。Sprint Review 絕對不應該被視為發布價值的關卡。

一項工作除非符合完成之定義，否則不能將其視為 Increment 的一部分。

### 承諾：完成之定義

完成之定義是當 Increment 符合產品所需的品質測量的正式描述。

當一個 Product Backlog item 符合完成之定義時，就會誕生一個 Increment。

完成之定義是藉由提供每個人對 **Increment** 中已完成工作的共同認知，而建立透明性。如果一個 **Product Backlog item** 不符合完成之定義，那麼它就不能發布，甚至不能在 **Sprint Review** 中展示。反之，它會回到 **Product Backlog** 中以供將來考慮。

如果 **Increment** 的完成之定義是組織標準的一部分，那麼所有 **Scrum Teams** 都必須以此為最低標準來遵守。如果它不是組織標準，那麼 **Scrum Team** 必須制定適合該產品的完成之定義。

**Developers** 需要遵循完成之定義。如果有多個 **Scrum Teams** 一起開發同一個產品，他們必須一起制定並遵守同樣的完成之定義。

## 結語

Scrum 是免費的，並在本指南中提供。本文所描述的 Scrum 框架是不可改變的。雖然實施部分的 Scrum 是可能的，但結果就不是 Scrum 了。Scrum 僅能以完整的形式存在，唯其如此也才能有效的成為其他技術、方法論和實務做法的容器。

## 致謝

### 人員

在為 Scrum 作出貢獻的成千上萬的眾人當中，我們要特別指出那些在最初提供幫助的人們：Jeff Sutherland 以及與他一起工作的 Jeff McKenna 和 John Scumniotales，還有 Ken Schwaber 以及與他一起工作的 Mike Smith 和 Chris Martin，還有他們一起工作。在隨後的幾年中，更多的其它人作出了貢獻，如果沒有他們的幫助，Scrum 將不會如同今天這般精煉。

### Scrum指南的歷史

Ken Schwaber 和 Jeff Sutherland 在 1995 年 OOPSLA 的大會上首次聯合發表 Scrum。這場演講本質上記錄了 Ken 和 Jeff 在過去幾年的學習成果，也首次公開了 Scrum 的正式定義。

Scrum 指南記錄了 Jeff Sutherland 和 Ken Schwaber 在三十多年間對 Scrum 的開發，演進，和維護。其他的資源在模式、流程和見解方面為 Scrum 框架提供了補充。這些可能可以提高生產力、價值、創造力和對結果的滿意度。

Scrum 的完整歷史在其他地方有所記載。我們對於首先嘗試並證實其有效的公司表達敬意：Individual, Inc., Newspaper, Fidelity Investments, 和 IDX (現為 GE Medical)。

### 致謝繁體中文譯者

本繁體 2020 中文指南（版本 1.1）是從 Ken Schwaber 和 Jeff Sutherland 所提供的英文原版（2020 版）翻譯而來。

中文翻譯團隊（[scrum-guide-chinese-translators@googlegroups.com](mailto:scrum-guide-chinese-translators@googlegroups.com)）包括：

葛仲安（Chungan Ke）[linkedin.com/in/chungan-ke](https://www.linkedin.com/in/chungan-ke) 李奇霖（Tony Lee）[linkedin.com/in/chilinlee](https://www.linkedin.com/in/chilinlee)

蘇於登（Ethan Soo）[ethansoo@hotmail.com](mailto:ethansoo@hotmail.com) 王泰瑞（Terry Wang）[agileTerryWang@gmail.com](mailto:agileTerryWang@gmail.com)

王晶（Philip Wang）[philip.wang@odd-e.com](mailto:philip.wang@odd-e.com) 周建成（Zhou Jian Cheng）[zhoujiancheng@gmail.com](mailto:zhoujiancheng@gmail.com)

林偉弘（Andrew Lin）[linkedin.com/in/andrewlinagile/](https://www.linkedin.com/in/andrewlinagile/)

同時，我們對以往中文版的譯者表示感謝：

2017: 繁體：張裕宇 (Finn YuYu Chang)、王泰瑞 (Terry Wang)、林偉弘 (Andrew Lin) 簡體：周建成

2013: 繁體：林偉弘 (Andrew Lin) 簡體：李麟德 (Derek Li)、王军 (Jim Wang)

2011: 簡體：鮑央舟、孙媛

## 2020 年 Scrum 指南版本更新說明

### 甚至更少的規定指示

這些年來，Scrum 指南開始變得有一些規定指示。2020 年版本旨在透過刪除或淡化規定性用語，使 Scrum 恢復到原本最低限度的框架。例如刪除了 Daily Scrum 的三個提問，淡化了關於 PBI 屬性的用語，淡化了關於 Sprint Backlog 中改進項目的用語，減化了取消 Sprint 的段落，等等其它更多部分。

### 一個團隊，專注於一個產品

目的在消除團隊中有其它團隊的概念，導致 PO 和 Dev 團隊之間出現「代理人」或「我們跟他們」的行為。現在只有一個 Scrum 團隊專注於同一目標，有三個不同當責（accountabilities）：PO，SM 和 Developers。

### 加入 Product Goal

2020 年 Scrum 指南引入了 Product Goal 的概念，為 Scrum Team 提供了一個更具價值的專注重點。每個 Sprint 都應使開發中的產品更接近整體的 Product Goal。

### Sprint Goal、完成之定義和 Product Goal 有了歸宿

先前版本的 Scrum 指南描述了 Sprint Goal 和完成之定義，但是沒有真正賦予它們明確的身份。它們不全然是 artifacts，反而是類似依附在 artifacts。隨著增加了 Product Goal，2020 版本對此提供了更清晰的說明。現在，三個 artifacts 中都包含一個相對應的「承諾」。對於 Product Backlog，它相對應的是 Product Goal，Sprint Backlog 則有 Sprint Goal，而 Increment 有完成之定義（Definition of Done；現在，「完成（Done）」英文不再加引號）。這些承諾的存在是為了帶來透明性，並專注於每個 artifact 的進度。

### 自我管理（Self-Managing）超越自我組織（Self-Organizing）

先前版本的 Scrum 指南將 Development Team 稱為自我組織的（self-organizing），他們能選擇人員和如何做事。2020 版本更加著重於 Scrum Team，強調了一個自我管理（self-managing）的 Scrum Team，他們能選擇人員，如何做事，和做什麼事。

### 三個 Sprint Planning 的主題

Sprint Planning 主題除了「什麼」和「如何」之外，2020 年 Scrum 指南還強調了第三個主題「為什麼」，這裏的「為什麼」指的是 Sprint Goal。

### 擴大讀者範圍，全面簡化用語

2020 年 Scrum 指南著重於移除冗長和錯綜複雜（complex）的陳述，同時也刪除所有與 IT 工作相關的推斷（例如，測試，系統，設計，需求等）。現在的 Scrum 指南不到 13 頁。