

# 協同作業支援：下一步發展

級別：入門

伍斯特理工學院的實踐教授 [Gary Pollice](#)

2008 年 7 月 15 日

摘自《The Rational Edge》：參閱軟體開發市場的新興功能，以支援團隊工作環境中的協同作業與創造力。

摘自 [The Rational Edge](#)。

人類總是在追求可以改革產業的突破，我的志向通常沒那麼偉大。我追求的是下一步提升或發展，能將我們向前推進。曾經有些偉大的構想改變我們建立軟體系統的方式，例如遞增式、反覆式開發，但通常沒有任何一個構想完全改變我們對宇宙的觀點。

最近我對一個領域很感興趣，我認為可能會在軟體開發方面創造下一波的基礎變革，那就是：協同作業與創造力支援。敏捷的移動讓團隊持續且良好的通訊，成為眾所矚目的需求，強調優秀的開發人員對其他實務瞭解透徹，並將其編入不斷擴大的實務與原則集合中。敏捷法只是開發軟體的一種方式。我看過所有的成功專案，都有一個共通特性：整個團隊的通訊及協同作業都有效率。

整個團隊包括客戶、開發人員、管理者、優秀的專業人員、編寫人員，以及參與專案的任何其他人。無論專案是使用瀑布法、反覆法或敏捷法，亦是如此。

很多工具都聲稱可以支援協同作業，然而一直到最近，實際支援協同作業的程度卻很小。但是我也對一種新型工具抱持希望，相信我們會看到對團隊協同作業的實際支援大幅提升，緊跟在協同作業支援後面的是創造力支援。協同作業與創造力關係非常密切。本月我將著眼於協同作業、一些現在可供使用的工具，並展望此概念未來的發展。協同作業與創造力自動支援的成功開發，也許會讓生產優異軟體系統的能力邁向另一個層次。

## 瞭解協同作業與創造力

首先，確定我們是否對協同作業與創造力有共通的瞭解。雖然有大量關於創造力的著作，但在這二個概念中，協同作業較容易理解，因為在軟體開發的環境定義

中，我們對協同作業有較多的認識。針對本文的目的，我將協同作業定義為「一群人合作達成共同的目標」。就其本質而言，協同作業的意思就是有一個人以上的參與。協同作業也是目標導向的作業，許多人為了一個特定目標而協同合作。在軟體開發中，該目的就是在預算內準時順利完成開發及部署軟體系統。

協同作業需要溝通，但只有溝通並不夠。協同作業需協議協同合作的目標、瞭解如何達成目標，以及察覺協同合作的狀態。我會再深入探討這一點。

我說我們對協同作業的認識多於創造力，但以科學層面來看，我不確定協同作業的研究是否像創造力的研究那麼多，我們對協同作業的瞭解可能是直覺的。曾經參與過團隊活動的人，都瞭解協同作業如何運作。以我搜尋的協同作業與創造力研究報告來看，創造力的報告似乎比較多。然而，這二者有著密切的關連，從瞭解其本質的觀點來看，兩者有一些相同的特性。

在各種學科中，創造力一直都是部分學術研究的重點。心理學家試著瞭解創造力的本質。有創造力的人如何想出點子？有創造力的人，其工作型態為何？我們可以教人變得有創造力？社會學家、電腦學家及其他人都曾參與過這項研究，因為這非常重要。全球化變成正常而不是異常時，有些國家會因為可以提供比別人更便宜的服務而繁榮興盛，有些則會因為他們的創造力而興盛。相信未來我們會看到對生產力、服務成本與產品的需求，以及某些綜合體需要的創造力，以維持經濟競爭優勢。這是很大的挑戰，風險也很高。

我最近讀到有關此主題的報告，是由 **Mihaly Csikszentmihalyi** 所撰寫的。<sup>[1](#)</sup> **Csikszentmihalyi** 表示可以從三個構成要素來瞭解創造力：

1. **領域**：領域包含一組符號規則與程序，例如數學或物理領域。
2. **外野手 (Field)**：外野手是一組作為領域「守門人」的人。他們是該領域的元老，判定哪些東西值得成為領域的一部分。
3. **個人 (Individual)**：創造力的個人部分是指「一個人有新的想法或看到新的模式，以及適當的外野手將這個新事物選為相關領域的一部分」。

研究人員使用此結構，以指導他們瞭解創造力及支持創造力的方法。我會再繼續說明。

### 協同作業與創造力之間的關係

協同作業與創造力二個概念的關係為何？我對此關係的看法如「圖 1」所示。協同作業的定義是需要一個團隊，且協同作業的工作是以任務為基礎，但經常有一些探索工作，例如「極致軟體製程」(eXtreme Programming) 中的 "spike"。<sup>[2](#)</sup> 根據 **Csikszentmihalyi** 所給定的構成要素，我不確定是否有人會認為這樣的工

作具有創造力。創造性的工作即可以是個人的，也可以團隊為基礎，但主要是探索性的工作。

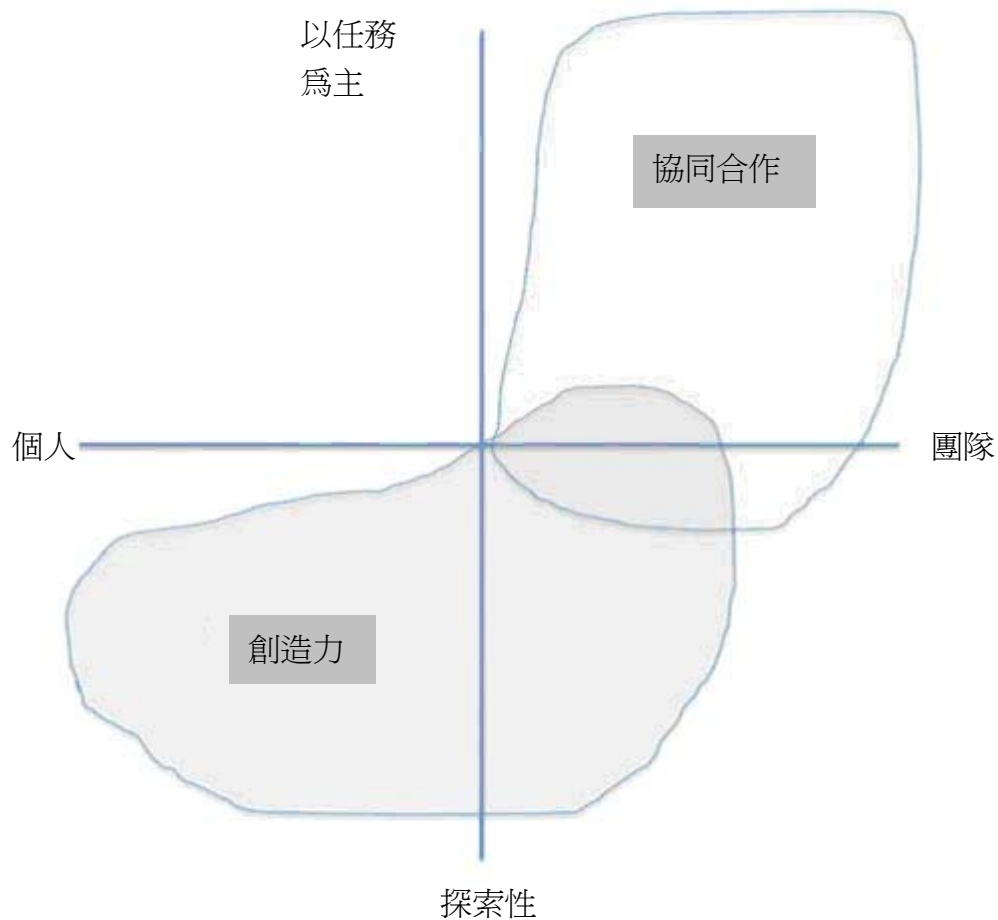


圖 1：協同作業與創造力之間的關係

在軟體開發的觀點方面，大部分軟體應用程式的撰寫，都是為了解決特定的問題。因此一般 IT 專案是以任務為主，並在「圖 1」的上半部分運作。然而，開發軟體工具的專案與其他具有類似情況的專案運作方式不同。一般個人或小型群組會投入創造力，開發工具真正的核心思想。一旦探究了各種方法，並選定其中一種，該專案就會轉換為協同作業模式，而團隊便在此模式下建置產品。<sup>3</sup>

### 支援協同作業

若要達成協同作業的目標有賴於幾點因素，其中最重要的部分因素如下：

- 清楚識別任務及被指派完成任務的團隊成員
- 察覺團隊成員的活動與進度，以及專案完成的整體進度
- 豐富的歷史記錄，提供專案在過去任何時候的狀態檢視

- 有效的同步與非同步通訊

## 協同作業工具的歷史

要支援成功的協同作業，就應該要支援這些因素。一起看看工具如何支援這些因素。我將從最簡單且最古老的工具開始，一直到我們現在使用的工具。

十多年來，協助團隊識別任務以及任務的指派，一直都是專案管理工具的一部分。針對非常簡單的專案，可以使用含有少數巨集的試算表。擁有大型團隊、必須整合多重構成要素，以及開發週期很長的專案，就要使用精密、複雜的專案排程工具。

專案管理工具對協同作業是有一些支援，但我們不一定會將其分類為協同作業支援工具。其中一個原因是，這些工具不會讓團隊成員輕易察覺隊友的活動。如果您是開發人員，且您的專案使用精密的專案管理工具，那麼請問您上次使用該工具來知道另一個開發人員在做什麼，是什麼時候？答案非常可能是「從未」。

察覺是一個很重要的議題，一個好的協同作業工具，會讓團隊成員輕易察覺團隊夥伴在做什麼。這樣的工具還可讓您不費吹灰之力，就把專案的整體進度資訊告知團隊的每一個人。這是生產力的問題。如果要花時間尋找所需的資訊，那些時間其實可用在更重要的任務上，例如，為實作新功能進行編碼。此外，如果要花時間與精力來找尋資訊，那大家就不會經常使用這些資訊，他們只會在發生危急時使用。在這種危急的情況下，都是以反應模式來使用資訊。

主動使用專案進度與任務指派資訊會更有意義，資訊會更有價值。試想一個程式設計師被指派在應用程式上實作新功能，她進行這項工作時，發現系統的另一個部分將會使用她正在實作的功能。她可以只實作該功能，將其移入後，需要使用該功能的程式設計師就必須調整其編碼。如果她可以快速找出將利用其功能的人，就可以與該程式設計師一起合作，以最適合雙方需求的方式來實作該功能。這就是我所謂的主動使用資訊。

豐富的歷史資訊有許多用處，至於協同作業，這些資訊可以讓團隊回顧專案，瞭解在哪裡可以做出更好的選擇，或是回復到前一版專案，再找尋其他路徑。許多年前，版本控制系統就已經有許多用來擷取豐富歷史資訊的功能。長久以來，**IBM Rational ClearCase** 一直都是軟體配置管理 (SCM) 工具市場中的最佳入口。該產品的設計中，包含分支、合併以及進行並行開發的能力。這樣的產品擁有提供豐富歷史的基礎，但我們還需要更多。

## 通訊的角色

我們希望能夠回到可以看到專案全貌的時光，此全貌包含保存在版本控制系統中的構件，但也包含了不容易保存於一般版本控制系統中的資訊。此資訊可能包含電子郵件、網路會議記錄等。這些資訊也必須以鼓勵探索及協同作業的方式呈現，

這些都需要通訊，通常面對面或透過電話與人交談，比寫電子郵件更有效率。我們進行即時交談時，比較能夠釐清問題及溝通意念。在當今高度分散的環境中，並不是隨時都能做到這一點，且事實上，人們身處同一個地方的情況越來越少。我們希望在可以的時候盡可能同步通訊，但如果做不到，就需要有效的非同步通訊。

通訊可產生對開發專案有價值的構件，也就是團隊應用於協同作業的資訊。我們面臨的挑戰之一是，要如何依需要擷取通訊，並讓團隊獲得這些資訊？最近我參與了幾個 **Worcester Polytechnic Institute (WPI)** 的學生專案，他們已經開始處理這個問題。

#### 目前的協同作業支援工具

上述的評論是十分籠統，現在我想要說明目前可支援協同作業的工具類型。此清單並不詳盡，但提供一個目前從業人員可使用的適當範例。

**線上協同作業社群及專案環境**，可供團隊起始專案及協同合作的線上社群數量急遽增加。其中，像 **sourceforge.net** 與 **Java** 社群專案就是開放原始碼社群的二個實例。<sup>4</sup>商業產品可支援相同的功能，可為想要管理自己的專案產品組合組織，提供更多支援。只要組織有使用這些產品的適當程序，即可利用這些產品，在分散的團隊間展開協同作業。

**版本控制及原始碼管理系統**，對任何團隊（無論分散或集中），都是不可或缺的。上一個類別中的大部分產品都含有這些產品，或提供對這些產品的介面。

**錯誤追蹤工具**，雖然這些工具只提供非同步功能，但其為開發團隊另一個不可或缺的需求。這類工具提供專案性能（缺點數量）及任務指派資訊的察覺功能。

**網路會議工具**。有些網路會議工具是隨著網際網路及網路技術發展為成熟、快速的通訊媒體而發展的。一般人會在廣播與電視上聽到這類工具的廣告，以及在許多流行的雜誌上看到其相關廣告。這些工具的功能互異，但是其本質都是要讓一組人啓用視訊、音訊及共用的電腦桌面功能，進行虛擬會議。這類工具所提供的經驗品質仰賴許多因素，包括參與者接受這些工具並學習適當使用的意願。

**IDE 內嵌的協同作業功能**。許多互動式開發環境都含有協同作業功能，**Eclipse** 平台率先採用許多外掛程式，讓 **Eclipse** 使用者能夠進行協同作業，且更容易

操作。較成熟的專案與外掛程式之一是 Mylyn 專案<sup>5</sup>。Mylyn 為程式設計師的工作，提供以任務為主的工具，也整合數項錯誤追蹤工具。在 WPI 中，我們有 Webfoot 專案<sup>6</sup>，這是持續進行中的專案，為 Eclipse 使用者提供豐富的協同作業功能。Eclipse Communication Framework 是很適合用來開發工具的技術架構<sup>7</sup>。該專案已經進行好幾年，為開發人員提供用來建置自己工具的功能，包含一些範例應用程式，例如具有最新 Eclipse 版的即時共用編輯器，而其他 IDE 新增許多 Eclipse 包含的功能。NetBeans IDE 曾經有一個非常好的共用編輯功能，其他工具則提供更專門的功能，例如 Smart Bear 的 Code Collaborator，

但可惜的是，與這些工具的互動並沒有什麼標準，還限制開發團隊的選擇。整個團隊都必須使用相同的 IDE，才能獲得協同作業功能的好處。

**持續整合工具**，在最近幾年開始流行，與 SCM 系統結合，可自動觀察程式碼儲存庫，並定期或在有任何變更時建置系統（包括執行測試與包裝構件）。也能夠通知整個團隊，或團隊中被選定的成員。通常會提供儀表板，可立即提供目前的建置狀態。

**Jazz**：下一個階段

看到 IBM Rational 的 Jazz 平台上市，我真的非常開心。<sup>8</sup>會將有效協同作業所需的許多功能放在一個平台中，且會建置在 Eclipse 平台上，所以您不需要學習很多新工具。Rational Team Concert (RTC) 產品是 Jazz 產品線中的第一個產品，我在 WPI 用過早期的版本，覺得用起來很有趣，設定與管理都很簡單，我與一些學生測試的結果，證明真的有用。我希望今年能夠將其應用到我的教學中，並讓學生團隊使用。

RTC 的三個版本有一系列令人印象深刻的功能，如「圖 2」所示，相較於我認為必要的協同作業屬性，您會看到這些十分符合。

我特別喜歡 RTC 的地方，是其提供的察覺功能，以及程序自訂潛能。現在，團隊成員可以看到其他人在任何給定的時間做些什麼，如果有問題提出，而知道答案的人在線上，程式設計師就可以輕易地合作找出答案，確保他們對任何問題都能達成共識。在我看來，這是向前邁出了一大步。

在本文中，我沒有談到很多關於程序的事。然而，無論有沒有寫出來，每個團隊都有一套程序，而 RTC 提供一個現成的簡單敏捷程序，該程序很容易自訂。我還沒有對該程序進行很多自訂，但我希望今年可以給我的學生一個完整、簡單的程序。

如果您還沒使用過 RTC，那麼建議您試用看看，前三名使用者可以免費獲得 Express 編輯器，可讓您體驗到我剛剛所談論的。

主要功能	Express-C	Express	標準
可自訂的程序	✓	✓	✓
可自訂的工作項目屬性及工作流程			✓
報告			✓
伺服器層級權限	✓	✓	✓
儀表板			✓
敏捷計劃	✓	✓	✓
工作項目追蹤	✓	✓	✓
以角色為基礎的程序權限			✓
ClearCase 連接器			✓
LDAP 匯入			✓
子版本整合	✓	✓	✓
ClearQuest 連接器			✓
建置管理	✓	✓	✓
原始碼管理	✓	✓	✓
LDAP 鑑別	✓	✓	✓

圖 2：IBM Rational Team Concert 功能

## 結語

目前是協同作業支援工具改革的開始，Jazz 雖然令人印象深刻，但只是個開始，希望有很多軟體工具廠商會開始發行其他協同作業功能。我希望這些努力會產生一些合理的互動標準，讓開發人員能夠自由採用他們覺得最適合且生產力最高的工具，並能夠與使用不同工具的團隊成員協同合作。

我期望對創造力的支援可以與協同作業支援同等重要，變得愈來愈穩固。我將在日後的文章中討論創造力支援。

## 參考資料

1. Csikszentmihalyi 在 Flow 的書已經成為最受歡迎的暢銷書，尤其是一些經理人想要找出方法，讓其團隊針對可轉化為商業成功典範的問題，開發出具有創造力的解決方案。

2. "spike" 是指一小段時間，用來探討如何運用新技術或不同方法來解決問題。
3. 對於獨自一個人做的專案，我比較不感興趣，反而認為一個人以上才能創造出最有趣的軟體。
4. 請參閱 <http://www.sourceforge.net> 及 <http://community.java.net/projects/>
5. <http://www.eclipse.org/mylyn/>
6. <https://sourceforge.wpi.edu/sf/projects/webfoot>
7. <http://www.eclipse.org/ecf/>
8. <http://www-306.ibm.com/software/rational/jazz/>

協同作業與創造力的參考資料

- [1] L.T. Cheng, C. De Souza, S. Hupfer et al., "Building Collaboration into IDEs," *ACM Queue*, vol. 1, no. 9, pp. 11, 2004.
- [2] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*, New York: HarperCollins, 1990.
- [3] M. Csikszentmihalyi, *Creativity: Flow and the Psychology of Discovery and Inventory*, New York: HarperCollins, 1996.
- [4] M. Csikszentmihalyi, *Finding Flow: The Psychology of Finding Engagement with Everyday Life*, New York: Basic Books, 1997.
- [5] T. Hewett, M. Czerwinski, M. Terry et al., "Creativity Support Tool Evaluation Methods and Metrics," in *Creativity Support Tools*, Washington, D.C., 2005, pp. 16.
- [6] B. Schneiderman, "Creativity Support Tools Accelerating Discovery and Innovation," *Communications of the ACM*, vol. 50, no. 12, pp. 13, 2007.
- [7] B. Schneiderman, G. Fischer, M. Czerwinski et al., "Introduction to Workshop Report," in *Creativity Support Tools*, Washington, D.C., 2005, pp. 7.
- [8] B. Schneiderman, and others, "Report of Workshop on Creativity Support Tools," [http://www.cs.umd.edu/hcil/CST/Papers/creativitybook\\_final.pdf](http://www.cs.umd.edu/hcil/CST/Papers/creativitybook_final.pdf) [July 15, 2005].



[9] B. Schneiderman, and others, "Creativity Support Tools: Report from a U.S. National Science Foundation Sponsored Workshop," *International Journal of Human-Computer Interaction*, vol. 20, no. 2, pp. 17, 2006.

#### 資源

- [參加論壇](#)。
- 我們已經為 *Rational Edge* 文章特別開闢一個[新論壇](#)，所以您現在可以依目前的議題或我們的保存檔中，分享您對這篇文章或其他文章的看法。看看您的全球同事的看法、建立您自己的討論，或是加入進行中的討論。按一下[這裡](#)。
- [全球 Rational 使用者群組社群](#)

#### 關於作者

**Gary Pollice** 是伍斯特理工學院 (位於美國麻薩諸塞州 伍斯特市) 的實踐教授，教授軟體工程、設計、測試及其他電腦科學課程，他也是學生專案的指導教授。進入學術界前，他花了 35 年以上的時間開發各種軟體，從商業應用程式到編譯器與工具都有。他在業界的最後一份工作是在 IBM Rational Software 服務，大家稱他為「RUP 倔老頭」，同時也是 Rational Suite 創始團隊的成員。他是 2004 年 Addison-Wesley 出版的《*小型團隊軟體發展：以 RUP 為中心的方法*》一書主要作者。他擁有數學學士及電腦科學碩士學位。