

雙人編程環境下人格特質與編程績效關係之研究

李有仁

國立政治大學資訊管理系

鄭江宇

國立中央大學資訊管理系

嚴秀茹

國立清華大學服務科學研究所

林旭峰

軟體專案顧問師

摘要

近年來敏捷軟體開發流程發展迅速，其中以雙人編程之協同概念最受到重視。然而到目前為止我們對於「什麼樣的人格特質組合得以提升編程績效」之議題仍缺乏相關了解，特別是在中小型軟體產業上。有鑑於此，本研究以人格特質量表（MBTI）、搭配不同之工作形態，試圖透過實驗設計方式來找出最佳配對組合。研究結果顯示：（1）專案經理必須了解，不同特質的雙人組合其編程績效比相同特質組合來得佳。（2）外部工作內容宜指派給具有外向特質的編程人員，而內部工作內容則適用於具有內向特質的編程人員。（3）感知式思考之編程人員較適合處理複雜度較低的工作，而複雜度高的工作則適合指派給直覺式思考之編程人員。

關鍵字：人格特質、雙人編程、編程績效

The Relationship between Personality Trait and Programming Performance under Pair: Programming Environment

Eldon Y. Li

Department of Management Information Systems, College of Commerce

C.Y. John Cheng

Department of Information Management, College of Management

H.J. Rebecca Yen

Institute of Service Science, College of Technology Management

Hsu Feng Lin

Independent Software Project Consultant

Abstract

The growth of agile approaches has attracted considerable attention to collaborative programming. However, it is not clear what specific pairs of personality traits could be suitable for increasing performance in collaborative programming. To understand this better, we conduct an experiment with the dimensions of personality and task type. The results suggest: (1) Project managers should be aware that complementary composition of personality types tends to outperform the composition single personality type. (2) Tasks involving external interfaces are better assigned to extrovert programmers. Conversely, tasks concerning the internal logic of a system are better assigned to introverts. (3) For ‘sensing’ programmers, low-complexity tasks form the best choice. Conversely, high-complexity tasks are more suitable for ‘intuitive’ persons.

Key words : Agile approach, personality, pair programming, project managers

壹、導論

軟體專案執行效率不彰始終困擾著軟體開發業者，其中以專案執行成本與執行進度延宕最受到關注（Gorla & Lam 2004）。然而試圖以預算縮減或加快專案執行速度為手段來解決上述問題，恐怕事與願違。根據Jeffery與Votta (1999)的說法，軟體專案係由許多複雜元素所組成之互動實體，包括人（human）、組織（organization）、技術（technology）等。這三者之中又以「人」最為重要，主要的原因在於編程人員通常擁有若干技能，如果缺乏這些智慧資本組織也將無法順利運行。因此在軟體專案管理方面，人事費用往往成為組織最大的成本支出（Howard 2001）。如何藉由人事成本支出得到最大化效益，實為軟體開發業者之一大挑戰。根據資策會資訊市場情報中心 (2006)所做的調查報告指出，我國軟體產業在工程能力和技術支援方面較為突出，而在流程及專案管理能力上則比較弱勢，特別是在開發流程部分。所幸於1999年所提倡之極致軟體製程方法（eXtreme Programming approach）即特別針對軟體開發流程中，專案人員配置問題提出不同見解。Beck (1999)認為相較於傳統單人編程方式，雙人編程（Pair programming）可以有效地透過兩位編程人員共同合作來提升軟體專案執行績效。此後，陸續有許多學者（Gorla & Lam 2004; McDowell et al. 2006; Williams et al. 2000）於不同研究情境下提出類似看法，他們指出雙人編程可以有效地（1）降低程式碼錯誤、（2）提升程式品質、（3）加速專案完成時間、（4）增進編程人員信心。

透過文獻得知雙人編程可以為軟體專案管理帶來許多好處，但仍有其先天上的限制（Olson et al. 2002）。根據定義，雙人編程的實施必須配置兩相鄰編程人員，其中一人擔任驅動者（driver），另一人則擔任領航員（navigator）。驅動者主要工作為程式碼撰寫，領航員則是在旁檢視編程過程中可能發生之缺失。然而Howard (2001)卻認為人們會依據其固有之人格特質來選擇做事方法，導致每一位編程人員形成其獨特地問題思考模式與解決習慣，一旦編程人員彼此之間因人格特質不相符而產生衝突，勢必阻礙專案之執行。矛盾地是，專案團隊運作通常少有單打獨鬥現象，加上專案成員對於集思廣益的渴望，因此編程人員妥善配置之重要性不言而喻（Coplien 1995）。除了人格特質會對專案運作造成影響之外，Sjoberg et al. (2002)認為工作型態（task type）以及工作環境（working environment）亦會對專案執行效率帶來影響。根據統計在軟體專案運作過程中，大約有50%時間是花費在問題討論與解決之上（DeMarco & Lister 1999），Teasley et al. (2002)更指出在專案進行的過程中，與他人共事之時間竟高達70%。雙人編程情境亦是如此，畢竟此種軟體開發流程之精神即在於要求兩配置編程人員，針對所指派之工作進行充分之溝通，因此除了人格特質之外，工作型態與工作環境確實有可能會影響到最後的專案執行效率。

雖然過去文獻已針對人格特質、工作型態、工作環境，加以探討此三者分別對編程績效之影響，然而本研究認為在雙人編程環境下，此三者係同時存在且無法分離因素，分別探討其各自對編程績效影響之適切性有待商榷。有鑑於此，本研究以軟體產業編程

人員為對象，試圖探究不同「人格特質」配對與不同「工作型態」對於雙人編程績效之影響。意即，在雙人編程環境下，不同人格特質組合是否仍會影響編程績效？不同的工作內容或是不同的工作複雜度是否需要特定的人格特質組合才能實現編程績效？參與雙人編程的人員是否對於其工作夥伴以及新方法之導入感到滿意？若能回答這些問題，將有助於企業提升軟體專案管理之績效。

貳、文獻探討

一、雙人編程

雙人編程係指兩編程人員一起坐在電腦前面，共同針對所交付之工作，以協同方式來完成相關的設計、演算法、編程等（Williams & Kessler 2000），其中一位扮演操作者，控制鍵盤及滑鼠，實際進行撰寫的工作。另一位則扮演監督者角色，幫助檢查程式開發方向是否正確並且檢查語法上之錯誤。兩人可以針對有困難的部分進行腦力激盪，以求得更好的解決方法，同時彼此也可以視情況互換角色，以達到更好的開發效果。自1990年代晚期，若干研究開始對雙人編程的開發效能進行評估，在實驗報告中可以發現雙人編程明顯地增進編程效能以及提高編程人員的工作滿意度。Nosek（1998）亦指出實行雙人編程可以減15%的錯誤，且92%編程人員會更享受他們的工作，同時96%編程人員對自身表現更有信心。Williams et al.（2000）於猶他大學嚐試評估雙人編程的產能及品質，其實驗結果指出雙人編程可以10%的程式品質，而96%編程人員也會比單人編程時更享受編程工作。

近年來雙人編程環境被大量應用於在許多大學程式設計相關課程上（Williams et al. 2000; Williams et al. 2002; Bevan et al. 2002; McDowell et al. 2002; Nagappan et al. 2003; Thomas et al. 2003; Schneider & Johnston 2005; Hedin et al. 2005; Gehringer 2003），研究結果發現雙人編程已成為教學上有力的工具，確實可以增進學生課堂上學習效果。尚有一些研究開始將雙人編程應用於分散式編程環境中（Williams & Kessler 2003），而Kircher et al.（2001）的研究報告則是探討雙人編程應用於西門子（Siemens）公司位於德國與印度的跨國開發人員合作上，他們利用電視影像做為媒介進行雙向的溝通，同時也建議使用pcAnywhere和NetMeeting來從事相關程式碼分享。相信也有人會對雙人編程的效益產生懷疑，陸續也有一些研究從經濟學角度來探討雙人編程之效益（Cockburn & Williams 2001）。大多數人剛接觸雙人編程的概念時，都會存有疑慮，認為雙人編程必然會花費雙倍的人力來開發程式，但從上述研究結論中可以發現，雙人編程只會比單獨編程多花15%的時間完成工作，但卻可降低15%的程式碼錯誤。也就是說，在提高程式品質之前提下，為了要達到跟雙人編程環境相同的程式品質時，單人編程會比雙人編程多花費40%~50%編程時間。Constantine（1995）亦表示：「雙人編程並非多餘，反而可以提高編程效率與改善軟體品質」，甚至使程式設計師信心與成就感兼備（Williams et al. 2000; Williams & Upchurch 2001）。

二、編程人員之人格特質

Hohmann (1997) 認為所謂人格特質是指「穩定情感特徵之複雜集合，而此情感特徵正好可以用來界定人類的行為傾向」。一般而言，人格特質可以區分成內向與外向兩大類別。不論是內向或外向的團隊成員，都有其獨特之行為特徵，例如Yellen et al. (1995) 在其研究中指出，相對於內向成員，外向成員具有更佳的群體決策能力。因此除了技術能力之外，人員組成勢必會對專案執行績效帶來不小之影響 (Guinan et al. 1998; Howard 2001; Rasch & Tosi 1992)，而這也衍生出「什麼樣的專案成員適合在一起共事？」之問題。在雙人編程環境下，Williams與Kessler (2002) 亦曾探討內向及外向編程人員彼此間配對的問題。其研究指出，若把內向與外向編程人員配對在一起，也許對於編程時間不減反增，但若是專案目標為知識傳授（如資深員工傳授新知予後進人員），此時內外向編程人員將會是最佳組合，畢竟外向人員本身即具有暢談的特質。因此本研究引用Williams與Kessler (2002) 做法，除了探討內向與外向兩種人格特質之外，額外加入感知 (sensing) 及直覺 (intuition) 兩種人格特質來分析編程人員之特質。根據Myers et al. (1998) 說法（如表1所示），內外向人格特質係屬於社會互動 (social interaction) 範疇，主要在於形容人如何被驅使進而表現出特定傾向，例如內向的人較為內斂沉穩但不擅長與外在世界互動，而外向的人則較為活潑，富有與外在世界互動之熱情。感知與直覺則是屬於資訊收集 (information gathering) 範疇，重點在於說明人如何處理所收集到的資訊，即感知的人較偏好以明確、具體的方式來處理資訊，而直覺的人較能夠容許模糊、不精確的資訊處理方式。由此可知「社會互動」與「資訊收集」在雙人編程環境下格外地重要，兩編程人員除了需要彼此溝通之外，有時甚至需要與顧客協商，藉由互動結果來收集並處理編程相關資訊，因此實有納入研究之必要。至於另外兩個維度，決策制定 (decision making) 與外在世界處理 (dealing with the external world) 則不列入本研究探討範圍。主要的原因在於，雙人編程之決策係經過雙方互動而產生，我們難以判斷究竟某決策是出自於那一位編程人員，至於外在世界處理方面則與研究主題無關，其主要描述人們如何選擇其特有之生活型態。

表1：人格特質維度 (Myers et al. 1998)

維度	描述	特質型態
社會互動 (Social interaction)	人們如何被驅使，進而表現出特定傾向。	外向 (Extrovert) / 內向 (Introvert)
資訊收集 (Information gathering)	人們如何處理所收集之資訊。	感知 (Sensing) / 直覺 (Intuition)
決策制定 (Decision making)	人們如何從事決策制定。	思考 (Thinking) / 感受 (Feeling)
外在世界處理 (Dealing with the external world)	人們如何選定其生活型態。	判斷 (Judging) / 認知 (Perceiving)

在選定人格特質維度後，本研究以Myers-Briggs Type Indicator (MBTI) 量表來衡量編程人員之人格特質。Bradley與Hebert (1997) 利用MBTI量表針對兩個專案團隊進行分析，研究結果顯示比較複雜的專案需要人格特質組合較為平均的團隊，也就是說該團隊中必須要有各種不同之人格特質組合，這個研究指引了組織專案團隊的新方向。Gorla 與Lam (2004) 延伸上一個研究，同樣地利用MBTI人格特質指標來找出不同工作職位之需求，研究結果亦顯示出不同工作角色需要不同之人格特質配置。然而人格特質究竟會不會影響編程效率卻眾說紛紜，本研究認為之所以會產生研究結果不一致的主因在於過去研究忽略其他可能影響編程效率之因素（如工作型態），因此僅探討人格特質對編程效率之影響並不充分。除此之外，過去以雙人編程為基礎，探討編程人員特質配對的相關研究大都以學生為研究對象（McDowell et al. 2002; Williams et al. 2000），而雙人編程所呈現之效益是否得以沿用至軟體產業尚缺乏有關研究，特別是在本國中小企業高度密集環境下，相關探討更顯得迫切。

三、工作型態

為了要探究雙人編程之效益能夠在何時（how）以及如何（how）發生，工作型態相關議題勢必得一並探討。雙人編程所強調的不單只是編程效率，其亦強調專案執行時編程夥伴之間的意見共享與討論。Molokken與Jorgensen (2003) 更指出相較於獨自一人工作，群體工作之效率可能來得更低，主要原因在於獨自一人工作不需要考慮到其他成員的想法、人格特質以及工作習慣等，因此工作型態之迥異將帶來不同的工作成效。一般而言，工作型態可以用若干屬性來區分，包括工作範疇、工作複雜度、工作期間等（Sjoberg et al. 2002）。例如Williams et al. (2000) 在其研究中指出，若是以集思廣益為出發點，高複雜度工作較適合以雙人配對來進行。因此雙人編程之效益必須以工作型態為依歸，畢竟某些複雜度低的工作較適合單人獨力完成（Gallis et al. 2002）。有鑑於此，本研究依據上述概念將編程工作型態予以區分為工作內容（task content）與工作複雜度（task complexity）兩大致類別，如下說明：

（一）工作內容

不同的系統使用者，對於同樣一個系統有著不同的關注焦點。例如，專案經理可能較為重視系統運作時之品質，而編程人員較為在意系統日後的可維護性，至於終端使用者則較為關心系統之易用性。若以系統與使用者相關性為考量，則系統可以概略區分為內部元件與外部元件。內部元件係指使用者不會察覺該元件之運作過程，例如程式設計中經常論及之程式結合性（coupling）。外部元件是指直接與使用者需求相關，且使用者得以直接觸及之系統部分，例如系統操作介面、資料庫查詢功能等。然而不論是內部元件或是外部元件，都是經過編程人員之手所設計出來的，唯一最大之差異在於設計該元件時是否需要考量到使用者需求，因此本研究將工作內容分為內部內容（internal content）與外部內容（external content），內部內容係指使用者無需理解系統內部之邏輯流（logic flow）運作，而外部內容則是指編程人員於設計系統時必須考量到使用者之需求。

(二) 工作複雜度

Payne (1976) 指出工作複雜度可被視為找尋問題解決方案時所需負荷之認知負擔程度，該學者亦針對工作複雜度與編程績效之間關聯性提出探討。研究結果發現，複雜度越高的工作會導致較長的完成時間。由於工作複雜度為一抽象概念，當研究進行時有必要予以量化。本研究選擇以cyclomatic complexity方法來計算工作複雜度，主要原因在於cyclomatic complexity 能夠簡潔地表達程式結構所含之路徑數量，最受到軟體品質管理廣泛地使用 (McCabe 1996)。本研究以該學者所建議之10為複雜度門檻 (cyclomatic complexity = 10)，不論是內部工作內容或是外部工作內容，只要其cyclomatic複雜度大於11以上，即將之界定為高複雜度工作。反之，若工作內容複雜度小於等於10，則歸類為低複雜度工作。

四、研究假說

根據上述文獻探討可以得知，編程績效很有可能受到不同的人格特質、工作內容、以及工作複雜度所影響。因此本研究以社會互動（內向、外向）、資訊收集（感知、直覺）做為人格特質衡量維度，搭配複雜度高、低之內、外部工作內容，交叉配對出24種所有可能組合（6×4），如下圖1所示。建立上述24種組合之後，本研究得以藉由下列研究假說來檢定這些組合對編程績效之影響：

- 假設1. 不同工作型態下，不同人格特質之雙人配對在編程績效有不同之差異。
假設2. 不同工作型態對於編程績效有著不同之差異。
假設3. 不同人格特質對於編程績效有著不同之差異。

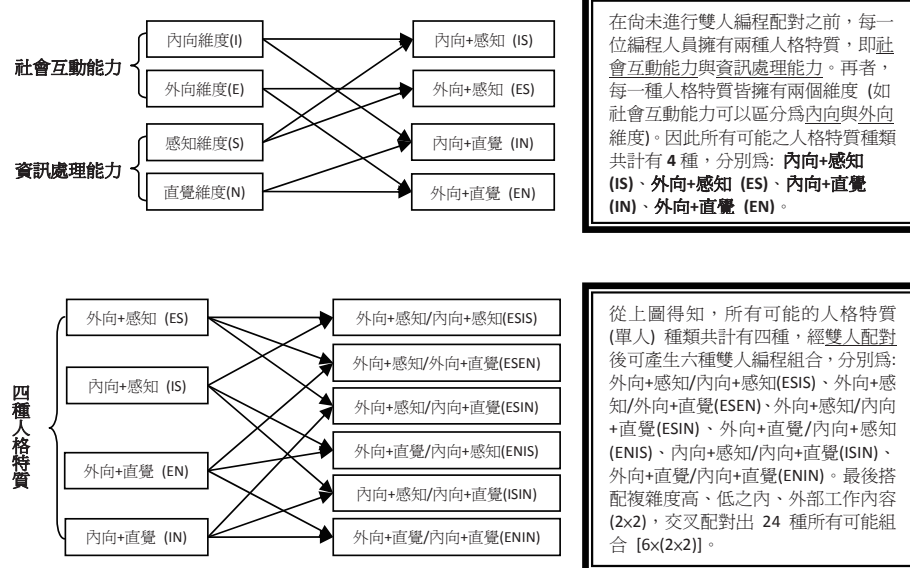


圖1：研究自變數配對組合

參、實驗設計

一、實驗專案選擇

雖然實驗室研究可以透過控制的手法將研究無關之干擾來源予以排除，但過多的控制可能會導致研究結果只能對應到類似的研究設定，無法推論至實驗室以外之情境。再者，由於本研究對象為中小型軟體產業之編程人員，我們很難要求受試者停止公司所指派之工作，轉而配合本研究的進行。在此情況下，實驗設計應以最貼近事實為前提來進行資料收集。實地實驗（field experiment）正好符合本研究訴求（Zikmund 2002），其精髓在於以最少的控制來分析實際生活中發生的問題，藉以提高實驗之外部效度。因此為了收集具有代表性之樣本進行分析，本研究以EMBA學生所任職之國內各大企業為對象，並透過下列原則進行專案篩選：

1. 專案規模：建議使用中、小型專案，若專案規模太大造成時程過長，容易發生前期收集資料因環境條件改變而無法與後期資料進行比較。
2. 編程人員數量：必須大於4人，若編程人員少於4人將無法隨時進行不同的配對，同時不同配對之間的差異將無法顯現。
3. 編程人員技術：編程人員必須是從未參與雙人編程開發環境，此外編程人員現有之技術，必須可以達到專案需求（如具備SCJP證照）。
4. 其他的特性：參與實驗人員包括專案管理人員、系統使用者以及編程人員必須有通暢的溝通管道，例如，聚集在一起工作或是能透過任何電子媒介進行溝通。

本研究依此原則經過訪談，尋找適合進行研究公司。最後挑選提供軟體開發服務公司（A公司）做為實驗標的。

二、公司背景

A公司成立於1994年，目前公司員工大約有50位，主要業務為提供專業軟體開發及系統整合服務。其客戶層大多數為台灣電子產業之知名大廠；近年來網際網路興起，該公司致力於提供企業E化服務，目前為客戶導入之系統大多數為電子網站（Web-Enabled）系統。本研究挑選A公司為實驗標的之原因如下：

1. 該公司擁有許多專案實作經驗，其編程人員技術能力具一定水準，導入雙人編程環境，夥伴間技術能力差距之影響可望減少。
2. 該公司在實作專案的經驗中，已發展出許多標準規範，例如，程式開發標準（Coding Standard）、統一的架構（Web enabled），標準而彈性的程式框架（Model-View-Control Framework）等。因此可以減少不同專案轉換過程中，所需要的技術探索時間。
3. 該公司目前以專案方式為其他公司開發各種網站系統，其專案進行方式主要為傳統軟體開發流程（瀑布式—Waterfall），為了確保其軟體品質及專案內容符合客戶之需求，必須投入大量的人力在於軟體測試；以往的經驗中可以發現，測試人

員與編程人員大約必須投入相同之精力。導入XP將可有效縮短其開發時程，提高軟體品質，所以該公司對雙人編程環境之建置有相當大的期望，願意全力配合進行實驗。

(一) 專案背景

從A公司現有的專案中，我們挑選其中兩個專案做為本研究實驗對象（專案A及專案B），這兩個專案之特性有許多共通性（如表2所示），因此在專案進行過程中，兩個專案之編程人員可互相支援，不需要因為切換專案，重新適應不同的開發方式，同時也適合將所收集之資料一起進行比對。其共通性內容 述如下：

1. 系統架構：此二個專案在系統架構上，皆採用三層式架構（3-Tiers Architecture，如圖2所示）。

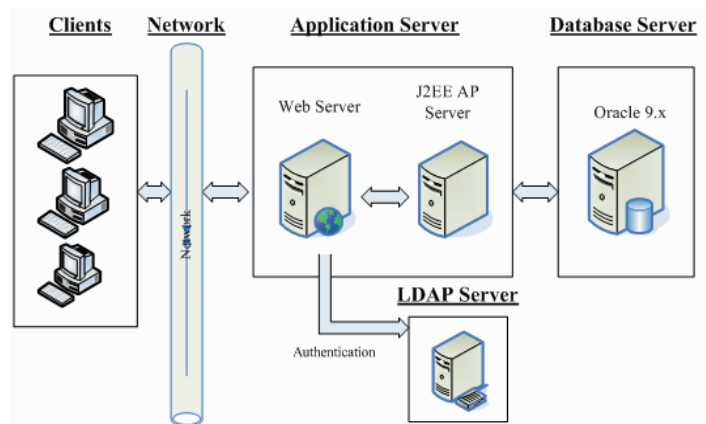


圖2：專案系統架構圖

2. 程式架構：採用標準Java語言，應用標準J2EE API，搭配Struts實作MVC（Model-View-Contral Framework），如圖3所示。

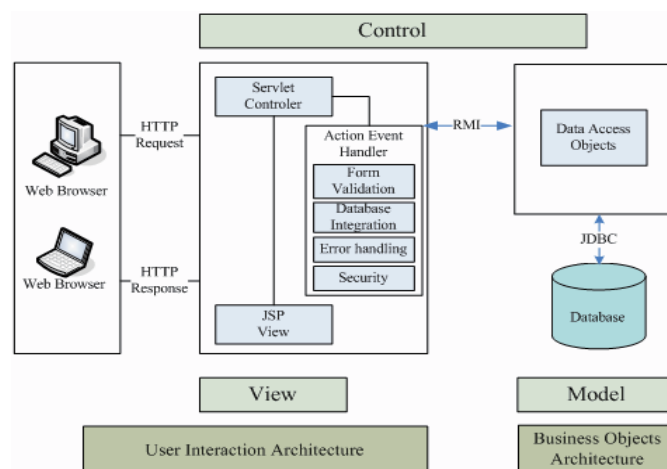


圖3：程式架構圖

3. 指派「教練」：每一個專案配置一名資深技術經理，負責指導專案所需之技術，並且確認編程人員遵循程式標準，控制程式品質。
4. 雙人編程環境：專案進行之前，皆已對所有參與人員進行教育訓練，分配每個人工作職掌（如圖4所示），並且確認合作模式。

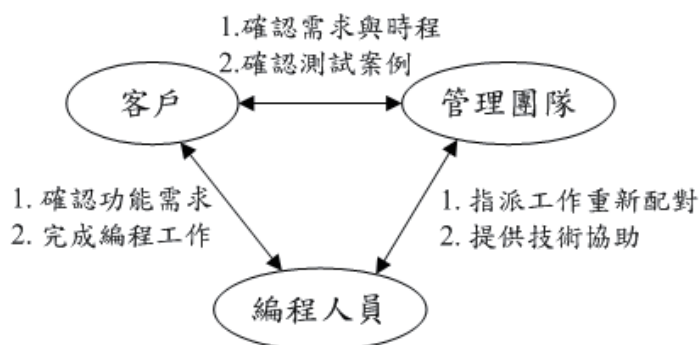


圖4：雙人編程環境組織圖

5. 專案開始之前，由專案經理與技術教練，將專案功能切割至適合的大小，在此稱為程式單元。並且對每一個單元進行工作內容及工作複雜度的評量，以利實行雙人編程環境。工作內容評估由專案經理及技術經理一起進行評估，若某功能需要與外部使用者互動則將其歸類為外部功能，若此功能屬於系統內部維護功能則歸類為內部功能。工作複雜度則由技術經理與客戶一起制定，對每一項功能，依客戶的要求，撰寫所有的測試案例，若單一功能測試案例個數大於10個則將此功能之工作複雜度歸類為複雜度高，若小於等於10個則將此功能複雜度歸類為複雜度低。
6. 與客戶溝通模式：專案進行時，將派駐人員至客戶端進行編程工作，所以當功能有疑問時，可隨時面對面或是內線電話進行確認，縮短時間成本。
7. 教練溝通模式：教練隨編程人員至客戶端進行支援，因此編程人員可得到應有支援。

表2：專業背景說明

	專案A	專案B
簡介	專案A為該公司對外客戶服務系統，主要使用者為向該公司下單之客戶，透過網站提供其客戶取得重要的訂單及工作製程資料。	專案B為內部製程檢核系統，主要使用者為內部工程師，透過系統試圖將不同的部門訊息得以互通。
內容	透過彈性的權限管控機制，提供客戶所需的資料。	系統主要功能為流程管控，需串接不同部門及功能。
特性	權限管控系統相當複雜，但功能與功能之間彼此獨立，不需要與其他功能互動。	系統各功能之間必須互相搭配，所以需要由教練主導配合連接測試。

(二) 編程人員配對

經詢問實驗對象公司之編程人員後得知，共有16位編程人員願意參與本研究。在進行實驗之前本研究以MBTI人格特質問卷，針對編程人員進行性向調查，結果分佈如表3所示，其中11位編程人員隸屬於專案A，其餘5位編程則屬於專案B。

表3：編程人員人格特質分佈

		專案A	專案B
個人性向	外向-感知 (Extravert-Sensing, ES)	4	2
	外向-直覺 (Extravert-Intuition, EN)	2	1
	內向-感知 (Introvert-Sensing, IS)	4	1
	內向-直覺 (Introvert-Intuition, IN)	1	1

在確定具有參與願意之編程人員性向後，隨即依照研究所需將這些人員進行配對。總配對次數為158次，但其中若干配對組合因實驗條件發生變化而無法將之列入分析，篩選條件如下：

1. 需求變更：當編程人員開始進行工作之後，顧客需求變更幅度過大，與事先評估之範圍產生落差，因此無法估算出編程人員績效。
2. 技能不足：某些功能在編程過程中，編程人員無法獨立完成，必須由其他人員協助，因此不能將其資料列入分析條件。
3. 個人因素：配對過程中，若編程人員因個人因素，無法全程參與，例如：請假，或是臨時支援別人，則時間紀錄則不精確。

最後成功收集可供分析之資料總數為128組，其案例分佈如表4所示。

表4：實驗案例分佈

工作複雜度		低		高	
工作內容		內部	外部	內部	外部
人格特質	ES-EN	5	5	4	7
	ES-IS	4	7	4	6
	ES-IN	5	6	4	7
	EN-IS	4	5	6	6
	EN-IN	5	5	5	7
	IS-IN	5	6	4	6
合計(組)		28	34	27	39

從表3與表4中可以得知，這樣的配對方式將導致某些編程人員參與一個以上之案例，進而使得內部效度稍嫌不足。然而如同過去雙人編程研究一般 (Nosek 1998; Gorla & Lam 2004)，我們亦遭受到樣本數匱乏之困擾。由於研究對象為中小型軟體產業，相對

於以大型企業為實驗對象，員工人數不足為先天限制，因此本研究採取實地實驗（field experiment）試圖以外部效度之優勢來彌補內部效度不足之缺失。此外，為了避免過度配對所造成之編程人員身心負擔，進而影響編程績效準確性，在實驗結束後本研究額外實施滿意度問卷調查，藉由編程人員參與雙人編程實驗之滿意程度來做為編程績效變化之輔證。

三、分析方法

本研究擬以變異數分析來計算不同人格特質與不同工作型，在雙人編程配對下對於編程績效之影響。其中人格特質為具有六種處理之自變數，工作型態則為具有四個處理之自變數（ 6×4 ）。如上所述，在樣本數不足以及配對所需之情況下，我們無法以隨機的方式將受試者隨機指派至24個實驗處理之中，故本研究採類似準實驗設計之概念，以非隨機過程建立受試者配對組合（即由受試者自行最決定是否接受過度配對）。然而非隨機過程可能亦會造成內部信度不足的問題，因此在依變數衡量方面，除了原先所提到的編程績效（ y_1 ）之外，本研究額外加入另一依變數「第一天完成比例（ y_2 ）」，檢視方式則是以XP實行原則中的測試案例（test case）來進行（Jeffries et al. 2000; Zhang 2004）。所謂編程績效（ y_1 ）係指『通過所有測試案例所需時間』，而第一天完成比例（ y_2 ）則是指『於第一天工作結束後檢視所有組合所完成測試案例之百分比』。本研究相信，若經實驗處理後自變數確實對依變數產生一定程度影響時，則此兩依變數（ y_1 & y_2 ）理當呈現負向關係。也就是說不論是採用第一天完成比例或是編程績效為依變數，其結果之差異應該不大，藉以確認實驗數據之合理性與有效性。

在變異數分析方法方面，係以二因子變異數分析（two-way ANOVA）進行而非多變量變異數分析（MANOVA），主要理由如下：

- (1) 在形式上雖然本研究提及兩種依變數（ y_1 & y_2 ），然而第一天完成比例（ y_2 ）並非分析重點，其主要用途僅在於呈現與編程績效（ y_1 ）之間的負向關聯性。因此在多個自變數與單一依變數之實驗情境下，宜採二因子變異數分析而非多變量變異數分析。
- (2) 兩依變數彼此之間具備「一定程度相關性」為多變量變異數分析之前提假設，然而過高的相關性可能導致共線性（Multicollinearity）的問題發生，一旦依變數之間存在著高度的共線性，研究者將難以釐清實驗處理之影響效果該歸屬於哪一個依變數（Hair et al. 2006, p. 410）。由於本研究假設編程績效（ y_1 ）與第一天完成比例（ y_2 ）彼此之間應當存在著負向相關，且相關程度越大越能表示實驗數據之合理性，因此不適合以MANOVA進行分析。
- (3) 變異數分析中包含重複量數分析（repeated measure），可用來觀察相同的受試者在前後兩次實驗中對依變數之影響，亦可稱為隨機化區組設計。然而本研究係以非隨機方式指派受試者，再加上研究目的並非探討編程績效（ y_1 ）與第一天完成比例（ y_2 ）此兩依變數實驗多次之差異，因此不符合重複量數分析中受試者組內設計之精神。

肆、資料分析

一、第一天完成比例

根據二因子變異數分析之流程得知，在進行因子主要效果分析之前，宜先行確認因子間交互作用是否顯著，如果交互作用顯著則無須探討主要效果。反之，如果因子間交互作用不顯著時，則可進行主效果分析，即比較邊緣平均數。因此，本研究以二因子變異數分析，先行了解不同工作型態下與不同人格特質是否存在顯著地交互作用，如表5所示。分析結果發現，不同工作型態下與不同人格特質之雙人配對在編程績效上並無顯著差異存在（ $F=0.835$, $P=0.637$ ）。然而在主效果方面，不同工作型態、不同人格特質，各自在編程績效上存在著顯著之差異（ $F=2.312$, $P=0.08$; $F=3.070$, $P=0.01$ ），因此可進行後續之單因子變異數分析。

表5：二因子變異數分析之交互作用

變異來源	型III平方和	自由度	平均平方和	F檢定
截距	79.39	1	79.39	1951.70***
工作型態	0.282	3	0.094	2.312*
人格特質	0.625	5	0.125	3.070**
工作×人格	0.510	15	0.034	0.835 ^{N.S.}
誤差	4.231	104	0.041	
校正後總數	5.599	127		

N.S. 未達顯著 * $P<0.1$ ** $P<0.05$ *** $P<0.01$

在工作型態方面，變異同質性檢定未達顯著效果（如表6所示），表示各組間變異數沒有顯著之差異，符合變異數分析之假設，得繼續進行主效果分析。分析結果發現，不同工作型態對於編程績效具有顯著之差異（ $F=2.169$, $P=0.095$ ）。

表6：單因子變異數分析之主效果（工作型態）

變異來源	型III平方和	自由度	平均平方和	F檢定	變異同質
截距	79.39	1	79.39	1850.83***	$F=0.605$
工作型態	0.279	3	0.093	2.169*	$df_1=3$
誤差	5.319	124	0.043		$df_1=124$
校正後總數	5.599	127			$p=0.613$ ^{N.S.}

N.S. 未達顯著 * $P<0.1$ *** $P<0.01$

在人格特質方面，變異同質性檢定未達顯著效果（如表7所示），表示各組間變異數沒有顯著之差異，亦符合變異數分析之假設，得繼續進行主效果分析。分析結果發現，不同人格特質對於編程績效具有顯著之差異（ $F=2.913$, $P=0.016$ ）。

表7：單因子變異數分析之主效果（人格特質）

變異來源	型III平方和	自由度	平均平方和	F檢定	變異同質
截距	81.95	1	81.95	1998.93**	F=1.641
工作型態	0.597	5	0.119	2.913**	df ₁ =5
誤差	5.001	122	0.041		df ₁ =122
校正後總數	5.599	127			p=0.154 ^{N.S.}

N.S. 未達顯著 **P<0.05

接著依照不同類別，針對第一天完成比例之平均數進行比較，分析結果如表8所示。從平均數（91.34%）可以看出，ES-IS的組合明顯高於其他組合，表示ES-IS的編程績效優於其他的組合。從這個分析中也可以發現，ES-EN的組合表現（67.57%）明顯較其他組別差，但若把ES或EN與其他人格特質搭配組合時，則其編程績效將會提高。另外我們也分析了不同工作內容及工作複雜度的標準差，可以發現工作內容屬於外部介面功能的研究變項中，代表不同人格特質組合表現的比較穩定。而工作內容屬於內部介面功能，不同人格特質組合其編程績效差異相對較大。我們再次強調，到目前為之所進行之分析僅針對「第一天完成比例」，用以確認是否與編程績效呈現負向相關性，而此並非本研究主要目的，畢竟專案經理較為在乎整體專案是否能如期完成。因此，即使第一天完成比例很高，但若干組別未能於專案結束前完成所指派之工作也將無濟於事。有鑑於此，變異數分析之事後比較僅針對「編程績效 y_1 」進行分析。

表8：第一天完成比例平均值

第一天完成比例（%）					
	低複雜/內部	低複雜/外部	高複雜/內部	高複雜/外部	平均數
ES-EN	64.80%	86.40%	55.50%	63.57%	67.57%
ES-IS	98.25%	84.86%	94.00%	88.25%	91.34%
ES-IN	77.40%	83.80%	75.50%	79.27%	78.99%
EN-IS	100.00%	83.83%	65.73%	74.93%	81.12%
EN-IN	95.60%	82.17%	75.78%	73.80%	81.84%
IS-IN	81.60%	83.57%	87.40%	73.17%	81.43%
標準差	0.14	0.01	0.14	0.08	

二、假說檢定

從編程績效之二因子變異數分析得知（如表9所示），不同工作型態下與不同人格特質之雙人配對在編程績效上並無顯著差異存在（ $F=0.835$, $P=0.637$ ），因此假設1不成立。然而在主效果方面，不同工作型態、不同人格特質，其各自在編程績效上有著之差異（ $F=2.526$, $P=0.062$; $F=2.137$, $P=0.067$ ），可進行後續之單因子變異數分析。

表9：二因子變異數分析之交互作用

假說1. 在不同工作型態下，不同人格特質之雙人配對在編程績效上有著不同之差異（不成立）				
變異來源	型III平方和	自由度	平均平方和	F檢定
截距	13458.12	1	13458.12	595.08***
工作型態	171.378	3	57.129	2.526*
人格特質	241.696	5	48.339	2.137*
工作×人格	237.489	15	15.833	0.779 ^{N.S.}
誤差	2352.05	104	22.616	
校正後總數	2975.39	127		

N.S. 未達顯著 *P<0.1 ***P<0.01

在工作型態方面，變異同質性檢定未達顯著效果（如表10所示），表示各組間變異數沒有顯著之差異，符合變異數分析之假設，得繼續進行主效果分析。分析結果發現，不同工作型態對於編程績效具有顯著之差異（ $F=2.169, P=0.069$ ），因此假設2成立。

表10：單因子變異數分析之主效果（工作型態）

假說2. 不同工作型態對於編程績效有著不同之差異（成立）					
變異來源	型III平方和	自由度	平均平方和	F檢定	變異同質性
截距	13798.28	1	13798.28	608.78***	F=0.478
工作型態	164.858	3	54.953	2.425*	df ₁ =3
誤差	2810.53	124	22.666		df ₁ =124
校正後總數	2975.39	127			p=0.698 ^{N.S.}

N.S. 未達顯著 *P<0.1 ***P<0.01

在人格特質方面，變異同質性檢定同樣未達顯著效果（如表11所示），表示各組間變異數沒有顯著之差異，符合變異數分析之假設，得以繼續進行主效果分析。分析結果發現，不同工作型態對於編程績效具有顯著之差異（ $F=2.033, P=0.079$ ），因此假設3成立。

表11：單因子變異數分析之主效果（人格特質）

假說2. 不同工作型態對於編程績效有著不同之差異（成立）					
變異來源	型III平方和	自由度	平均平方和	F檢定	變異同質性
截距	14214.48	1	14214.48	631.40***	F=1.208
工作型態	228.842	5	45.768	2.033*	df ₁ =5
誤差	2746.55	122	22.513		df ₁ =122
校正後總數	2975.39	127			p=0.309 ^{N.S.}

N.S. 未達顯著 *P<0.1 ***P<0.01

接著針對總完成時數進行分析，分析結果如表12所示。從平均數可以看出，ES-IS的總完成時數（8.75）明顯比其他組合的總完成時數低，表示ES-IS的編程績效優於其他的

組合。從這個分析中也可以發現，ES-EN的表現（13.16）明顯比其他組合差，但若把ES或EN與其他人格特質搭配組合時，則其編程績效將會提高。另外我們也分析了不同工作內容及工作複雜度的標準差，可以發現工作內容屬於外部介面功能的研究變項中，不同人格特質組合表現的比較穩定。而工作內容屬於內部介面功能中，不同人格特質組合其編程績效差異較大。

表12：總完成時數平均值

總完成時數（小時）					
	低複雜/內部	低複雜/外部	高複雜/內部	高複雜/外部	平均數
ES-EN	13.60	9.40	15.63	14.00	13.16
ES-IS	6.25	9.93	8.75	10.08	8.75
ES-IN	11.40	8.60	11.25	11.50	10.69
EN-IS	5.50	10.17	13.50	11.50	10.17
EN-IN	6.70	10.25	11.00	11.57	9.88
IS-IN	10.60	9.29	9.13	12.00	10.25
標準差	3.30	0.63	2.63	1.27	

綜合上述結果進行相關分析，結果發現總完成時間與第一天完成的比例有高度的負相關（-0.96），從這樣的結果可以確認實驗數據之合理性，也就是說不論採用第一天完成比例或是總完成時數做為分析標的，其結果之差異應該不大。因此配合研究目的所需，我們以總完成時數作為後續之探討。

（一）事後檢定分析

由於實驗中每一個處理所指派之受試者人數不一，因此以Scheffe法作為事後比較基礎，分析結果如下：

1. 工作複雜度高及外部工作內容：

由表12分析結果可以得知，相對於其他組別，ES-EN需耗費較長的時間來完成高複雜度/外部工作，且經由事後檢定可以發現ES-EN與ES-IS存在顯著的差異（ $p=0.006$ ），與其他組別則沒有顯著差異（請參閱附表1）。

2. 工作複雜度高及內部工作內容：

由表12分析結果可以發現，ES-EN編程時效比其他的組合差，相對於其他組別，ES-EN需耗費較長的時間來完成高複雜度/內部工作，且經由事後檢定可以發現ES-EN與ES-IS、ES-IN皆存在顯著的差異（ $p=0.047$, $p=0.081$ ），而與其他組別則沒有顯著差異（請參閱附表2）。

3. 工作複雜度低及外部工作內容：

根據表12分析結果可以得知，在工作複雜度低及外部工作內容的情況下，各種人格特質組合所耗費之總完成時間並沒有顯著差異（請參閱附表3）。

4. 工作複雜度低及內部工作內容：

此部分針對工作複雜度低以及工作內容屬於內部介面功能的變項進行分析，由表12得知，ES-EN編程績效比其他的組合差，且經由事後檢定可以發現ES-EN與ES-IS存在顯著的差異（ $p=0.022$ ），與EN-IS存在顯著的差異（ $p=0.017$ ），與EN-IN存在顯著的差異（ $p=0.018$ ），但與其他組別則沒有顯著差異（請參閱附表4）。

（二）個人工作滿意度分析

軟體專案成功與否除了成本效益之考量外，仍須注意編程人員的工作滿意度（Williams et al. 2000）。主要原因為編程人員係屬公司資產，若過於專注成本效益而忽略其工作情緒，將有可能導致員工離職、團隊成員相處不融洽等問題。以公司永續經營的角度而言，喪失這些無形資產所衍生之成本將難以想像。有鑑於此，除了衡量編程績效之外，本研究另外測量編程人員工作滿意度。我們參照過去文獻（McDowell et al. 2006）以李克特七點尺度分別衡量此二問項，即「您對於雙人配對之工作方式是否感到滿意？」以及「若未來有機會，您是否仍願意參與雙人編程工作？」。分析結果顯示，有69%的編程人員對於在雙人編程之工作環境感到滿意。值得注意的是，編程人員從事編程工作之年資並不相同，因此不同經驗資歷也可能對編程績效造成影響，故以其年資分為兩類，分別為資淺編程人員11人及資深編程人員5人；資深編程人員是指曾經參與三個專案以上或是工作年資滿一年且有能力獨立完成編程工作，分析結果如圖5所示。從結果中可以發現，資深編程人員工作滿意度會小於資淺編程人員。可能的原因在於資深人員已具備程式撰寫之熟稔經驗，所以雙人編程對他們所帶來之效益相對較少，然而對於資淺編程人員而言，本研究認為雙人編程有助於增加其對專案執行所需之相關經驗與知識。

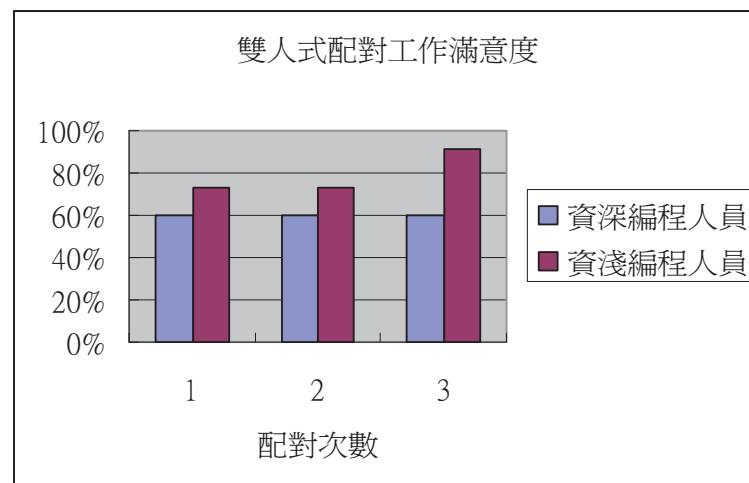


圖5：雙人編程工作滿意度

伍、結論與建議

一、研究結論

本研究延續前人之研究，利用MBTI量表測量編程人員人格特質，根據Gorla和Lam (2004)的建議，以社會互動能力（Social interaction）、資訊收集能力（Information gathering）以及工作型態（Task type）為組合條件，進行雙人編程環境之建立；研究結果顯示：

（一）複雜度低/內部工作（Low/Internal）

在工作複雜度低且內部工作的情境下，表現最佳之組別依次為EN-IS（5.50）、ES-IS（6.25）、EN-IN（6.70），而表現最差的組別則是ES-EN（13.60），其餘組別則沒有顯著之差異。探究可能原因為具備內向-感知（IS）編程人員，在工作複雜度低且內部工作的情況下能夠充分發揮其特質與互補作用。由於內向-感知（Introvert-Sensing）人員較不擅長與人互動，而且除非所遇到之問題已經有現成的解決方案，否則較不喜歡處理前所未見之新問題，如此才得以在複雜度低與內部例行編程工作上發揮其特殊人格特質。因此除了不同社會互動能力之配對效果，若能夠再加上具有感知特質的人員，特別能夠在複雜度低且不需與顧客互動之內部工作上有良好的表現。另一方面，由於ES-EN兩人皆具有外向人格特質，而且在資訊收集方面其中一人為直覺式（intuition）思考者，這樣的配對並非適合工作複雜度低、內部工作。探究可能之原因在於，外向且直覺式思考之編程人員不論是在行為或思緒方面皆較為活潑且彈性，一旦遭遇枯燥乏味的例行性內部工作時較無法保有編程之熱情。

（二）複雜度低/外部工作（Low/External）

透過事後檢定分析可以得知，在此工作類型下，人格特質彼此之間沒有顯著的差異。但仍可從其平均數中比較編程績效之優劣。相較於其他類別之工作，此類工作性質較為例行且穩定，但卻需要與使用者溝通其需求。如同上述所示，相同人格特質配對所表現出之績效仍然比其他交叉配對組還來得差（如ES-EN vs. ES-IN）。ES-IN與IS-IN為表現最為優良之組合（8.60 & 9.29），這表示外向-感知（Extravert-Sensing）人員特別適合被指派處理複雜度低/外部性質的工作，主要的原因在於ES富有善於溝通、實事求是之人格特質，此特質恰好可與IN人員互補。但值得注意的是，IS-IN組合屬相同人格特質配對但卻獲得較佳的編程績效。探究可能之原因在於，此類型工作複雜度較低，恰好與其內向-感知（Introvert-Sensing）適配，但卻礙於需要與顧客溝通需求，因此其編程績效屬第二優良之組別，無法超越ES-IN組。

（三）複雜度高/內部工作（High/Internal）

在此類別中，表現最為突出的是ES-IS（8.75），其次為IS-IN（9.13），而表現最差的組別仍是ES-EN（15.63）。如上所述，不同配對之社會互動能力組別的績效仍比相同配對還來得高。然而在此類別中卻出現例外情況，即IS-IN績效表現為次優。探究可能之

原因在於內向-直覺（Introvert-Intuition）人員除了不擅長與外界事物互動之外，直覺般特質使他們樂於接受較為困難的挑戰，因此複雜度高/內部工作性質，特別適用在IS-IN組合上。雖然ES-IS的表現優於IS-IN（8.75 vs. 9.13），但在事後比較分析上卻無顯著之差異。

（四）複雜度高/外部工作（High/External）

相較於上述三個類別，此工作類別除了複雜度高之外，還需要與使用者保持良好之互動，因此總耗費時間會比其他類別還來得高。由於ES-EN與IS-IN（14.00 & 12.00）皆為相同配對之社會互動能力，因此表現較其他四組還來得低。表現最好的組別為ES-IS（10.08），其次為ES-IN與EN-IS（11.50 & 11.50）。外向-直覺（Extravert-Intuition）人員特別適合被指派到這樣的情境下，主要的原因在於他們富有良好表達之能力與原創力，在配對環境中得以充份發揮其獨有之人格特質。雖然ES-IS表現略高於EN-IS，但從兩者事後分析比較得以發現並沒有顯著差異。

（五）最差組合/最佳組合（ES-EN/ES-IS）

由表12裡的總平均數可以得知，表現最佳與最差的組別分別為ES-IS (8.75)和ES-EN (13.16)。從研究結果究發現，不論工作複雜度或工作內容為何，此兩類別始終呈現出六大人格特質裡之最佳與最差組別，推斷可能之原因如下（由於Low/External在事後分析中沒有發現顯著之差異，因此僅就其他三個工作型態分別說明）：

- （1）以複雜度低/內部工作（Low/Internal）而言，由於外向編程人員較喜歡接觸外在世界的人、事、物，因此當兩人皆為外向人格特質時，一旦面對複雜度極低且與系統相關之例行性工作時，反而容易表現出較差的編程績效。相反地，若兩人為不同之人格特質配對且資訊收集能力剛好符合此複雜度低之內部工作時，其本人實事求是的精神再加上內外向人格特質之互補效果，將可以有效地提升編程績效。
- （2）以複雜度高/內部工作（High/Internal）為例，雖說同樣是外向特質的兩編程人員應該對於複雜度高的工作較為擅長，但其中一人在資訊收集能力方面卻是以直覺式思考，不適合從事內部系統相關之例行性工作。此時，ES-IS組合會有較佳的編程績效，原因在於除了其具有內外向特質之互補優勢，在資訊收集能力方面兩人皆是感知思考，因此在面對例行性工作更能得心應手。
- （3）以複雜度高/外部工作（High/External）來說，如同上述一般，同樣是外向特質的兩編程人員應該對於複雜度高的工作較為擅長，但其中一人在資訊收集能力方面卻是以感知式思考，一板一眼的處事風格不適合從事外部與顧客需求相關之工作，畢竟顧客需求總是變化萬千，難以掌握。此時ES-IS組合會有較佳的編程績效，原因在於除了其具有內外向特質之互補優勢，但其在資訊收集能力方面皆是感知式思考，因此相對於其他兩種工作型態，其編程績效亦呈現遞減狀態。

二、雙人編程組合之實務上建議

綜合上述結論，若兩人皆為相同社會互動能力配對（或相同資訊收集能力）時，面對不同的工作內容及工作複雜度，其編程效能明顯偏低。但若將其配對成不同社會互動能力（或不同資訊收集能力）時，其編程效能就會明顯的提高。此項結論同時也符合Gorla和Lam (2004)所建議，即軟體開發團隊的人格特質越平衡，則其效能越高。此外，資淺編程人員參與雙人編程環境之後，其工作滿意度明顯會有提昇，而資深編程人員則沒有明顯的提昇。這樣的結果表示，資淺編程人員經由雙人編程訓練之後，不僅能加速其編程能力，且有助於提昇對自我肯定；而資深編程人員也可以藉由這樣的訓練，加速經驗及技術的傳承。所以雙人編程環境的建置，可以成為公司內部人員在職訓練的良好選擇。最後本研究整理出下列建議，以作為建置雙人編程環境實務上人員配對之建議。

1. 雙人編程環境之效益：實施雙人編程有助提昇專案成員之編程績效，同時亦能提高個人之工作滿意度。
2. 人格特質之影響：雙人編程中之人格特質配對，確實會影響編程績效；從社會互動能力（Social interaction）來看，外向和內向之搭配編程效能較高，而二個都是內向的搭配則居次，最差的搭配則是兩者都是外向。從資訊收集能力（Information gathering）的配對來看，本研究並沒有發現太大之差異，可能的原因為本研究雙人環境之建置為面對面溝通，資訊尚屬充足。然而若所建置之環境係透過IT來進行雙人溝通，互動效果可能不及面對面方式，因此仍然必須要考慮此能力之配對。
3. 工作複雜度之影響：由於不同人格特質所擅長的工作複雜程度會有差異，因此人員的配對與指派需搭配不同之工作複雜度，意即投其所好。
4. 工作內容之影響：假設工作內容必須與使用者進行溝通時，則配對之中必須要有一位編程人員是屬於外向的，才能得到比較好的績效。若工作內容屬於系統內部維護工作，則人格特質之影響較小，但在任何一個配對之中，至少必須要有一個內向之編程人員，才能維持一定編程績效。

三、研究限制

本研究係以專業軟體開發公司A公司為例，探討雙人編程環境下，針對不同工作內容及工作複雜度，應如何進行編程人員之組合搭配，期望能提高編程績效。由於現今台灣的軟體產業中，雙人編程環境之建置，仍然是一個新興的應用，尚有許多條件及觀念上的不成熟，因而直接造成取樣上的限制。為了消除此項主要的限制，惟透過先期教育訓練進行觀念溝通，將可增加分析案例之信度與效度。此外，有關於「資訊系統開發」成功議題方面，本研究論及之編程績效依變數係以效率層面為出發點，並非效能層面。也就是說編程績效僅能夠表示編程人員於特定時間內完成專案所要求之進度，至於整體專案執行是否成功則屬效能層面之議題，為專案經理職責（Sheth & Sisodia 2002）。因此編程績效可被視為資訊系統開發成功之必要但非充分條件，成功之關鍵仍

與專案經理決策有關。

四、後續研究建議

本研究於評估雙人編程環境下之編程績效，已初步取得成效。但由於樣本數之不足，實無法取得全面性之建議，故對後續之研究有下列之建議：

本研究為單一公司之研究，後續可以本研究作為基礎，選擇多個相同甚至不同背景或企業文化之個案，進行比較研究，例如以不同行業且屬於公司MIS部門自行開發之軟體為個案，與資訊委外公司加以比較；將對不同搭配之差異有更深切之認知。

本研究只針對編程工作進行探討，未來可以將其研究範圍擴大至系統分析或是系統設計人員之搭配組合，此結果將有可能改善軟體專案開發團隊整體績效。

由於影響編程績效之因素很多，本研究僅就選取之個案的特殊性，擇以工作內容及工作複雜度等變數，與人格特質搭配組合之關係加以探討，建議後續研究者，可依不同的研究個案，摘取適切的變項做更深入之研究。

建置雙人編程環境，必須有適切的工作分配為基礎，若沒有經過良好的規劃，例如：彈性的專案管理、良好的溝通模式…等，將可能會得到反效果。因此建議後續研究者，對於跨公司之間合作之專案，並不適合應用雙人編程環境，以減少溝通上問題發生。

參考文獻

1. 資策會資訊市場情報中心 (MIC) “台灣軟體產業現況與發展,” 2006 (available online at <http://mic.iii.org.tw/intelligence>).
2. Beck, K. “Embrace Change with Extreme Programming,” *IEEE Computer* (32:10), 1999, pp. 70-77
3. Bevan, J., Werner, L. and McDowell, C. *Guidelines for the User of Pair Programming in a Freshman Programming Class*, Conference on Software Engineering Education and Training, 2002.
4. Bradley, J.H. and Hebert, F.J. “The Effect of Personality Type on Team Performance,” *Journal of Management Development* (16:5), 1997, pp. 337-353
5. Cockburn, A. and Williams, L. *The Costs and Benefits of Pair Programming*, in Giancarlo Succi and Michele Marchesi (eds.) *Extreme Programming Examined*. Boston: Addison Wesley, 2001, pp. 223-243.
6. Constantine L.L. *Constantine on Peopleware*, Yourdon Press, Englewood Cliffs, N. J., 1995.
7. Coplien, J.O. *A Generative Development-Process Pattern Language*, in *Pattern Language of Program Design*, J.O. Coplien and D.C. Schmidt, Eds. Massachusetts, USA: Addison-

- Wesley, pp. 183-237, 1995.
8. DeMarco, T. and Lister, T. *Peopleware: Productive Projects and Teams*, 2ed. New York: Dorset House Publishing, 1999.
 9. Gallis, H., Arisholm, E. and Dyba, T. *A Transition from Partner Programming to Pair Programming: an Industrial Case Study*, Pair Programming Workshop in 17th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), 2002.
 10. Gehringer, E.F. *A Pair-Programming Experiment in a Non-Programming Course*, ACM Press, 2003, pp. 187-190.
 11. Gorla, N. and Lam, Y.W. "Who Should Work with Whom?: Building Effective Software Project Teams," *Communications of the ACM* (47:6), 2004, pp. 79-82
 12. Guinan, P.J., Coopridge, J.G. and Faraj, S. "Enabling Software Development Team Performance during Requirements. Definition: a Behavioral versus Technical Approach," *Information Systems Research* (9:2), 1998, pp. 101-125
 13. Hair, J.F., Black, B., Babin, B., Anderson, R.E. and Tatham, R.L. *Multivariate Data Analysis* (6th ed.), Prentice Hall, 2006.
 14. Hedin, G., Bendix, L. and Magnusson, B. "Teaching Extreme Programming to Large Groups of Students," *The Journal of Systems and Software* (74), 2005, pp. 133-146
 15. Hohmann, L. *Journey of the Software Professional: A Sociology of Software Development*, New Jersey, USA: Prentice-Hall, 1997.
 16. Howard, A. "Software Engineering Project Management," *Communications of the ACM* (44:5), 2001, pp. 23-24
 17. Jeffery, D.R. and Votta, L.G. "Guest Editor's Special Section Introduction," *IEEE Transactions on Software Engineering* (25:4), 1999, pp. 433-437
 18. Jeffries, R.E., Anderson, A. and Hendricson, C. *Extreme Programming Installed*, Addison Wesley Longman, 2000.
 19. Kircher, M., Jain, P., Corsaro, A. and Levine, D. *Distributed Extreme Programming*, Proceedings of XP2001 - eXtreme Programming and Flexible Processes in Software Engineering, Villasimius, Sardinia, Italy, May 21-23, 2001, pp. 66-71.
 20. McCabe, T. "Cyclomatic Complexity and the Year 2000," *IEEE Software* (13:3), 1996, pp. 115-117
 21. McDowell, C., Werner, L., Bullock, H. E. and Fernald, J. *The Effects of Pair Programming on Performance in an Introductory Programming Course*, Proceedings of the Thirty-Third Technical Symposium on Computer Science Education (SIGCSE 2002), ACM Press, 2002.
 22. McDowell, C., Werner, L., Bullock, H.E. and Fernald, J. "Pair Programming Improves Retention, Confidence," *Communication of the ACM* (49:8), 2006, pp. 90-95
 23. Molokken, K. and Jorgensen, M. *Software Effort Estimation: Unstructured Group Discussion as a Method to Reduce Individual Bias*, Proceedings of 15th Annual Workshop

- of the Psychology of Programming Interest Group, 2003.
24. Myers, I. B., McCauley, M. H., Quenk, N. L. and Hammer, A. L. *MBTI® Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator® (3rd ed.)*. Palo Alto, CA: CPP, Inc., 1998.
25. Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C. and Balik, S. *Improving the CSI Experience with Pair Programming*, Proceedings of the Thirty-Fourth Technical Symposium on Computer Science Education (SIGCSE 2003), ACM Press, 2003.
26. Nosek, J.T. "The Case for Collaborative Programming," *Communications of ACM* (41:3), 1998, pp. 105-108
27. Olson, J., Teasley, S., Covi, L. and Olson, G. *The (currently)Unique Advantages of Collocated Work*, In: Hinds, P., Kiesler, S. (Eds.), *Distributed Work*. MIT Press, Cambridge, MA, pp. 113-135, 2002.
28. Payne, J.W. "Task Complexity and Contingent Processing in Decision Making: an Information Search and Protocol Analysis," *Organizational Behavior and Human Performance* (16), 1976, pp. 366-387
29. Rasch, R.H. and Tosi, H.L. "Factors Affecting Software Developers Performance: An Integrated Approach," *MIS Quarterly* (16:3), 1992, pp. 395-413
30. Schneider, J.G., and Johnston, L. "eXtreme Programming – Helpful or Harmful in Educating Undergraduates?," *The Journal of Systems and Software* (74), 2005, pp. 121-132
31. Sheth, J.N. and Sisodia, R.S. "Marketing Productivity: Issues and Analysis," *Journal of Business Research* (55), 2002, pp. 349-362
32. Sjoberg, D.I.K., Anda, B., Arisholm, T., Dyba, M., Jorgensen, A., Karahasanovic, E.F., Koren, E.F. and Vokac, M. *Conducting Realistic Experiment in Software Engineering*, Proceedings of International Symposium on Empirical Software Engineering (ISESE' 02), pp. 17-26, 2002.
33. Teasley, S.D., Covi, L.A., Krishnan, M.S. and Olson, J.S. "Rapid Software Development through Team Collocation," *IEEE Transactions on Software Engineering* (28:7), 2002, pp. 671-683
34. Thomas, L., Ratcliffe, M. and Robertson, A. *Code Warriors and Code-a-Phobes: A Study in Attitude and Pair Programming*, Proceedings of the Thirty-Fourth Technical Symposium on Computer Science Education (SIGCSE 2003), ACM Press, 2003.
35. Williams, L. and Kessler, R. *Pair Programming Illustrated*, Boston, Massachusetts: Addison Wesley, 2003.
36. Williams, L. and Kessler, R.R. "All I Really Need to Know about Pair Programming I Learned in Kindergarten," *Communications of the ACM* (43:5), 2000, pp. 20-25
37. Williams, L., Kessler, R.R., Cunningham, W. and Jeffries, R. "Strengthening the Case for Pair Programming," *IEEE Software* (17:4), 2000, pp. 20-25

38. Williams, L. and Upchurch, R. *Extreme Programming in Software Engineering Education*, Proceedings of ASEE/IEEE Frontiers in Education, Reno, Nevada, October 2001. IEEE Community.
39. Yellen, R.E., Winniford, M.A. and Sanford, C.C. "Extroversion and Introversion in Electronically Supported Meetings," *Information & Management* (28), 1995, pp. 63-74
40. Zhang, Y. "Test-Driven Modeling for Model-Driven Development," *IEEE Software* (20:5), 2004, pp. 80-86
41. ZikMund, W.G. *Business Research Method*, Ohio, USA: Thomson South-Western, 2002.

附表1：事後比較顯著性檢定 (High-External)

High-External			
組別 (I)	組別 (J)	組差異 (I-J)	顯著性
1. ES-EN (mean: 14.00)	2	-3.92	0.006 ***
	3	-2.50	0.157 (*)
	4	-2.50	0.132 (*)
	5	-2.43	0.273 (*)
	6	-2.00	0.488 (*)
2. ES-IS (mean: 10.08)	1	-3.92	0.006***
	3	-1.42	0.346 (*)
	4	-1.42	0.272 (*)
	5	-1.49	0.990 (*)
	6	-1.92	0.307 (*)
3. ES-IN (mean: 11.50)	1	-2.50	0.157 (*)
	2	-1.42	0.346 (*)
	4	-0.00	0.629 (*)
	5	-0.07	0.615 (*)
	6	-0.50	0.698 (*)
4. EN-IS (mean: 11.50)	1	-2.50	0.132 (*)
	2	-1.42	0.272 (*)
	3	-0.00	0.629 (*)
	5	-0.07	0.901 (*)
	6	-0.50	0.890 (*)
5. EN-IN (mean: 11.57)	1	-2.43	0.273 (*)
	2	-1.49	0.990 (*)
	3	-0.07	0.615 (*)
	4	-0.07	0.901 (*)
	6	-0.43	0.965 (*)
6. IS-IN (mean: 12.00)	1	-2.00	0.488 (*)
	2	-1.92	0.307 (*)
	3	-0.50	0.698 (*)
	4	-0.50	0.890 (*)
	5	-0.43	0.965 (*)

*** P<0.01

附表2：事後比較顯著性檢定（High-Internal）

High-Internal			
組別 (I)	組別 (J)	組差異 (I-J)	顯著性
1. ES-EN (mean: 15.63)	2	-6.61	0.047**
	3	-4.38	0.325 (*)
	4	-2.13	0.627 (*)
	5	-4.63	0.246 (*)
	6	-6.50	0.081**
2. ES-IS (mean: 8.75)	1	-6.61	0.047**
	3	-2.50	0.202 (*)
	4	-4.75	0.131 (*)
	5	-2.25	0.136 (*)
	6	-0.38	0.394 (*)
3. ES-IN (mean: 11.25)	1	-4.38	0.325 (*)
	2	-2.50	0.202 (*)
	4	-2.25	0.609 (*)
	5	-0.25	0.985 (*)
	6	-2.12	0.385 (*)
4. EN-IS (mean: 13.50)	1	-2.13	0.627 (*)
	2	-4.75	0.131 (*)
	3	-2.25	0.609 (*)
	5	-2.50	0.539 (*)
	6	-4.37	0.217 (*)
5. EN-IN (mean: 11.00)	1	-4.63	0.246 (*)
	2	-2.25	0.136 (*)
	3	-0.25	0.985 (*)
	4	-2.50	0.539 (*)
	6	-1.87	0.310 (*)
6. IS-IN (mean: 9.13)	1	-6.50	0.081 (*)
	2	-0.38	0.394 (*)
	3	-2.12	0.385 (*)
	4	-4.37	0.217 (*)
	5	-1.87	0.310 (*)

* P<0.1 ** P<0.05

附表3：事後比較顯著性檢定（Low-External）

Low-External			
組別 (I)	組別 (J)	組差異 (I-J)	顯著性
1. ES-EN (mean: 9.40)	2	-0.53	0.889
	3	-0.80	0.831
	4	-0.77	0.826
	5	-0.85	0.662
	6	-0.11	0.789
2. ES-IS (mean: 9.93)	1	-0.53	0.889
	3	-1.33	0.936
	4	-0.24	0.934
	5	-0.32	0.808
	6	-0.64	0.909
3. ES-IN (mean: 8.60)	1	-0.80	0.831
	2	-1.33	0.936
	4	-1.57	0.998
	5	-1.65	0.895
	6	-0.96	0.986
4. EN-IS (mean: 10.17)	1	-0.77	0.826
	2	-0.24	0.934
	3	-1.57	0.998
	5	-0.08	0.895
	6	-0.88	0.986
5. EN-IN (mean: 10.25)	1	-0.85	0.662
	2	-0.32	0.808
	3	-1.65	0.895
	4	-0.08	0.887
	6	-0.96	0.895
6. IS-IN (mean: 9.29)	1	-0.11	0.789
	2	-0.64	0.909
	3	-0.96	0.986
	4	-0.88	0.983
	5	-0.96	0.895

※ 所有組別皆未達顯著

附表4：事後比較顯著性檢定（Low-Internal）

Low-Internal			
組別 (I)	組別 (J)	組差異 (I-J)	顯著性
1. ES-EN (mean: 13.16)	2	-6.91	0.022**
	3	-1.76	0.512
	4	-7.66	0.017**
	5	-6.46	0.018**
	6	-2.56	0.353 (*)
2. ES-IS (mean: 6.25)	1	-6.91	0.022**
	3	-5.51	0.249 (
	4	-0.75	0.391 ()
	5	-0.45	0.439 ()
	6	-4.35	0.326 ()
3. ES-IN (mean: 11.40)	1	-1.76	0.512 (*)
	2	-5.51	0.249 (*)
	4	-5.90	0.217 (*)
	5	-4.70	0.306 (*)
	6	-0.80	0.844 (*)
4. EN-IS (mean: 5.50)	1	-7.66	0.017**
	2	-0.75	0.391 (*)
	3	-5.90	0.217 (*)
	5	-1.20	0.178 (*)
	6	-5.10	0.279 (*)
5. EN-IN (mean: 6.70)	1	-6.46	0.018**
	2	-0.45	0.439 (*)
	3	-4.70	0.306 (*)
	4	-1.20	0.178 (*)
	6	-3.90	0.349 (*)
6. IS-IN (mean: 10.60)	1	-2.56	0.353 (*)
	2	-4.35	0.326 (*)
	3	-0.80	0.844 (*)
	4	-5.10	0.279 (*)
	5	-3.90	0.349 (*)

** P<0.05

