

敏捷教練

你已經學習有關 Scrum、XP、精實以及看板。你知道他們有何共通點，並了解他們能達到些什麼。若你與其他人一同建構軟體，那麼你至少已經看到了一些事——一些可以幫助你們團隊的實踐、觀念和態度的改變。

現在，就去試試吧。讓你們團隊敏捷起來，就是現在！

現在看起來還不是那麼實際，對吧？在書上學習有關價值、原則、心態和實踐，並實際去改變一個團隊的工作方式還是不太一樣的。

有些團隊有辦法從 Scrum 或 XP 的書上採用實踐，並立即見到很棒的成效。在讀了本書的前 9 章之後，你應該可以發現為什麼：這些團隊的心態早已與敏捷宣言及方法論的價值及原則不相衝突了。對一個這樣能相容的團隊，採用敏捷會感覺很簡單，因為團隊的個人不須改變他們對他們工作的想法。所以若你已經有與你試著要採用的敏捷方法論相容的心態，你就更可能採用成功。

但若你並沒有一個能與 Scrum、XP 或另一個敏捷方法論相容的心態呢？又或者你身處在一個敏捷價值難以成功的環境中呢？再者若在你們團隊裡個人貢獻的回報遠比團隊努力多呢？還有若錯誤會被嚴厲懲罰呢？而若你身處在一個會扼殺創新，或你們團隊沒有辦法接觸到能協助你了解你所建構的軟體的客戶，用戶或其他人的一個環境呢？這些都是敏捷採用時的障礙。

這時候就需要一位敏捷教練（Agile Coach）。敏捷教練就是一位幫助團隊採用敏捷的人。

他或她會幫助團隊中的每個人學習一個新的態度及心態，並克服會妨礙團隊採用一個敏捷方法論的心理上、情緒上和技術上的障礙。敏捷教練不僅會協助團隊上的每個人，幫助其了解他們被要求導入的新實踐是「如何」進行，也會協助其了解「為何」要這樣做。教練會幫助團隊克服發生在每個被要求在工作中嘗試新事物的人身上那些對改變感到正常的不喜歡——甚至是害怕。

我們已經在本書中看到有人的結果是「只比沒做好一點」的例子：一個團隊採用了一個敏捷方法論，但他們的成員們只得到了微小的進步，因為他們並沒有真的改變他們對工作的想法，或他們對一同建構軟體的態度。換句話說，你們團隊需要一個敏捷心態來採用敏捷方法論。敏捷有著敏捷宣言的價值及原則，可以幫助團隊調整成正確的心態，而每一個方法論都因為同樣的原因而有其各自的價值與原則。當一個團隊的每個人都有相容於其所採用的特定方法論與敏捷價值及原則的心態時，團隊就能取得最棒的結果。

敏捷教練的目標是幫助團隊調整成更好更敏捷的心態。一個好教練會幫助團隊選擇一個最適合他們心態的方法論，也會用一個他們最能接受的方式來向他們介紹這個方法論的價值、原則與實踐。教練會幫助團隊採用這些實踐，然後使用這些實踐來幫助團隊學習並內化這些價值與原則，慢慢地改變他們的態度，最後調整成正確的心態以避免落入「只比沒做好一點」的結果。

在本章中，你將會學到關於敏捷教練秘技：教導團隊如何學習，一個敏捷教練如何幫助團隊調整心態來更容易採用一個敏捷方法論，以及教練如何可以幫助你們團隊變得更敏捷。



劇情：一個在被大型網路集團收購的公司開發手機照相 App 的團隊

Catherine — 開發人員

Timothy — 另一位開發人員

Dan — 他們的老闆

第 3 幕：只剩一件事（又來了？！）……

「嘿，Cathy！妳有空嗎？我有很棒的消息唷。」

當 Catherine 的老闆 Dan 叫她進他辦公室時，她小聲地嘟囔了一下。她心想，我永遠沒辦法習慣這種事。當她坐下來時候，她已準備好面對所謂的「很棒的消息」。

「這整個敏捷的事妳做的很棒。真的，棒呆了。」

Catherine 回想起過去她和 Timothy 花了超過 8 個月時間使用看板方法論來改善他們的流程。她已經確認他們流程中的數個瓶頸，特別是 Dan 與其他經理們看起來會對他們隨機拋出的新功能這個大瓶頸。當她加入一個佇列之後神奇的事就發生了：當他們開始為了爭搶佇列上的順位時，他們慢慢地停止要求她與 Tim 去做超出他們能力範圍所能負擔的工作。

實際上，想到這裡，這還是這幾個月來 Dan 第一次把她叫進辦公室來跟她說「很棒的消息」——通常這個指的是他有一個會讓她的專案完全偏離的最後一刻重大改變。

「我很樂意聽到你很高興，Dan，」Catherine 用半帶諷刺的意味說道。

Dan 繼續講，「我們母公司的人注意到了。他們已經在了解這些敏捷的事當中，既然我們已經取得一些成功，他們要我們教其他團隊如何做。」

Catherine 說道，「等等。所以你是要我去……幹嘛？指導他們？」

「正是如此，」Dan 說道。

「我不知道怎麼當個教練啊，」Catherine 回說。

Dan 直直地看著她的眼睛。「這由不得妳，Cathy。來吧，振作點去幹活吧。」

教練了解為何人們總是不想改變

在你的認知中，多數人是想試著把工作做好的。他們想要他們的同儕和主管看到他們在交付給他們的工作上是有好的表現。當某人已經對其的工作發展出一定程度的舒適與熟悉感，他最不想做的事就是有人跑過來讓他採用一個全新的做事方法。

—— Andrew Stellman 和 Jennifer Greene 所著的
《Applied Software Project Management》

敏捷教練多數時間是在幫助團隊中的人們改變工作的方式。這對於教練和團隊來說都是個挑戰，因為只有教練看到全貌。對團隊中的人們，他們被要求採用如何做他們工作的新方法，但他們不見得知道為何要這麼做。

許多時候，試著採用某個實踐的團隊會出於好意去改變這個實踐，讓它失去原本的作用。例如，我們在第5章中看到團隊經常把每日 Scrum 會議（Daily Scrum）變成每日進度會議。每日 Scrum 會議的一個重要目的是用自我組織來取代一個口令一個動作式的專案管理；在會議上每個人問的3個問題是要讓團隊能自我掌控他們自己的專案。但許多試著採用 Scrum 的團隊最後會變成把它當成每日進度會議來開，讓團隊成員們報告個別進度，Scrum 主持人看起來就像是會指派工作給大家的實質上的專案經理。

同樣地，一些團隊把故事加到商業需求書中，但卻像是多數瀑布式團隊一樣地把這些文件當成規格來看待。他們仍是在做預先最大需求（Big Requirements Up Front, BRUF）開發，是把用戶故事加了進來。或一些寧願嘗試採用 XP 的團隊只會確保他們對建構程式碼後所開發的測試有很高的程式碼覆蓋率，而不願做測試驅動開發——這表示測試對設計沒有任何影響，因為他們是在測試被寫下時才完成的。

在這些情況中，這些團隊中的人們是真的試著去採用敏捷。但在這些例子中，團隊中的人們並沒有真的了解這些實踐是如何融入一個更大的方法論生態體系中。所以他們寧願專注在他們覺得熟悉的實踐部分而不願試著改變他們工作的方式。而且我們為何要期待任何改變呢？一個只知道一個口令一個動作式的專案開發的團隊並沒有自我組織的經驗，也沒有其他背景可以幫助他們更了解每日 Scrum 會議。

讓我們客觀地來看。多數要採用敏捷的團隊已經在建構軟體並取得一些成果。（完全失敗的團隊很少會有那麼開明的經理會在一開始就來讓他們嘗試敏捷。）他們習慣進行小的漸進式調整，因為他們不想中斷目前的進度。

這就告訴我們採用敏捷的最大絆腳石是什麼了：當一個團隊成員這樣想「我已經看過敏捷了，我已經採用了我所熟悉的實踐，我們團隊運作得比之前好，這樣就夠了」。這就是我們所說的「只比沒做好一點」結果，而這就是人們透過把敏捷限縮到很特定、很小的改善，讓原本令人期待的改善變得令人失望，也使得敏捷就這樣沒了。

為何團隊成員們經常堅持只採用看起來是他們所熟悉的實踐，並拒絕那些沒有辦法立即顯現他們當天所做的事情的成效的其他實踐呢？

因為每個新的實踐都是個改變，而改變總是有失敗的時候。當辦公室裡進行的改變失敗時，有人就得回家吃自己。

這是每個敏捷教練需要事先記在腦海中的。引導的工作是幫助團隊改變，而改變可能會引起被要求改變的人們一個很大的——但卻完全可以理解的——情緒性反應。為什麼？因為工作不就是为了混口飯吃。

當我們在工作上被要求學習與做新的事情，若我們覺得我們不能很快與簡單地精通它，它會引起我們嚴重的情緒不適。我們的腦袋裡面有一個基本的狩獵採集（Hunter-Gatherer）部分是這樣想的：「昨天我知道我可以做我的工作並把食物帶回家給我的家人，但今天我沒辦法這麼確定了。」這是一個重要的原因，為何被期待去學習工作上的新事物會引起焦慮與看似不理性的反應（在這裡看來也不是那麼完全無法理解就是了）。

另一個關於為何人們拒絕工作上的改變，並被他們已知的所吸引的原因，是他們不覺得自己有時間可以坐下來思考這背後的所有原因。舉例而言，讓團隊試著在一個試驗專案中採用敏捷是很常見的。一開始通常是有個人讀了關於敏捷的書，現在在領導整個團隊進行採用——同時間還得處理一般專案都會有的截止日期、Bug、衝突、需求變更以及其他事項。對一個工作的全新想法而言，這不是個適合的試驗場。人們會盡力做，但若他們有些事還不太了解，他們會維持「敏捷」的標籤與這些方法論和實踐的名稱，但工作上不會有太大的改變。他們舊的專案中的里程碑現在被標示為「衝刺」或可能有人放了一個從來沒影響過被完成的工作的白板。最終，團隊就是做著之前所做的事，因為他們知道過去在這行得通，而截止日期還在等著他們。

當一個團隊使用敏捷實踐之名，但卻沒有改變他們工作的方式，我們不難看到為何團隊成員們很快地就對敏捷背後的理念失去興趣了。對他們而言，敏捷不過就是他們正在做的事情的另一種名稱罷了。他們認為自己已經採用了敏捷，但他們還沒有改變自己工作的方式，所以他們得到了一樣的結果。

團隊中的人們會認為敏捷沒用，就算他們從未明白他們不曾真實生活中看過類似敏捷的東西。他們會有這種反應是因為他們在不了解的狀況下被要求改變工作的方式，也沒人在承諾新實踐與想法的完整性的狀況下來幫助他們。

這就是為什麼團隊需要敏捷教練。敏捷教練有個重要工作，就是找出人們何時被要求做新事物，並幫助他們對改變感到放心。教練需要讓人們知道改變的內容，讓他們了解為何（而不是只有什麼）。這可以幫助團隊真的改變工作的方式，而不是只是拿了敏捷方法論中的一些名稱並把它們貼到現在正在做的工作上。

舉例而言，一位好的教練會說明——用大家都聽得懂的语言——每日 Scrum 會議如何幫助團隊自我組織、測試驅動開發如何幫助團隊進行功能性的思考並走向漸進式設計，或者用戶故事如何幫助團隊的每個人了解軟體用戶的觀點。教練不只是幫助每個人採用新規則，所以大家開始看到他們的新目標，還有這些新事物最終會帶給他們些什麼。

教練要傾聽團隊因應改變時面有難色的徵兆

若你花時間指導許多團隊，你會從許多不同人那裡聽到相同的情緒。這裡有一些團隊成員說的話會表示他們對於改變感到不適，而當你遇到時你可以做的是——從指導的觀點來看，這對觀察事物是有幫助的，即便你不是一位敏捷教練：

「我們已經很會建構軟體了。為何你要告訴我們不一樣的作法？」

要和成功爭辯不容易。若你在和一個能辦好事情的軟體團隊談話，他們是有權知道為何要做出改變的——光告訴他們這是老闆交代的並不够，因為這只是打擊他們的士氣罷了。身為一個教練，你需要對團隊過去所做的工作持正向態度。但對他們所遇到的問題不要羞於指出。每個敏捷方法論中的實踐都是要解決團隊遭遇的問題；若你可以幫助你們團隊了解為何問題會發生，並提供他們解決方案，他們就更可能會接納改變。

「這太冒險了。」

這是個針對敏捷很常見的反應，特別是對熟悉一個口令一個動作式的專案管理的人。緩衝在哪？風險管控表在哪？會讓專案慢下來與給我許多 CYA 的官僚流程又在哪？一個專案計劃可能是不透明的，這會讓團隊放心：他們只需要分享模糊的里程碑，也可以準備緩衝來減少甚至消除變更。當你用進度報告來作為軟體進度的衡量方式時，你可以控制你要傳達給用戶和客戶的訊息。

對於熟悉這些事務的團隊，敏捷可能讓人覺得有很大風險。在第 3 章，我們學到「可用的軟體是專案進度的主要衡量」這個敏捷原則。如果敏捷團隊中有什麼不對勁的地方，每一個人都會知道。作為敏捷教練，你的工作是幫助經理、用戶和客戶感受到純粹的進度——以及給予團隊一個允許他們今天失敗，只要他們明天能從失敗中學到功課的安全環境。如果現在這仍不切實際，那麼你的工作就是幫助每個人（尤其是老闆），設定一個改變將來團隊氣氛和態度的目標。

「對我而言，雙人程式設計（或測試驅動開發，或其他實踐）是行不通的。」

一個習慣獨自工作的人可能從骨子裡覺得雙人程式設計會拖慢自己的進度。他甚至可能有一個合理的觀點——如果他為一個有著「錯誤是不可接受的」心態的團隊（或老闆）工作，那麼有人看著他寫程式會讓他覺得不舒服也是合理的。有教過一個十幾歲少年開車的人，就能同樣感受到一個資深團隊成員想到要讓一個新手成員控制他們搭檔的鍵盤時的不安感受。人們有很多合理理由對這些實踐感到不適，尤其是當團隊有著與這些實踐不相匹配的心態。一位好的敏捷教練首先會幫助該團隊選擇相容他們心態的實踐。當團隊成員開始操作實踐，他們就會看到實踐帶來的好處並理解這些實踐為何以及如何運作。在一位優秀教練的指導下，每一個人會自然而然地開始朝著更敏捷的心態轉變。

「對我們這樣的公司而言，敏捷是行不通的。」

人們通常會這樣說，因為他們習慣於在投入工作之前先制定大型和詳細的計劃，他們無法想像用其他的方式工作。一個口令一個動作式的專案經理和老闆從展開工作前就制定好了範圍、需求，以及紙上專案計劃那裡得到的安慰。同樣地，在寫第一行程式前就有整個系統的設計稿紙會讓架構師和團隊主管感到放心。現在他們被告知要相信團隊會在最後責任時刻做決策——這意味著放棄了控制權。因此，他們會說類似於「我們公司是非常複雜的。其他公司的人或許可以直接進入專案。但由於我們業務運作的特殊性質，我們需要提前完成所有規劃。」事實上，每一間公司都是複雜的，每一個專案都需要分析和規劃。一位好的敏捷教練會幫助經理、業務人員，以及團隊主管看到，一個團隊在被允許把一項專案分成更小部分之後，會更善於捕捉專案的複雜性，並有自由在最後責任時刻做出決策。

「這就是我們已經在做的，只是有了個不同的名稱罷了。」

這是人們拒絕敏捷方法的最常見的原因之一。他們會想辦法繼續做他們正在做的事，只是在為他們已經讀過或聽過的敏捷實踐選擇相符的新名稱。如果他們把失敗的實踐命名為敏捷，這會讓所有敏捷顯得微不足道——甚至是錯的。事實上，如果你在網路上搜尋類似於「敏捷太爛了」的關鍵字，你就會找到真的這麼做的人，他們譴責敏捷是沒有實質內容的「炒作」，只是以往瀑布式實踐的另一個名稱。他們甚至認為敏捷使簡單、常識性的軟體開發變「模糊」了。作為一位教練，你需要認識到他們這麼做並不是心懷惡意或有不良企圖，只不過是有些沒有真正用過（卻相信自己已經用過）敏捷的人的自然反應。作為一位教練，你的工作就是幫助你們團隊了解敏捷與他們已經在做的不同，而且敏捷並不爛。

教練了解人們如何學習

精通任何東西（如果是可能的話）的一個好模式來自於武術。一個武術學生要達到精通水準的過程有守（Shu）、破（Ha）、離（Ri）的3個階段。守：謹守規則。破：破壞規則。離：成為規則。

——Lyssa Adkins 所著的

《Coaching Agile Teams: A Companion for ScrumMasters》

《Extreme Programming Explained》的作者 Kent Beck 採用類似的階段來描述極限程式設計（XP）。當他被問到 XP 和軟體工程學院的「能力成熟度模型」（Capability Maturity Model）的5個階段，他用 XP 的3個成熟度階段來回答：

1. 按照所寫的做每一件事。
2. 這麼做以後，嘗試在規則上做變化。
3. 最後，不再在乎你是否還在做 XP。

——Alistair Cockburn 所著的

《Agile Software Development: The Cooperative Game》第二版

再回到第2章，你學到不同的人如何用不同的眼光看待敏捷。就像每一個遇到大象的瞎子，不同的人一開始對「敏捷是什麼？」這個問題會有各自的答案。

程式設計師可能看到 XP 的程式設計實踐諸如測試驅動開發和漸進式設計，然後認定敏捷就是關於軟體程式、設計和架構的理論。另一方面，一個專案經理在讀完 Scrum 的衝刺、規劃、待處理項目（Backlog）和回顧（Retrospectives），則可能認定敏捷是與改善專案管理有關的。

我們花了數百頁的篇幅來探索 Scrum、XP、精實和看板。我們討論過心態、價值、原則和實踐。我們已經向你展示了如何把它們歸納一起，以幫助你們團隊為用戶和客戶帶來更多價值。我們已經幫你找到你今天為了幫助你們團隊所能做的，並且也為你提供了改變團隊心態的工具（例如「你覺得這個好嗎？」），我們已給予你許多學習敏捷以及了解 Scrum、XP、精實和看板的工具。

這並不是大多數人採用敏捷的方式。

儘管我們會很想讓世界上的每位開發人員都來購買和閱讀我們的書，事實上，大多數團隊的大多數成員並不是透過閱讀來學習敏捷的。他們是透過運用來學習的。把要做的事情分解成小的、簡單的區塊，使事情做起來更簡單。

換句話說，在大多數人嘗試學習敏捷時，他們真正想要的是一個讓他們能更快速建構更好軟體的簡單規則清單。

如果你是新的敏捷教練，這可能會很讓你有挫敗感。想像你在試著教導一個團隊如何有效地採用 Scrum。你從第5章知道，如果一個人不了解自我組織（Self-Organization）、共同承諾（Collective Commitment），他們是無法「掌握」Scrum 的。因此你花時間來解釋公開、勇氣、專注、承諾以及尊重等 Scrum 的價值。並且也談到自我組織如何運作，但是團隊卻感到受挫——這些似乎都是很抽象的概念。他們只想談論白板和故事。你想要他們「掌握」Scrum；他們似乎決心要得到「只比沒做好一點」的結果。你似乎無計可施。這是怎麼回事？

關於教學，有句老話是這樣說的：在他們所在的地方，而不是在你希望他們在的地方與他們會合。一位好的敏捷教練了解人們的學習方式，並且為他們提供資訊、指導和範例，以幫助他們達到學習的下一個階段。教練明白人們不會一夜之間改變。

在《Agile Software Development: The Cooperative Game》書中，Alistair Cockburn 談到一個名為守破離（Shuhari）的基本概念，它對於賞識找出團隊所在階段的教練來說，是一個很有價值的工具。守破離這個概念取自武術，但它是一個了解人們如何學習幾乎所有東西的好方法。

在守破離（Shuhari）概念裡，學習分三個階段。第一個階段守（「服從」或「遵守」），描述了一個剛接觸到新觀念或方法論的人的心態。她想要有簡單的規則可以遵守。這是為什麼人們緊抓著實踐的一個原因：該團隊可以制定關於採用衝刺、雙人程式設計、使用白板等的規則，然後每個人可以看出這些規則有沒有被遵守。

對於在工作中學習新事物的成人來講，規則至關重要的，因為成人不像上學的小孩子一樣習慣學習新事物。當你給一個正在學習新系統——例如敏捷方法論——的人一個簡單的規則去遵守，你就給了她一個起點。Scrum、XP 和看板裡的部分簡易實踐就很適用於在守這個階段的人，因為這些足夠簡易的實踐可以讓人們專注於做得更好。

藉由設定如現今團隊中的準則（就像我們在每天早上 10:30 召開一個每個人都可在會議中回答 3 個問題的 Scrum 會議），便可以幫助他們正確地學習（我們利用自我組織不斷地檢討及改變我們的計劃）。

然而，你的訓練目標不是建立一個靈活的教義。敏捷狂熱者是一個下定決心在每個團隊中只有絕對敏捷的人。除此之外，他還會大力向所有人推動此種方法。敏捷狂熱者往往只專注於自己所學到的規則，並停在守的階段。他們認為自己所遇到的每一個問題都可以被他們所知道的準則所解決。敏捷狂熱者並不會努力訓練，因為他們不會浪費時間在正接受學習的人們身上。

一個敏捷狂熱者需要了解的是帶領我們進入 Shuhari 的下一個階段破（「分離」或「破裂」）當中，而這已超越一般簡單的規則。在這個階段中，人們藉由準則來做實踐，並且可以開始理解和內化這些足以驅動他們的準則。

在本章中，你已經學會了要改變團隊心態的有效方法就是採用方法論的實踐，並透過實踐（和大量的團隊討論）來了解其價值與原則。當成員一起達到破階段，他們就會開始從只比沒做好一點的結果中進步，並且從實際採用敏捷的心態中獲得生產力的實際收益。

我們還沒有談論太多有關於最後離（「單獨」）的階段。當你到達離的學習階段，你在心態、價值和原則當中將游刃有餘。因為超脫，所以你將不在乎方法論。當你的問題可以被具體實踐，那麼你和團隊只需繼續此種作法。你一點也不在乎流程的名稱，不論它是 Scrum、極限編程、看板或瀑布。但它一點也不像是在無人嘗試過的敏捷團隊中所產生的混亂。

若每個人都達到離的階段，則團隊將運作良好，每個人只需做對的事。Cockburn 指出這個聽起來究竟有多像「令人困惑的禪」。在《Agile Software Development: The Cooperative Game》書中，他談到在離這個學習階段的人對於守階段的學習者所認為困難的事說道：

「做應該做的。」

「當你真的執行，你將不自覺自己正在做。」

「技術可使事情做得更好。」

對於一些已在行為順暢階段的人而言或許是真理，但對於還在分離階段的人卻只會造成困惑。而對於正在尋找依循流程的人而言則一點用處都沒有。

一個優秀的敏捷教練首要工作是與成員對談並歸納出每個人目前的學習階段。對於教練而言，如果成員處於離的學習階段，代表他們非常容易共事，因為每個成員知道如何學習並樂於實踐。

反之，在守階段的成員則不想要模糊的準則。例如，你是否會在白板貼上便利貼或需求卡片？對於教練而言或許只是小事，但對於一個從未使用過白板的人而言，他將無法得知是否為瑣碎或重要的決定，因此你的工作便是提供他一個準則。接著你可以幫助他理解為什麼你會這麼選擇或這麼說。例如使用便利貼及需求卡片一樣可以讓工作做得好。兩者是有點不同，舉例而言，你可以在一個需求卡片的兩面書寫，但是便利貼只有一面，並且可以使用異同處來幫助成員學習如何履行相同的原則。

使用 Shuhari 幫助團隊了解方法論的價值

讓你自己像敏捷教練般思考，試著教導團隊「導入」Scrum。Shuhari 可以幫助你理解為什麼團隊在學習價值時感到挫折，為什麼他們只想要談論類似白板和故事等比較具體的訓練。白板或故事並不難呈現出守階段的概念：你今天就能夠採用的簡單準則。Scrum 便充滿了守階段的準則（如同「每天見面並回答這 3 個問題」）；這也是團隊在眾多採用其作法成功案例中的一個。

另一方面，公開和承諾等價值則是破階段的想法：抽象的概念掌管你如何在系統中使用準則。自我組織和共同承諾，只會出現在一個所有成員都已經真正理解並內化這些價值的團隊中。Shuhari 幫助我們理解為什麼團隊需要先經過實踐的動作，才可以「分離」並開始了解公開和承諾的真正含義。

這就是為什麼 Scrum 的實踐在對於幫助團隊了解自我價值而言，是非常有效的第一步。一個好的教練是有耐心地在每一次教授團隊價值中等待機會。通常這些機會都會在不同觀點的兩個人看待 Scrum 的不同面向時出現。

例如，一個開發人員在全速衝刺的中途發現一個無法完成的功能，並在接下來的每日 Scrum 會議中，要求產品負責人把它推展到下一個衝刺點中。然而，產品負責人會感到生氣，因為她害怕客戶或經理也會對此感到生氣，並希望開發人員能「不惜一切代價」來完成它，即使這意味著將有可能偷工減料或無法完整地完程式。

對此問題，守階段回答如下：Scrum 的準則決定了團隊只能在衝刺審查中展示那些「做完的已完成」功能，而不完整的工作則必須被推到下次的衝刺。

敏捷教練了解 Scrum 的準則，並用此解決衝突。但一個好的敏捷教練也看到，這裡有一個破階段教學的機會。從一個不完整的角度出發也可以變得完整（如同我們在第 2 章所學習到的）。一個教練可以幫助產品負責人看到由開發人員角度所看到的東西，而這樣乏味的工作會增加技術上的負債（簡稱技術債），也將令整個基礎程式更加難以維護。相反地，這樣的技術債會導致未來的衝刺需要更長的時間。因此，儘管該產品負責人可

能因為忽視 Scrum 的準則而能使客戶盡快得到功能，但最終產品負責人的情勢會更糟，因為未來的發展會更慢。這將有助於產品負責人更詳細了解專注這項 Scrum 價值，以及一個團隊如何專注於讓完整的工作能更迅速地提供更高品質的產品。

教練更可以幫助開發人員了解功能的重要性、如何傳遞價值給客戶，以及如何在未來利用一個好的價值來幫助開發人員拉近與產品負責人的工作關係，並找到一種將功能普及到細微之處，並且更迅速提供價值給客戶的方法。這將幫助開發人員了解更多關於承諾這項 Scrum 價值，以及一個團隊可以更有效地理解什麼對於客戶和用戶而言是有價值的。

這是一個對方法論有深刻理解的教練如何將重要且具有價值的角色融入團隊的例子。教練幫助團隊了解並採用該方法論的實踐和準則。透過這些規則，教練可以幫助成員了解敏捷價值，並使用這些價值來幫助每個成員增進更敏捷的心態。

教練了解如何讓方法論產生作用

專案中的衝突和問題每天都在發生。例如一個良好的 Scrum 主持人會利用他大部分的時間來解決問題和困難。然而許多這樣的問題，並不一定會出現重大機會讓團隊能夠學習 Scrum。

所以在本例中，我們只是要讓你了解教練如何知道，在教授團隊專注和承諾等 Scrum 價值時，在這個特殊的衝突點中會存在著這個機會。而針對這個特殊的問題，教練是否有所認知？

教練知道這是一個機會，因為他明白不僅 Scrum 團隊如何不斷地運用迭代，以及為什麼需要運用。他知道如果一個團隊被允許將一個未做完的已完成工作納入，那麼有時間限制的迭代將會完全無效。如果沒有這條規則，迭代會變成簡單的專案里程碑：每次的迭代範圍變化愈來愈多，團隊將不再需要在每次迭代結束時交付可用的軟體。

這會傷害能讓敏捷運作的幾個重要的價值與原則。例如，教練知道可用的軟體是進度的主要衡量方式，而在衝刺尾聲交付不完整的軟體會給顧客和用戶一個錯誤的進度衡量。教練也知道敏捷團隊重視客戶合作——但他也知道客戶合作不代表客戶總是對的。這表示團隊發現計劃中有個實際的問題時，他們可以和客戶合作找到一個最有價值的解決方案。他發現若客戶與產品負責人可以壓迫開發人員在一個衝刺審查中去包括未完成的軟體，那麼團隊會對工作的真實進度變得更謹慎，而最終客戶合作會在團隊與客戶間對合約協商讓步。

教練知道這一切都是因為他了解敏捷與 Scrum 的價值與原則，並了解這些價值與原則如何驅動實踐。他知道是什麼讓方法「有用」。一位 Scrum 教練會知道共同承諾與自我組織。一位 XP 教練會知道擁抱改變與漸進式設計。一位看板方法教練會知道改善團隊的流程是如何地需要設定 WIP 上限並用來控制工作流程——而每當有人試著阻止這些 WIP 上限被設定時，整個方法就被破壞了。

在《Coaching Agile Teams》中，Lyssa Adkins 對教練可以如何處理團隊的問題有一個建議——我們在這本書稍早前有談到這個，但這值得再說一次：

讓團隊失敗：當然，別在團隊在懸崖邊搖搖欲墜時站著發呆。但記得利用每個衝刺中出現的許多機會來讓他們失敗看看。一起失敗與一起重新站起來的團隊，會比被保護的團隊更強更快。他們可能會讓你大吃一驚。你認為會傷害他們的東西，實際上卻可能是對他們有用的。等著瞧吧。

對任何敏捷教練這都是很棒的忠告，特別是有那種 Adkins 提到的有「一個口令一個動作」傾向的，或需要控制團隊學習與進度每個方面的那種教練。一位好教練明白團隊有時也可以應該失敗，因為那是達成成功專案與團隊的最有效果與效率的途徑。

這就是回顧特別有價值之處。我們學到 Scrum 及 XP 團隊如何利用回顧專案來找到改善的方法。對一位敏捷教練來說，這是一個幫助團隊從失敗中學習的好機會——而讓他們失敗的主要目標就是學習。若團隊是因導入的實踐不夠徹底而失敗，教練可以幫助他們學習如何做得更好的技巧。但有時是因為團隊需要改進心態所以才失敗的——而因為敏捷教練知道為何方法與實踐會有用，他或她可以利用失敗來幫助團隊學習更多有關可以幫助他們做出更好決定並避免失敗的特定價值或原則。這就是團隊成長的方式。

但一位好的教練也明白何時團隊是不能失敗的。就如同房子整建時有些樑柱和牆壁是不能更動的，每個方法論都有無法在不犧牲專案的完整性的前提下而不能被更動的「支撐樑」。一位教練需要對方法論有破階段的了解：他對其所以成為一套系統有深入的了解，也清楚其為何有用。在第 8 章中，我們學到老闆因要求團隊移除 WIP 上限而不慎把整個看板方法的努力都砍掉了。若是有守階段理解能力的人可能會讓 WIP 上限被改變，認為至少在某些地方有看板方法會比較好。一位好的看板方法教練了解當一些事情（像是在一塊看板上的特定欄位）可以被改變時，WIP 必須不變，因為這是整個看板方法維繫在一起的關鍵。

一位敏捷教練工作更難的部分是找到方法向團隊、老闆和客戶說明。一個簡單的守階段的對於方法論規則的說明就能搞定，但還不足以幫助團隊調整成「只比沒做好一點」結果的心態。有時團隊會滿足於遵循一組簡單的規則；其他時候，他們會覺得自己只是把一組自訂的瀑布式規則與一組自訂的敏捷規則做交換。一個破階段的理解會幫助他們看到為何遵循這些規則可以做出更好的軟體；但也有可能聽起來像 Cockburn 所說的如同「令人困惑的禪」。敏捷教練就是要幫助團隊用一個他們能掌握的速度，來邁向他們的破階段理解，並避免吸取太多對他們沒幫助的說明。

教練的原則

在本書中若有一件事是我們不斷強調的，那就是敏捷需要正確的心態，而團隊透過價值與原則取得這樣的心態。這就是為什麼每個敏捷方法論都有一組價值，而團隊成員們只有在了解與內化這些價值後，才能把敏捷方法論的潛力發揮出來。

所以教練有自己的一組價值並不讓人驚訝。John Wooden 是 UCLA 男籃隊在 1960 與 1970 年代的教練，他被公認為是運動史上最偉大的教練之一。在他的著作《*Practical Modern Basketball*》當中，他列舉了 5 項教練的基本原則：勤奮、熱情、條件、基礎與培養團隊精神。這些對於敏捷教練來說也是同樣有道理的：

勤奮 (Industriousness)

改變一個團隊建構軟體的方式，表示要努力做一件他們沒做過的事。開發人員必須思考規劃與測試，而不只是撰寫程式。產品負責人需要明白團隊在做些什麼，而不只是把功能需求丟給他們。Scrum 主持人需要學習如何賦予團隊控制權，同時仍能維持參與專案。這些都是新技巧，而新技巧需要花時間練習。

熱情 (Enthusiasm)

當你心在工作，而你對新的工作方式感到興奮時，就會把所有讓你分心的人從你身邊趕走。而有許多事可以讓你感到有熱情：你正在解決從前讓你感到頭痛的問題。當每個人都帶著熱情使用敏捷方法論時，你不只會得到更好的軟體；你會得到一個更有創新力、更快樂的團隊，而且每天都對一起工作感到興奮。

條件 (Condition)

當每個團隊成員都做著他或她所擅長的事時，敏捷就有用了。有個實際的間接假設認為，每個人都有一份對工作的驕傲：他們試著建構自己所能建構的最好的軟體，

並試著在建構時做得更好——而他們真的是被他們工作中的驕傲所驅動。這就是為什麼一個敏捷團隊的每位成員都持續認真地精進他們的技巧，因為這樣他們可以帶給專案最好的狀態。

基礎 (Fundamentals)

Wooden 寫道：「最精密的系統也會敗在沒顧好基礎。教練必須確定他一定不會被一個複雜的系統給『弄得暈頭轉向』」——特別是對一位敏捷教練來說更是如此。敏捷可行的原因在於具有簡單與直覺的價值，而其方法論是由不複雜的實踐所構成。這些是敏捷的基礎，而一位好的敏捷教練會讓團隊持續專注在這些事情上，並幫助團隊維持事情盡量簡單。

培養團隊精神 (Development Of Team Spirit)

我們已經討論過關於自我組織、整個團隊、充滿活力的工作以及授權團體：敏捷團隊讓彼此成長並創造了一個合作創新的工作環境的方法。另一方面，一個敏捷團隊需要注意那些只關注個人表現、職業規劃、履歷或升遷的團隊成員，並幫助他們改變態度。Wooden 針對這種人對於團隊精神的影響以及該怎麼做都說得很清楚：「教練必須斟酌每個命令並運用所有可得的方法來培養其團隊的良好團隊精神。團隊合作與無私必須在所有時候都被鼓勵，而每個團隊成員必須渴望（而不只是願意）犧牲個人榮耀以換取團隊的利益。自私、忌妒、自我與對彼此的批評，會毀掉團隊精神以及團隊的所有潛能。教練必須注意此事並時時提醒以防微杜漸。」

這 5 項原則是構成敏捷教練心態的堅實基礎。要成為偉大的敏捷教練的重要步驟，就是理解、內化，並且運用這些原則，就像在使用敏捷宣言的價值和任何敏捷方法論時那樣。



重點提要

- 敏捷教練是可以幫助團隊採用敏捷的那個人，同時也可以幫助團隊裡的每個人學習新的心態。
- 有效率的敏捷教練可以幫助團隊舒緩新實踐的不適感，讓他們只專注在感覺熟悉的那些實踐的部分。
- 教練知道團隊不喜歡改變的徵兆。
- 人們透過稱為 shuhari 的 3 個階段來學習新的系統或心態。

- 在第 1 個 **shu** 階段，有人首次鼓勵點子，此人還在學習規格——她通常會對特定的指令有最好的回應。
- 在 **ha** 階段有人開始學習比較大型的系統，並且調整心態以適應這樣的系統。
- 在 **ri** 階段，人們已經對於系統的概念非常熟悉——舉例而言，他們知道何時可以不必遵循規則。
- 有效率的敏捷教練知道為什麼系統可以運作，也理解為什麼會存在那些規則。



學習資源

下列是一些可以幫助你深入學習本章概念的資源：

- 更多關於敏捷教練的知識，可以參閱 Lyssa Adkins 所著的《*Coaching Agile Teams*》（Addison-Wesley, 2010）。
- 更多關於 shuhari 以及人們如何採用這套新系統的知識，可以參閱 Alistair Cockburn 所著的《*Agile Software Development: The Cooperative Game*》第二版（Addison-Wesley, 2006）。
- 更多關於教練的基礎知識，可以參閱 John Wooden 所著的《*Practical Modern Basketball*》第三版（Benjamin Cummings, 1998）。
- 更多關於團隊抵制改變的原因以及如何幫助他們不再反抗，可以參閱 Andrew Stellman 和 Jennifer Greene 所著的《*Applied Software Project Management*》（O'Reilly, 2005）。