RUP®/XP 準則：雙人程式設計

Robert C. Martin
Object Mentor, Inc.

Rational Software 白皮書

TP 158, 3/01

Rational®
the software development company

目錄

概觀

雙人，簡要說明

雙人程式設計是一種技術，由雙人程式設計師撰寫專案的軟體。雙人在單一工作站一起工作。由雙人中的一個成員驅動工作站，另一人則小心監督所產生的程式碼。驅動者做戰術性地思考，關注他目前撰寫的程式碼行。觀察者驗證語法，並從策略上思考整個程式。他們經常交換角色，所產生的程式碼之撰寫速度比單人撰寫更快，而且更少出錯。而且，至少有兩位開發人員熟諳此程式碼。

雙人案例

考量一般程式碼複查階段作業。一人需要八小時開發的模組，由八人花一小時複查。最終結果是在該模組上花了十六小時。然而，複查人員無法花費所需的時間來熟悉該程式碼，而且他們的複查相當初淺。單一開發人員雖然熟諳程式碼，但或許太過熟悉而無法找到問題主體。

請對照它與雙人程式設計的作法。如果雙人開發模組需要八小時，總共要花費十六小時。然而，在此情況下，有*兩個*開發人員會熟諳程式碼。一個開發人員看不到的問題，另一個人會看到。

雙人程式設計的案例很簡單，但其影響微妙而不可捉摸。雙人程式設計是撰寫和複查程式碼比較有效的方式。有兩人熟悉模組，則撰寫到程式碼的問題會減少很多。程式碼將具有更好的結構，而對它的熟諳程度則是許多人的兩倍。如果這些是唯一好處也就罷了，但雙人行動還提供更多好處。

雙人更有信心：單一程式設計師可能害怕嘗試，但雙人則具備嘗試的勇氣和評估的技能。

雙人培養團隊合作：由於模組不是由單人撰寫，所以程式碼成爲團隊財產，而不是特定開發人員的財產。

雙人促進知識的散播：有越多開發人員組成雙人組，就會有更多的系統知識散播到整個團隊。其結果爲團隊的成員熟悉整個系統，而不是每一個成員只知道他們自己特定的那個部分。

雙人提高生產力：單人程式設計經過能量爆發之後，接著就是相對休止期。雙人可調整彼此的步調。當一人倦怠時，雙方可交換角色。他們能夠保持強勁有力，比單人可容忍的時間更久。

雙人增添工作樂趣：與另一位開發人員一起工作具有教育性、激勵性和趣味性。雙人提高工作滿意度和整體士氣。

作法

雙人

當負責某項作業的開發人員要求別人協助時，即形成雙人作業。規則是：別人一問起，您就要答應。這不表示您必須立即停止手邊的工作。而是表示您必須協商出一個您可以提供協助的時間，以及您可以因此獲得協助的另一個時間。

您的夥伴不必負起該作業的責任。此責任仍屬於身爲作業擁有者的您。您的夥伴也不承諾作業完成之前一直留在擁有者您的身邊。您的夥伴只承諾提供協助。

雙人的成員之一變成驅動者，另一人變成觀察者。驅動者輸入程式碼、執行編譯器、執行單元測試等等。觀察者檢查每一個按鍵、每一個指令、每一個測試結果，並提供協助和建議。雙方全天候參與。

有時候驅動者最知道該做什麼，觀察者只要服從就好。有時候，觀察者會要求驅動者執行動作。有時候，驅動者感到沮喪，而將鍵盤交給觀察者，於是兩人交換角色。有時候，觀察者會主動要求鍵盤及交換角色。在雙人階段作業中，這種事經常發生。

變更雙人組

雙人夥伴不是長期夥伴。一般雙人階段作業大約持續半天左右。任何夥伴都可以基於任何原因而拒絕成爲夥伴。發生此情形時，作業的擁有者必須尋找另一個雙人夥伴。這可能表示作業擁有者這時候該報答上週的雙人夥伴。在另一方面，也許他（她）應該請一位有適當經驗的人幫忙處理特別棘手的問題。

雙人的變更可使系統的知識散播到整個開發團隊。在短時間內，團隊的每一個成員將花時間來處理幾乎系統的每一個部分。這大大減少專案更換人員的敏感度，也使每一個程式設計師更有信心處理整個系統。

群體擁有權

由於每一個人都處理系統中的所有不同模組，因此沒有人擁有特定模組。這表示系統的責任不是按逐一模組的方式來劃分。而是整個團隊共同負責整個系統。團隊任何成員可以基於任何原因移出及變更系統中的任何模組。當雙人組變更模組 X 而造成模組 Y 的單元測試失敗時，雙人組要負責修復模組 Y。

調整步調與合作

雙人程式設計是非常密集的溝通形式。口頭對話常常變得爭論不休，外在觀察者可能很難理解。身爲觀察者，您可能會聽到雙人組說出這些字眼：「分號」或「右大括弧」。或者，當程式設計師同意或不同意螢幕上出現的程式碼時，您只聽到他喃喃自語。兩人埋首於程式碼中，顯然很多溝通都是非口語的。這裡，肢體語言就扮演很重要的一部分。一個雙人夥伴告訴他的夥伴他不滿意該程式碼時，不發一語。扮個鬼臉、嘆一口氣、坐立不安—全都是爲了增加夥伴之間溝通的頻寬。

有時候一位夥伴抓著滑鼠，另一位則操作鍵盤。持滑鼠者控制模組內要進行工作的位置。持鍵盤者控制在該位置改變或新增的內容。有時候，一位夥伴輸入，另一位夥伴預見函數呼叫進入，並在撰寫程式者正好需要此規格時，將 API 文件開啓到正確頁面。

當一位夥伴倦怠時，另一位可以帶頭工作，讓夥伴扮演觀察者角色，休息一下。有時候，兩位夥伴精力充沛，常常交換鍵盤和滑鼠。

總而言之，規則和程序變少了。唯一的約束是雙方都必須保持參與感，兩人之間要加強溝通。一位夥伴在輸入資料時，另一位夥伴若無其事望著窗外，這樣子不是真正的雙人組。

孤單無伴的開發人員可以做什麼呢？

您不能一直當雙人組。有些專案—亦即，那些採用 *eXtreme Programming* (XP)（請看參照 [1]）流程的專案—遵循雙人組必須產生所有正式作業程式碼的規則。在此情況下，當您不是雙人組時，可以檢查電子郵件，研讀新技術或 API，詳讀您不熟悉的程式碼，或與關係人談論有關現行反覆或未來計劃。固然，在找不到雙人夥伴時，開發人員總是可以在那幾小時內做些有用的事。

有些專案對雙人組比較沒有那麼嚴格的要求。有些專案可讓單人開發人員撰寫測試碼。有些專案可讓單人開發人員撰寫抽象類別或介面。有些專案只是讓開發人員決定何時最適合雙人組。不過，有件事可以確定—根據研究結果顯示，實施雙人組時，錯誤率大幅下滑。

有些人不喜歡雙人組

有些人不喜歡雙人組的概念。根據我們的經驗，有實際嘗試一週左右的人，會發現他們的不適感消失了，並且樂在雙人組，覺得它很有幫助。很少人會繼續厭惡此作法。因此，對大部分人而言，只是嘗試和習慣的問題而已。對於那些認真嘗試過之後仍然覺得厭惡此作法的人，團隊必須尋找適合他們做的工作。

設備、場所和底層機制

螢幕和鍵盤佈置

設備佈置是成功雙人組的一大關鍵。如果夥伴座位不在旁邊，不能快速交換鍵盤，則雙人組成效不彰。規則是：您必須能夠不更換座位就可以來回遞送鍵盤和滑鼠。

最好的安排通常是一個長型平面桌。將螢幕放在中間，讓兩把座椅面對螢幕。坐下時讓螢幕在兩人之間。確定可以在兩人之間輕易來回滑動鍵盤和滑鼠。並確定您持有鍵盤時，您的座位舒適挺直。確定螢幕不必旋轉就可以同時讓兩人看到。

投手暖身區

爲了幫助更換雙人組夥伴，以投手暖身區的安排方式來工作是明智的作法。在一間房間內放置數個雙人工作站。使用滑輪座椅和亞麻油地氈或磁磚地板。安排工作站讓雙人組彼此面對面。其目標是要增加溝通機會。有時候最重要的溝通是僥倖得來的。我們想要增加這類溝通發生的機會。

內角

現在有許多小隔間將工作站放在內角。開發人員座位面對小隔間的一角，前面是螢幕。這對個人工作很方便；但雙人組在這種環境下幾乎不可能存在。如果您有工作站放在內角的小隔間，請在別處設置一些雙人組工作站—會議室也可以。在使用筆記型電腦的會議室，雙人組會很有效率。

問題和考量

雙人組平分生產力

理所當然，兩人共同處理一項作業所花費的每人時數是一人處理同一項作業的兩倍。這看似合理，其實不然。根據獨立研究結果（請看參照 [2]）顯示，雙人組工作不會損失多少生產力。那些相同研究結果顯示，雙人組實際上會減少錯誤率*和程式碼規模*，還可大大提高工作樂趣。

雙人組夥伴之間的爭論

作業擁有者對所有設計爭論有最後決定權，但是解決紛爭的最好方式是兩種都試試看，再選擇其中最適合的一種。

專家

傳統看法是建議特定領域（例如資料庫或 GUI）的專業開發人員應該只將精力投入那些領域。然而，在雙人程式設計環境中，那些專家變成不對他人分享其專業的顧問。開發人員可進一步瞭解其專業以外的作業，然後向專家取得協助。如此一來，專家的知識就可以透過專案小組散播，大大減少專案更換人員的敏感度。

噪音

雙人組在寫程式時會製造噪音。充滿雙人組的投手暖身區不斷有低音量的嘈雜聲。有人覺得噪音令人討厭，會分散注意力。但這不算什麼大問題。如果您覺得噪音會分散注意力，可搬到會議室一陣子。

牛仔

許多團隊對於他們裡面有一或多個牛仔型撰寫程式者感到驕傲。這些程式設計師完成工作的速度比別人快，無法和他人共事，不准別人讀取其程式碼（或經允許才可以讀取程式碼）。要對付這些開發人員，最好的方式是使他們離開專案，或變成不撰寫正式作業程式碼的角色。也許他們可以撰寫短期工具或做一些折磨人的測試工作。

物質障礙和風格障礙

有些人使用 QWERTY 鍵盤。有些人喜好 DVORAK。有些人需要特殊鍵盤、滑鼠、顯示器、腳踏開關等等。有些人沒有耳機和高分貝的音樂無法寫程式。有些人必須在旁邊準備好多零食才行。有些人喜歡 emacs。有些人愛用 VI。有些人仍然想要使用 WordPad 來工作。

固然，障礙多的不勝枚舉。但每一樣都可以利用一點點心思和妥協能力獲得解決。讓這點小事阻礙的團隊，是一個無論如何努力都不會成功的團隊。

團隊如何規劃雙人組？

我們不認爲作業應該指派給雙人組。而應該由每一個開發人員負責一組作業。我們希望雙人組自然形成。每一個盡責的開發人員會要求其他開發人員給予短暫的協助。作業的擁有者保有擁有權並負有責任。雙人組夥伴只是幫手。

每一個開發人員在提供作業預估時必須考慮他（她）在雙人組工作中的時間量。

結論

雙人程式設計是程式碼複查的替代方法，通過測試且接受度很高。不僅如此，它是在根本上完全不同的撰寫軟體方式。它的好處不只在於生產力和品質，還影響團隊的健全和士氣。

參照

[1] *eXtreme Programming eXplained*, Kent Beck, Addison Wesley, 2000.

[2] *Strengthening the Case for Pair Programming*, Laurie Williams, University of Utah, July/Aug 2000 IEEE Software.

Rational
the software development company

兩個總公司：

Rational Software
18880 Homestead Road
Cupertino, CA 95014
電話：(408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
電話：(781) 676-2400

免付費專線：(800) 728-1212
電子郵件：info@rational.com
網址：www.rational.com
國際辦事處：www.rational.com/worldwide