

## 教育部教學實踐研究計畫成果報告

計畫編號/Project Number：PEE1100827

學門專案分類/Division：工程

執行期間/Funding Period：2021.08.01 – 2023.01.31

### 計畫名稱：

培養團隊吸收與衝突協調能力之敏捷式創新教學設計於資訊畢業專題課程

配合課程名稱：資訊系統畢業專題

計畫主持人(Principal Investigator)：陳仲儼

協同主持人(Co-Principal Investigator)：

執行機構及系所(Institution/Department/Program)：國立中央大學資訊管理學系

成果報告公開日期：

立即公開 延後公開(統一於2024年9月30日公開)

繳交報告日期(Report Submission Date)：2023/3/1

# 培養團隊吸收與衝突協調能力之敏捷式創新教學設計於資訊畢業專題課程

## 一. 本文

### 1. 緒論與研究問題

大學畢業專題(Capstone project course 或稱為 Senior projects)對於資訊相關科系而言是一門重要的攻頂課程。畢業專題主要是希望學生能夠將過往學過的技術得以應用、磨練解決問題的能力，以及透過與成員團隊密切合作共同開發一個實際的軟體系統以完成這一門課程(Sedelmaier and Landes, 2020; Tsai et al., 2018; Chen et al., 2014; Chen et a., 2011)。此外在資訊科技與網路迅速發展下，也逐漸發展出宅世代與個人化生活方式。人們的生活透過網路與線上聊天以鍵盤代替見面、取代說話，逐漸失去面對面溝通與社交協調的機會甚至於能力。這對於習慣於自行練寫程式的資訊領域之大學學生而言更是如此。這是由於資訊教育領域通常較著重於資訊與軟體發展技術的養成，也就是學會撰寫程式，許多學生因而認為只要學會寫程式即是未來職場的人生勝利組。另外就軟體與系統發展來說，過去是以瀑布式(Waterfall)與內容結構化為主。瀑布式發展主要是要求需求在一開始即談定後就盡量不再更動，並在過程中分不同階段—例如需求、系統分析、系統設計、撰寫開發等階段，將系統內容一次完整帶出。而結構化則是將系統內容明確劃分，以利循序整合。專案成員在這樣的開發模式與明確分工的過程中只要將分內工作完成，是可不用花太多的時間與成本在溝通與處理變動(Sedelmaier and Landes, 2020)。

然而，現今由於資訊與軟體需求逐漸個人化與特色化，加上商務環境加速變遷，使得軟體發展這件事越發複雜。現今的軟體開發面臨使用者可能隨時會有新需求，並藉由溝通來了解功能與演示部分成果、而且軟體內容越發複雜、越需要更多人員一起合作與不斷地溝通以逐步完成開發。因此演化式發展成為現在與未來軟體開發所需，而近年來逐漸受重視的敏捷發展法(Agile method)就是這樣的開發概念下之重要實踐。無論是演化式發展或是敏捷發展法，其核心概念在高度不確定的變動環境下不再一開始即詳細清楚劃分，而是增量(Incremental)與迭代(Iterative)發展、並強調開發流程要不斷地調整與再規劃，以能更符合與因應當下實際的情況與客戶需要。因此，如何進行敏捷開發是現今演化式軟體發展趨勢下之重要課題，而這些課題背後的教育意涵—即協同合作與溝通，正是畢業專題課程對於學生在實際從事系統開發時所應給予的訓練，特別是正值高年級的專題學生在進入研究所或邁入職場前，對於溝通與社交協調能力所必須養成的重要與最後時機。

運用敏捷發展成為現今畢業專題課程所採用之趨勢(Kamthan, 2021; Tsai et al., 2018; Ding et al., 2017)。然而，有別於現有研究大多著眼於運作方式之學習與各式工程學習成效，本計畫特別從衝突與協調方面加以探討。這是由於軟體開發是屬於一個創新研發且高度變動的過程(Lee et al., 2021)，需要團隊成員間的腦力激盪。因此在開發過程中會有許多意見上之衝突，並且有必要在畢業專題課程就讓學生體驗並養成溝通協調衝突的能力。此外就演化式發展來說，能否在有限的時間之內從紛雜意見中發展共識、並有效找到解決方案並落實在產品創意與品質，是專題能否順利進行之關鍵。然而現有文獻鮮少針對敏捷式學生專題來探討如何來建立這些軟實力。而自 2007 年起，計畫主持人在系上擔任畢業專題指導老師。由於軟體系統發展強調創新研發與計畫要配合變化，而衝突則是創新與再規劃過程中不可豁免之重要元素，因此畢業專題強調學生團隊學習與創新研發以及因應變化之能力、並且擁抱衝突以激盪出更佳的創意與做法。然而，計畫主持人看到了歷屆以來學生在從事畢業專題時常有的自我認知一即宅起來把程式寫出來最重要，而忽略了培養團隊吸收能力與衝突解決能力。對於以後在職場上相當重視團隊協作的軟體開發而言，這些能力是非常基本的素養，但都不是在典型課堂所能學習到。因此，本計畫將以專題過程這種非典型的課堂場域為主，來幫助學生藉由學習這項活動來實踐團隊吸收與衝突協調能力的養成，加上實施敏捷發展法，針對團隊合作中的協調能力進一步檢驗於學生在畢業專題的成效，並特別有以下之研究問題：

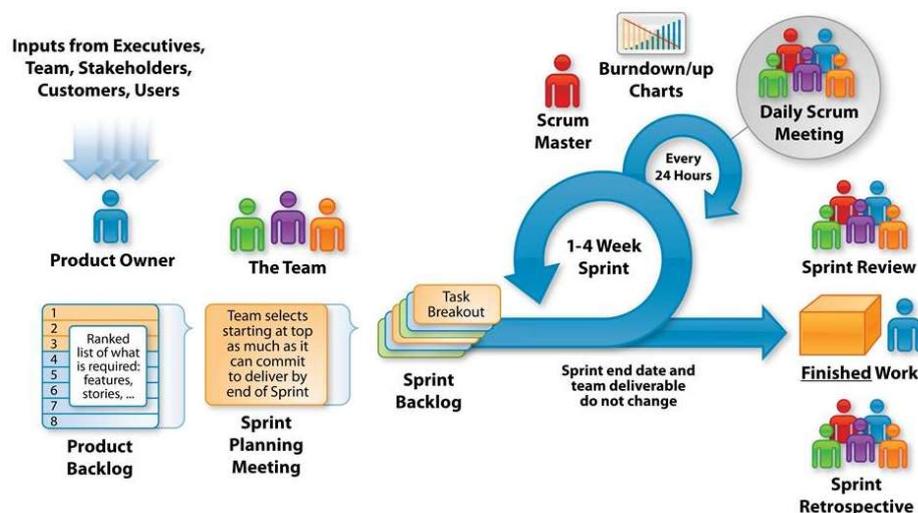
1. 敏捷式專題是否會隨著系統規模增加而能有效幫助控制產品品質。
2. 敏捷式專題是否可以有效協助提出產品特色以展現團隊之衝突協調能力。

為了探討上述的問題，本計畫提出了『敏捷式專題教學設計』(教學方法與細節見後說明)，並在課程過程中收集學生團隊的資料來了解成效。這些團隊均採用敏捷中的 Scrum 模式(Maximini et al., 2018; Lous et al., 2018)來開發，並且運作了 12 次以上的衝刺(Sprint)，本報告將之截頭截尾以採其中五次進行分析以確保內部(Internal validity)與結果效度(Conclusion validity)(Slack and Draugalis, 2001; Parker, 1993)。資料的收集則包含數據紀錄以及與訪談內容。問卷內容根據團隊衝突協調能力的問卷模型，對專題學生進行訪測，問卷設計參考 Liang et al. (2010)、Ali et al. (2018)與 Popaitoon and Siengthai (2014)所提出的問卷模型。針對晤談而言，內容為半開放的形式，即是先設計好簡答型題目並進一步收集學生之敘述性回答。實驗數據部分則擷取至專題組於該開發過程的實驗工作簿與會議資料。資料收集完畢後再將數據加以整理、並進行統計分析來驗證上述研究問題與所產生之效益並加以討論。在本次成果報告中，

## 2. 文獻探討

### 2.1. 資訊系統開發：敏捷發展法

近年來，敏捷發展在軟體與資訊系統開發領域日趨受到重視，這是由於現在的開發和用戶商務環境變化快速而致。敏捷發展主要是可以讓用戶或使用者更快速的見到並使用系統，同時可以及早獲得使用後的回饋讓開發團隊能夠持續地修正與優化(Lee and Chen, 2019b; Lee et al., 2020a)。然而，敏捷開發是一種概念，因此在學理上陸續的發展出許多模式來實踐敏捷發展。其中最重要的莫過於是 Scrum 模式(Schwaber and Beedle, 2002; Maximini et al., 2018)，如圖一所示。Scrum 模式強調團隊合作與相互溝通，此模式定義出了四個會議模式，分別是 Sprint planning、Daily scrum、Sprint review、Retrospective，並將軟體開發的關鍵人員分為三種主要角色：Product owner、Scrum master、Development team，這樣的分化有助於團隊合作與溝通的進行。同樣地，Scrum 會依照敏捷發展的策略做增量與迭代，將產品需求視為待出清的需求開發清單(Product backlog)，開發團隊一次僅針對部分的需求做開發並以逐步的方式實作出系統。每一次的開發循環稱之為衝刺(Sprint)，用以規畫該次團隊所需要做的工作項目並且完成之。



圖一：敏捷發展 Scrum 模式

除此之外，敏捷發展強調自組織發展(Self-organization) (Hoda et al., 2012; Almadhoun and Hamdan, 2018)的能力，並且與傳統相比，管理者從命令者轉為從旁協助與支持性的角色(Parker et al., 2015; Lous et al., 2018; Bhatta and Thite, 2019)。在此情況下，對於專題學生團隊來說，如何不總是依靠指導老師發號施令而自主的組織團隊並開發專題是值得關注的課題。由於敏捷發展提倡用戶可以因應不同開發環境的變化允許在過程中提出變更，因此不論是敏捷發展或是 Scrum，軟體開發的流程都強調不斷地修正與再規劃(Tripp & Armstrong, 2018; Campanelli et

al., 2018; Lee et al., 2020)，因此開發過程充滿動態變化。此外由於 Scrum 為強調軟體演化與迭代發展的敏捷式開發概念中最重要實踐方式(Akbar, 2019; Williams and Cockburn, 2003)，因此有需要在學生的資訊畢業專題中特別加強這一方面的訓練與實作能力(Mahnic, 2011; Rodríguez et al., 2014; Tsai et al., 2018)。

## 2.2. 衝突協調能力

許多研究指出，創新研發類型的工作為一個衝突的過程，這對於軟體開發來說亦是如此(Lee et al., 2020; Lee et al., 2021; Liang et al., 2010)。這是因為軟體開發是由不同人員合作發展而成，故在過程中的衝突更是無法避免。衝突展現在許多方面。因此，有必要在專題教育特別凸顯這一部份並且讓學生了解與學習實際的處理。衝突並非一定是負面的，在與成員們腦力激盪的過程中，衝突可以轉化為反思與啟發，以真正幫助產品的創新發展以及團隊的成長。但也有研究呼籲，團隊衝突應該加以教育與導引如何解決，否則會負面影響團隊運作績效與決策品質(Jehn and Mannix, 2001; De Wit et al., 2012; De Clercq et al., 2015)。因此有需要在專題開發中讓學生學習如何積極面對與成熟解決衝突。而面對衝突的化解，有研究即指出軟體開發團隊成員間的協調能力即是一項重要的因素(Andres and Zmud, 2002; Liu et al., 2011; Kudaravalli et al., 2017)，這對於自組織的敏捷團隊而言，更是重要(Berntzen et al., 2019)。

軟體開發的衝突展現在許多方面。就一般情況來說，基本上衝突可分為工作衝突(Task conflict)與關係衝突(Relationship conflict)等(De Dreu and Weingart, 2003)。本計畫著重於工作衝突與相關教育，這是因為除了與專題課程的專業訓練有關之外，工作衝突也常導致其他衝突類型的發生(van Woerkom & Van Engen, 2009)。對於團隊如何處理與解決衝突，學者指出協調能力的養成是關鍵之一。對此，在學理上有研究提出協調理論並進一步定義出團隊活動中的四大協調機制來解決衝突(Chen et al., 2013; Crowston et al., 2015; Malone and Crowston, 1994; Balta and Krcmar, 2018; Berntzen yet al., 2019): 從供應關係(Managing producer/consumer relationships)、從資源共享(Managing shared resources)、從同步制約(Managing simultaneity constraints)、以及從任務關係(Managing task / subtask)。本計畫認為，雖然協調理論是基於一般團隊活動的合作行為所建立，但對於軟體開發中的衝突處理，也應該可嘗試運用之。這主要是由於軟體開發的衝突有許多是來自於供需雙方的認知差異，並且軟體開發過程也存在著資源共享的問題，以及軟體開發的諸多工作任務確實也有高度的脈絡依存關聯與快速跟進(Fast tracking)的需要。因此本計畫嘗試運用上述的四大協調機制於專題教育中，並實驗學生在運用後對於團隊吸收能力與衝突協調能力等兩項學習主題的學習成效的提升程度。

### 3. 研究設計與方法

#### 3.1. 教學設計

由於專題開發需要學生創新的發想與緊密的團隊合作，因此在過程中會有許多問題的產生與亟待解決，衝突是最主要的問題來源，此次計劃主要是著重在解決工作衝突的部分，例如在開發工作項與流程規劃時所產生的工作衝突。因此本計劃將讓學生遵循 Scrum 模式中的 Sprint planning、Sprint review、Retrospective 與 Daily scrum 等四個會議來實踐團隊合作與溝通。然而由於其並無具體的操作內容，故本計畫針對教學主題—即團隊吸收與衝突協調能力的養成以及演化式發展的核心活動—軟體流程調適等如何展現在這四種會議中、加上運用德菲法(Humphrey-Murto and de Wit, 2019; Skulmoski et al., 2007)以及 NGT 法則(Muthulingam et al., 2019; Potter et al., 2004; Gallagher et al., 1993)於促進成員意見的自由表達與共識的達成，進一步提出具體的實施方法，本計畫稱為『敏捷式專題教學設計』，內容如表一所述。

對於衝突意見的協調，本計畫運用德菲法(Delphi method)(Humphrey-Murto and de Wit, 2019; Skulmoski et al., 2007)。德菲法原本並非用於軟體開發，在本計畫運用其做為團隊異中求同、達成共識的實施方法。過程為：一開始針對某爭議性議題先由專家各自寫下估算數字與理由，在強調提出意見者在不具名的情況下交由主持人一一將各想法念出。當差異過大得進入第二回合，各自再想下想法後交由主持人再公告一次。如此重複進行中參與者會因為相互參考而能夠將議題收斂，直到可接受之範圍。德菲法之特色在於：(1)不具名：在排除同儕壓力或效仿作用之情況下能夠真正呈現出專業的說明與少數但具創意之意見；(2)收斂式演化式 (Convergent evolution)：透過重複估算的回合來越加理解他人與修正自己直到達成某種程度之共識。目前有少數研究將德菲法運用在敏捷團隊運作中(Torrecilla-Salinas, et al., 2019)，然而主要還是用於專家群體決策(例如開發時程估算)，本計畫則運用在成員衝突的協調與解決之上，並檢討學生的在這方面的學習成效。綜合上述，本計畫所建立的『敏捷式專題教學設計』細節如表一所示：

表一：結合衝突協調教學主題之敏捷式專題設計內容

衝刺循環	說明	具體實施設計
Sprint planning 衝刺規劃會議	會議上半部由學生團隊、Scrum master、與 Product owner(可為指導老師)一起規劃待開發功能清單(User stories)。會議下半部學生則由團隊與 Scrum master 根據待開發清單規劃與調整下一周期的開發內容與做法、並辨識可能的開發風險。由學生團隊自行召開，並做會議紀錄。	Sprint planning 會議主要是規劃下一次開發周期的流程與做法，此即軟體流程調適的任務要素之一。在這會議中，本課程將運用德菲法(Delphi method)與 Planning poker <sup>註一</sup> 來讓學生練習如何有效進行腦力激盪與工作規劃、並在思考解決方案與工作排定中進行協調

Daily scrum 每日會議	學生團隊與 Scrum master 每日需要齊聚約 15 分鐘以相互關懷前一天的開發狀況與接下來的風險預防	與處理衝突。
Sprint review 衝刺成果會議	學生團隊、Scrum master、與 Product owner 一起召開檢視過去的衝刺周期的開發成果	無論是 Sprint review 或者是 Retrospective 會議，都是強調檢討與成長，以作為團隊在規劃與如何修正下一個開發周期時參考。因此在相互檢討的運作設計上，本課程將運用 Nominal group technique (NGT) <sup>註二</sup> 法則，以利學生能成熟地相互啟發與成長。
Retrospective 衝刺檢討會議	學生團隊與 Scrum master 進行本次開發的檢討，紀錄衝突議題與解決內容，以利下次開發周期的改進，並提列風險作為下一次開發週期的檢視與預防；並且表揚本次表現優良的成員。指導老師視情況參與，主要是指導會議進行的方式。	
<p>註一：Planning poker 是敏捷發展法用以取代傳統人日估算的方式。其訴求在於運用類比概念，如根據某功能的故事點數來類比目前欲開發的功能的規模。估算時則以 Planning poker 的 13 張牌(牌面數字從 1/2, 0, 1, 2, 3, 5, 8, 13, 40, 100 等)來組合並提出看法，並搭配德菲法來逐漸收斂出大家對於工作規劃上的共識。</p> <p>註二：NGT (Nominal Group Technique)法則原非用於軟體開發，本計畫運用其做為分享與檢討具體方法。過程為，對於某事件的原因檢討，成員不具名寫下各自的想法，郊遊主持人一一念出，並開始一起討論，最後以投票方式決定主要原因。由於式不具名，因此每個人皆可以自由地表達心中想法，再讓團隊去討論。並且一旦決議形成，則為團隊的共識，並非是個人意見。</p>		

### 3.2. 專題內容描述

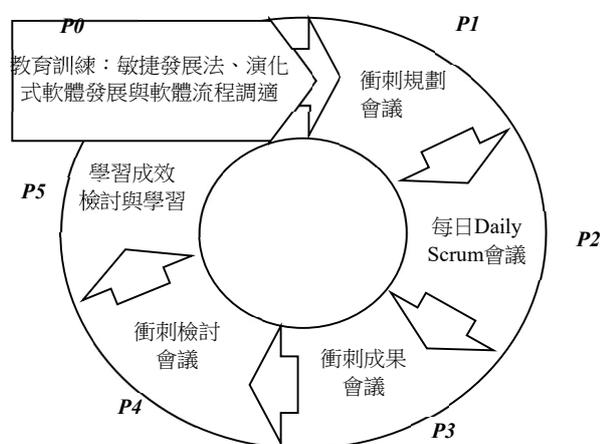
參與本次計畫的專題開發團隊依序標籤為 A, B, C, D 與 E 共五組。學生團隊皆為國立中央大學資訊管理學系的資訊系統發展專題課程而進行專題開發。為求實施對象的綜合性，五組專題中有兩組為產學合作(簡稱為企業組)，其旨在推廣合作企業的資訊系統，讓消費大眾能透過該應用程式以更便捷管道獲得該公司各項產品的資訊或是進行購買。其餘三組則是循傳統專題開發方式—即自行創意發想並將之發展成系統功能—例如組 A 研發手機用行程記錄 APP 應用程式，其能夠以拍照掃描的方式辨識並且匯入行程；組 B 則是開發一個 APP 可將生活中書面圖像資訊以電子化並能加以編輯，進而創造出使用者自己的電子筆記。

對於上述教學設計，以下進一步扼要說明執行內容。首先，對於 Product Owner (PO)的設定，企業組是由合作企業指派關鍵使用者來擔任，而自行研發的兩組則由指導教授擔任之。再來，各專題的開發內容依據兩學期的時間來酌量定義，但不由指導教授來介入干涉，以真正作到自組織團隊(Self-organizing teams)的實踐，同時也是讓學生能練習獨當一面，跟使用者/PO 的接洽來實際體會功能需求與相關意見之協調。指導教授則是從旁指導如何運作，像是德菲法或 Planning Poker 如何進行等等。

## 4. 教學暨研究成果

### 4.1. 教學過程與成果

敏捷式專題教學設計的過程如圖二所示。在課程一開始前兩周(即圖中 P0)，指導老師先針對敏捷發展以及本計畫所運用到的技巧與概念來對學生進行教育訓練。之後，專題學生依 Scrum 四個會議來運作每一次之衝刺。而為了便利社群活動之進行與文書的紀錄，本計畫同時要求學生學習目前廣為使用於敏捷發展的軟體工具—例如 Trello 以及 Jira (Atlassian, 2020a; Atlassian, 2020b)。此外計畫主持人在期初並參與學生會議來導引敏捷發展之進行—特別是對於系統需求從提出到解決方案的建立過程團隊在吸收能力的四個機轉情形。以及針對所面臨的問題如何從四個協調機制來導引學生進一步協調解決。



圖二：專題課程實施之歷程設計

在每一次衝刺循環結束後，指導老師與專題學生進行學習成效檢討與學習(P5)，以記錄每次衝突與轉化歷程—特別是四個協調機能運用於衝突處理與團隊的四個知識機轉於轉化之過程。由於本計畫的教學評量會需要在過程中依評量方式(問卷與晤談等)來分別收集資料與數據，故會利用這一步驟(P5)來進行之。若以 18 週的學期而言以及扣除期末時間，加上每個衝刺依難易程度從一至三週不等，平均為 2 週的變形長度，因此學生團隊所運作的衝刺循環預計會有 6~7 圈。每圈均會進行當次的學習成效檢討與學習。

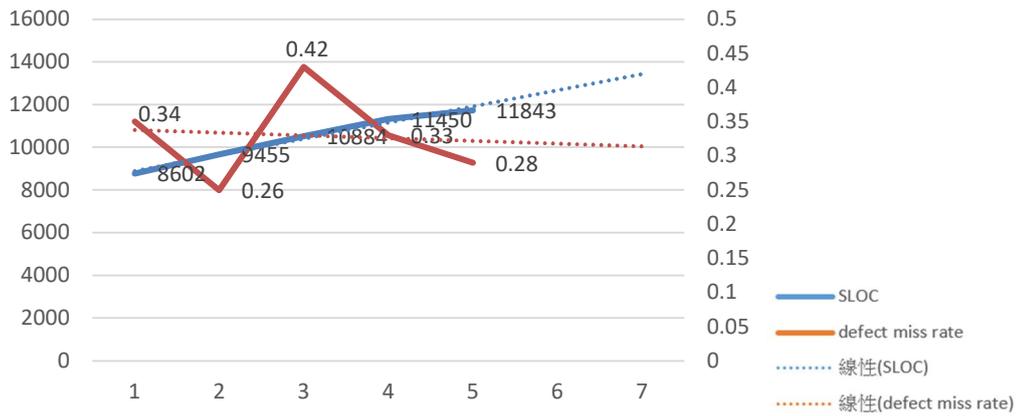
在專題的開發過程中，我們為了從資料面來剖析成果，收集了專題團隊的相關資料與數據，以利分析提出的敏捷式專題教學設計。具體而言，我們從四大方面來收集資料。第一，為了掌握各組別在執行 Scrum 的情形，計劃團隊收集了每一次 Sprint 所產出的 Planning meeting、Review meeting 與 Retrospective meeting 的開會紀錄。第二，為了檢驗所提出的敏捷式專題創新教學設計是否隨著專題過程中系統規模增加能有效地幫助控制產品品質，計劃團隊收集每次 Sprint 完成的程式碼行數以及發現的 Bug 數量。第三，為了解敏捷式專題是否有效協助解

決衝突，分析開發團隊是否隨著專題過程之推進而能提升協調能力，計劃團隊收集了團隊的衝突協調時間與每次 Sprint 完成的程式碼行數。第四，軟體開發當中創新是一項重要的元素，本計畫以問卷的方式衡量各組的產品特色，問卷內容則參考 Cropley and Cropley (2010b)的創新量表，如附件所示。最後，我們呈現了專題學生會議與系統研發成果之情形，如附錄所示。會議部分由於計畫執行時恰逢國內針對 COVID-19 疫情而嚴密監控下，學校皆採遠距教學，故會議之進行是採取線上方式。

由於本計畫的進行是在學生已開發畢業專題一段時間後才開始，所以各組的程式已累積有一定的規模。為了衡量敏捷式專題創新教學設計是否會隨著專題過程中系統規模增加能有效地幫助控制產品品質，我們收集了各組專題團隊程式碼行數作為系統規模的依據，而程式碼行數又可分為可執行的程式碼行數以及包含註解的程式碼行數，故我們使用已剔除註解的程式碼行數，如此一來可以降低過多的註解造成可能程式規模過大的假象。在產品品質方面，本計畫使用 Defect removal efficiency 作為衡量專案團隊找錯誤(Bug)的能力，使用到的公式為缺陷遺失率 Defect miss rate (Kan, 2002, p. 124; Jones, 2008)，其代表在本次 Sprint 有被發現的缺陷當中尚未被發現的缺陷之機率，其公式為： $1 - D_i / [(D_{i+1} - D_{ci+1}) + D_i]$ 。其中  $D_i$ :本次 sprint 發現並解決的 Bug 數， $D_{ci}$ :本次 sprint 產生並解決的 Bug 數。以下進一步針對研究問題來加以探討相關成果。

### (一)、產品品質

在資料分析階段，本計畫探討學生專題團隊在敏捷開發之下，隨著程式碼規模逐漸增加之際，能否有效地控制住產品品質。從圖三可看到在歷經每一次的衝刺，程式碼規模是逐步的增加，而缺陷遺失率雖有一些起伏，但似乎是逐步下降的情形。本計畫進一步探討如此的結果是否具有代表性。因此，我們針對程式碼規模與缺陷錯失率進行線性回歸而得圖三的結果。從圖中可明顯看出程式碼規模逐漸增加的趨勢，同時伴隨著缺陷遺失率逐漸的減緩。這驗證了使用敏捷開發的團隊在逐步地增加程式規模時，可以有效地在每一次衝刺循環中找到程式缺陷，進而能夠控制住產品品質。



圖三：程式碼平均規模與 Defect miss rate 關係圖

另外，探討了各組別之間的缺陷錯失率是否有所差異，本計畫提出了以下假設，並進行 T 檢定分析：

H<sub>0</sub>:各組別的缺陷遺失率沒有顯著差異

H<sub>1</sub>:各組別的缺陷遺失率有顯著差異

表二：t 檢定—成對母體平均數差異檢定

平均數	0.206	0.316	0.206	0.434	0.316	0.434
變異數	0.01708	0.00388	0.01708	0.04518	0.00388	0.04518
觀察值個數	5	5	5	5	5	5
皮耳森相關係數	-0.9084		-0.1082		-0.2043	
假設的均數差	0		0		0	
自由度	4		4		4	
t 統計	-1.3009		-1.9512		-1.1305	
P(T<=t) 單尾	0.1316		0.0614		0.1607	
臨界值：單尾	2.1318		2.1318		2.1318	
P(T<=t) 雙尾	0.2632		0.1228		0.3214	
臨界值：雙尾	2.7764		2.7764		2.7764	

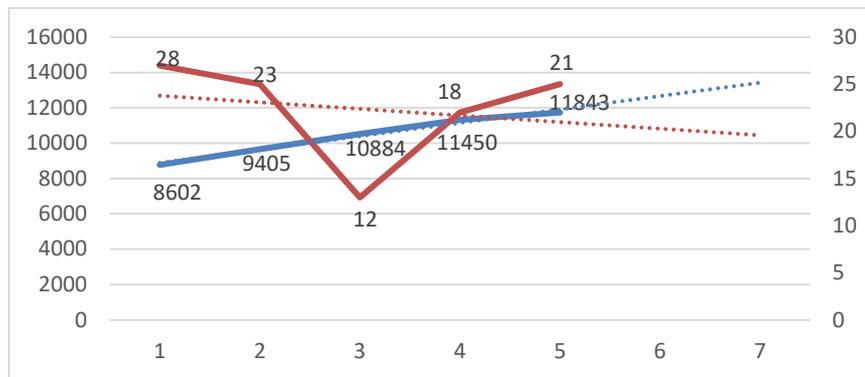
各組之間 T 檢定的結果如上表二(以 A, B, C 三組為例)，分別有 6 次的衝刺循環在本次的分析當中。以組 A 為例，其平均缺陷遺失率為 0.206，組 B 的平均缺陷遺失率為 0.316，組 A 與組 B 的相關係數為-0.9084，t 統計值為-1.3009，雙尾顯著性 p 值=0.2632>0.05，接受 H<sub>0</sub>。因此組 A 與組 B 的缺陷遺失率並沒有顯著差異。再則，組 A 的平均缺陷遺失率為 0.206，組 C 的平均缺陷遺失率為 0.434，兩組的相關係數為-0.1082，t 統計值為-1.9512，雙尾顯著性 p 值=0.1228>0.05，接受 H<sub>0</sub>。因此兩組的缺陷遺失率並沒有顯著差異。另外組 B 的平均缺陷遺失率為 0.316，組 C 的平均缺陷遺失率為 0.434，兩組的相關係數為-0.2043，t 統計值為-1.1305，雙尾顯著性 p 值=0.3214>0.05，接受 H<sub>0</sub>。因此兩組的缺陷遺失率並沒有顯著差異。在三組之間兩兩比對之後，我們可以發現各組之間的缺陷遺失率並沒有顯著差異，從這裡可以看出這三組開發團隊都是使用敏捷開發，並且在這樣的開發過程下，能夠一致的將程式當中的缺陷有效地找出，並且隨著程式碼規模愈趨龐大，能夠一致有效地維持甚至是降低缺陷遺失率，

進而能夠提升產品品質，使得每一次的衝刺結束時都能夠提交出品質穩定的產品，而不會因為規模愈趨龐大，造成缺陷遺失率與規模呈現正比的成長。

## (二)、團隊協調能力

從圖四來看，學生開發團隊在敏捷開發之下於有限期數的觀察中，團隊對於處理衝突與協調能力似乎有提升。本計畫進一步透過線性回歸法得到如圖四之結果。圖中也畫出了線性回歸來看之後的趨勢為負相關，代表使用敏捷開發的團隊在隨著程式規模增加之下，團隊在衝突議題的協調上所花費的時間有顯著縮短，並有效地提升團隊的衝突協調能力。

衝突的發生對於開發團隊來說不是一件壞事，這代表著團隊成員對於共同的目標能否實現相當在意。以開發團隊的角度來說，也可以更快的讓團隊成員對於彼此有更加深入的了解。但若是能夠更有效率的處理衝突，縮短在面對衝突時所需要花費的時間，這能夠使開發的步調更佳的順暢。而在實施敏捷發展之下，以 Scrum 模式為例，從一開始的規劃會議，規劃下一次待開發清單，再來是每日的站立會議來了解各成員間的開發情形以及交流，最後的成果會議來檢視開發成果以及檢討會議來為內容做改善，這一連串有條理的運作模式有助於團隊對於衝突可以更加有效的解決。

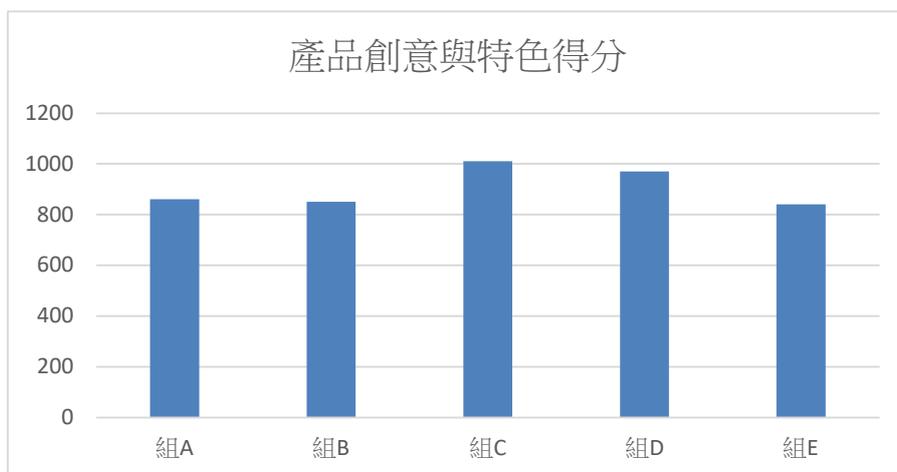


圖四：程式碼規模與衝突協調時間之關係圖

## (三)、產品特色量表

軟體系統發展強調創新研發，一個產品有好的創新性可能可以為我們帶來更便利的服務或是截然不同的使用模式，因此本計畫希望可以衡量出各產品的創新、特色。產品特色量表是參考了 Cropley and Cropley (2010b)所設計的量表製作成符合本計畫的量表(見附錄)，並將累積的資料整理成圖五。我們統計了各組別在產品特色量表總共獲得的分數。為了衡量各組產品特色的分數，我們採用同仁審查(Peer review)(Bohannon, 2013; Lee et al., 2013)來檢視創新成果。同仁審查是一種技術成果審查方式，即是一位作者的成果被同領域的其他同儕加以評審。在該圖中我們可以看到，組 C 獲得了最高分，他們的 APP 主要是希望能夠幫助使用者將

課本、雜誌上的圖文案例經由拍攝之後利用他們的物件辨識技術以及強大的編輯系統來創造出屬於使用者的個人化圖文筆記，這樣的創造力也讓該組獲得了最高的票數。



圖五：產品特色評分

#### 4.2. 教師教學反思

首先，本課程歷經計畫後期的疫情嚴峻時期，然而，這反而凸顯出敏捷發展其高度彈性與應變能力，真正練習到在彈性開發環境下的團隊合作。然而，針對彈性開發與團隊合作，主持人有以下反思與建議。首先，如古諺所言之工欲善其事必先利其器，彈性開發必須借助於資訊科技，特別是所使用的視訊工具。主持人有兩建議，首先在視訊會議時務必要求所有成員都能夠開鏡頭。此雖是一小舉動，但在開鏡頭的情況下，團隊會議顯得有效率，且學生較能夠聚焦於會議的運作。再來，在視訊會議時，可不定時請不同學生發表意見，以避免往往只有少數幾位或是專題組長發言的狀況，較能彌補非面對面會議所失去的參與感，並真正實踐團隊合作。

另外，經過這次計畫的執行，計畫主持人深刻了解到資訊領域的畢業專題是適合敏捷式開發。其原因在於：第一，在現在的業界當中，大多數的專案團隊都是使用敏捷發展資訊軟體，可以讓學生提早接觸了解其運作模式；第二，敏捷發展以更快速、更有效率的方式運作，在此模式下協助開發團隊逐步製作畢業專題。在這樣的模式—即讓學生自組織(Hoda et al., 2012)運作開發之下，與過往老師說一項學生照做一項的開發方式相比，可以確保專題團隊的開發步調，學生的探索動機更強，最後可讓專題有更高的完成度。

就老師而言，其在專題指導的專業與技術是無庸置疑；但就如何帶學生以及與學生互動上應該是可以討論與尋求改進空間。而透過敏捷式專題過程，這部分或許是可以做到。也就是說，做老師的也可以透過敏捷開發的會議設計的學生反饋來了解自身在教學方面是否有任何改進之需要。例如在每次衝刺結束的檢討會議，學生除了提出自己在下一次開發應該如何改

變以求更好的開發績效之外，教師也可以藉此機會進行問卷調查，來了解教學做法上是否有任何問題值得進一步檢討改進，以真正做到教學相長的境界。

#### 4.3. 學生學習回饋

在專題期末的座談會中，我們特別針對所提出的敏捷是專題運作模式與過程中的資料收集進行檢討，希望從學生角度來了解相關的回饋。首先對於敏捷運作的方式，學生們都非常贊同是非常具有彈性的開發模式，特別適用於像專題這種高度不確定以及邊摸索邊發展的研發型專案。但必須要注意的是，敏捷可以使專案盡早看到成果，但無法讓專案提早結束。相反地，有學生指出，因為是還在學習緣故，對於每次衝刺的規劃往往都不甚精準，導致於常常時間到功能尚未完整開發完畢，這時就需要再開一次衝刺來完善當期的要求。

再則，對於衝突協調的學習，學生普遍對於所提出的方法感到學以致用。特別是德菲法，讓學生能夠勇於提出任何相異意見，真正作到了協調而不是『團隊成員誰說了算』。此外搭配了NGT(Nominal Group Technology)的概念，使得學生再充分協調後，能夠以團隊為依歸，進而促進對團隊的向心力。然而，也有學生指出，若同學缺乏了學習動機(特別提到無論是什麼課程總是會有少數同學只是為了混學分而讓其他成員負擔加重)，這些方法還是功虧一簣。指導教授也肯定這樣的心得，並且提出可嘗試以同儕評分方式來記錄成員間彼此對於專題的積極度與貢獻，以利指導教授在評分時參考。

#### 4.4. 建議與省思

感謝教育部通過本次計畫，讓計畫主持人能夠針對畢業專題這一重要的課程發揮得更好。除了上述的成果，也因為這堂課而獲得學校頒發教學優良獎(2022年)，並且得以將部分成果發表於學術研討會。而學生的標線也不遑多讓，有兩組分別獲得校外專題競賽第一名及系上選拔(共20多組)佳作，雖非太大的殊榮，但也有激勵之效果。在未來，基於這次計畫的學習與成果以及各方的回饋意見，主持人將繼續努力，在畢業專題的授課上能更進步。

## 二. 參考文獻 References

1. Ali, I., Musawir, A.U., & Ali, M. (2018). Impact of knowledge sharing and absorptive capacity on project performance: the moderating role of social processes. *Journal of Knowledge Management*, 22(2), 453-477.
2. Akbar, R. (2019). Tailoring Agile-Based Software Development Processes. *IEEE Access*, 7, 139852-139869.

3. Almadhoun, W., & Hamdan, M. (2018). Optimizing the self-organizing team size using a genetic algorithm in agile practices. *Journal of Intelligent Systems*, 29(1), 1151-1165.
4. Andres, H.P., & Zmud, R.W. (2002). A contingency approach to software project coordination. *Journal of Management Information Systems*, 18(3), 41-70.
5. Berntzen, M., Moe, N.B., & Stray, V. (2019, May). The product owner in large-scale agile: an empirical study through the lens of relational coordination theory. In *International Conference on Agile Software Development* (121-136). Springer.
6. Bhatta, N.M.K., & Thite, M. (2019). Agile approach to e-HRM project management. e-HRM: Digital approaches, directions & applications.
7. Chen, C.Y. and Chong, P. (2011). Software engineering education: A study on conducting collaborative senior project development, *Journal of Systems and Software*, 84(3), 479-491
8. Chen, C.Y., Hong, Y.C., Chen, P.C. (2014). Effects of the meetings-flow approach on quality teamwork in the training of software capstone projects, *IEEE Transactions on Education*, 57(3), 201-208
9. De Dreu, C.K., & Weingart, L.R. (2003). Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. *Journal of applied Psychology*, 88(4), 741.
10. Ding, D., Yousef, M., & Yue, X. (2017). A case study for teaching students agile and scrum in Capstone course. *Journal of Computing Sciences in Colleges*, 32(5), 95-101.
11. Hoda, R., Noble, J., & Marshall, S. (2012). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, 39(3), 422-444
12. Humphrey-Murto, S., & de Wit, M. (2019). The Delphi method—more research please. *Journal of clinical epidemiology*, 106, 136-139.
13. Kamthan, P. (2021). On the nature of collaborations in agile software engineering course projects. In *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming* (pp. 1529-1551). IGI Global.
14. Kudaravalli, S., Faraj, S., & Johnson, S.L. (2017). A Configural Approach to Coordinating Expertise in Software Development Teams. *MIS quarterly*, 41(1).
15. Lee, J.C., Chen, C.Y. (2019b). Investigating the environmental antecedents of organizations' intention to adopt agile software development. *Journal of Enterprise Information Management*, 32(5), 869-886
16. Lee, J.C., Wang, Y.T., Chen, C.Y. (2020). The effect of transactive memory systems on process tailoring in software projects: the moderating role of task conflict and shared temporal cognitions. *Journal of Systems and Software*, 164, 110545
17. Lee, J.C., Chou, I.C., Chen, C.Y. (2021). The effect of process tailoring on software project performance: the role of team absorptive capacity and its knowledge-based enablers. *Information Systems Journal* (accept)
18. Liu, J.Y.C., Chen, H.G., Chen, C.C., & Sheu, T. S. (2011). Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management*, 29(5), 547-556.
19. Liang, T.P., Jiang, J. , Klein, G.S. , & Liu, Y.C. . (2010). Software quality as influenced by informational diversity, task conflict, and learning in project teams. *IEEE Transactions on Engineering Management*, 57(3), 477-487.
20. Lous, P., Tell, P., Michelsen, C.B., Dittrich, Y., & Ebdrup, A. (2018). From Scrum to Agile: a journey to tackle the challenges of distributed development in an Agile team. In *Proceedings of the 2018 International Conference on Software and System Process* (pp. 11-20).

21. Mahnic, V. (2011). A capstone course on agile software development using scrum. *IEEE Transactions on Education*, 55(1), 99-106.
22. Mohammed, S., & Nadkarni, S. (2011). Time-based individual differences and team performance: The moderating role of temporal leadership. *Academy of Management Journal*, 54, 489–508.
23. Mohammed, S., & Nadkarni, S. (2014). Are we all on the same temporal page? The moderating effects of temporal team cognition on the polychronicity diversity–team performance relationship. *Journal of Applied Psychology*, 99(3), 404–422.
24. Parker, R. M. (1993). Threats to the validity of research. *Rehabilitation Counseling Bulletin*.
25. Parker, D.W., Holesgrove, M., & Pathak, R. (2015). Improving productivity with self-organised teams and agile leadership. *International Journal of Productivity and Performance*. 64(1), 112-128.
26. Popaitoon, S. and Siengthai, S. (2014). The moderating effect of human resource management practices on the relationship between knowledge absorptive capacity and project performance in project-oriented companies. *International Journal of Project Management*, 32(6), 908-920.
27. Rodríguez, G., Soria, Á., & Campo, M. (2016). Measuring the impact of agile coaching on students' performance. *IEEE Transactions on Education*, 59(3), 202-209.
28. Santos, C. M., Passos, A. M., Uitdewilligen, S., & Nübold, A. (2016). Shared temporal cognitions as substitute for temporal leadership: An analysis of their effects on temporal conflict and team performance. *The Leadership Quarterly*, 27(4), 574–587.
29. Sedelmaier, Y., & Landes, D. (2020). Analyzing Challenges in Software Engineering Capstone Projects. *ICSEA 2020*, 145.
30. Skulmoski, G.J., Hartman, F.T., & Krahn, J. (2007). The Delphi method for graduate research. *Journal of Information Technology Education: Research*, 6(1), 1-21.
31. Slack, M. K., & Draugalis Jr, J. R. (2001). Establishing the internal and external validity of experimental studies. *American Journal of Health-System Pharmacy*, 58(22), 2173-2181.
32. Standifer, R. L., Raes, A. M. L., Peus, C., Passos, A. M., Santos, C. M., & Weisweiler, S. (2015). Time in teams: cognitions, conflict and team satisfaction. *Journal of Managerial Psychology*, 30(6), 692–708.
33. Torrecilla-Salinas, C. J., De Troyer, O., Escalona, M. J., & Mejías, M. (2019). A Delphi-based expert judgment method applied to the validation of a mature Agile framework for Web development projects. *Information Technology and Management*, 20(1), 9-40.
34. Tsai, W.L., Chen, C.S., Chen, C.Y. (2018). Snowman: Agile development method with institutionalized communication and documentation for capstone projects. *Asia Pacific Management Review*, 23(1), 12-19.
35. van Woerkom, M., & Van Engen, M. L. (2009). Learning from conflicts? The relations between task and relationship conflicts, team learning and team performance. *European Journal of Work and Organizational Psychology*, 18(4), 381-404.
36. Williams, L., & Cockburn, A. (2003). Agile software development: it's about feedback and change. *IEEE computer*, 36(6), 39-43.
37. Xu, P., & Ramesh, B. (2007). Software process tailoring: an empirical investigation. *Journal of Management Information Systems*, 24(2), 293-328.

### 三. 附件 Appendix (請勿超過 10 頁)

#### (1) 產品特色量表

專題成果之產品特色量表		填答日期____月____日					
欲評分之組別 (飛鷹組、日曆組、掃蕩組): _____							
請依『同意的程度』勾選 1~7 的整數。							
項 目	1	2	3	4	5	6	7
1.該產品所使用的技術可以襯托出其基於傳統的理论基礎或技術發展而成的。							
2.該產品可以展現出符合該產品主題的特色功能。							
3.該產品的功能符合其產品主題之服務範圍 (例:手機 APP 功能影響到手機的核心功能)。							
4.該產品易於操作使用。							
5.該產品之功能不易被取代。							
6.該產品突顯出其他產品針對共同問題所使用之解決方法的缺點。							
7.該產品之功能讓使用者可預測產品未來發展的趨勢。							
8.該產品使用的解決方案為現有與自行開發的技術所結合而成。							
9.該產品是利用現有的解決方法為基礎開發而成。							
10.該產品以現有的知識於新的應用上使用。							
11.該產品使用前所未見的技术供使用者體驗。							
12.該產品為開發者使用既有的技術利用在不同的應用上。							
13.該產品發展出嶄新的技術基礎並指引出未來可能的解決方案。							
14.該產品有些新創功能為這些功能的未來發展提供了好的基礎。							
15.該產品用嶄新的角度來詮釋欲解決的現有問題。							
16.該產品擁有的技術或功能可作為區隔新舊技術的標準。							
17.該產品有些功能賦予了新的使用概念。							

(2) 課程活動照片



計畫前期專題會議：德菲法教學現場



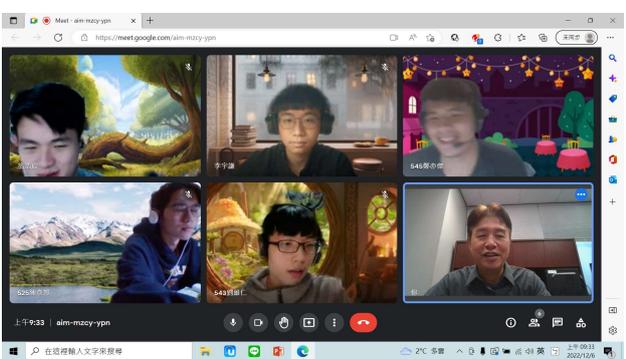
計畫前期專題會議教學：Spring review meeting



遠距會議：Spring review 會議



遠距會議：Spring planning 會議



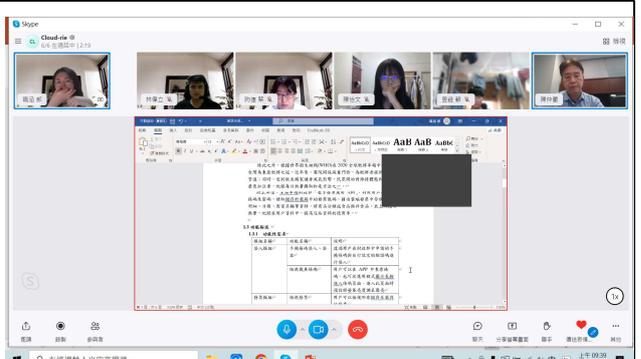
遠距會議：Spring planning 會議



專題隊伍獲獎照片



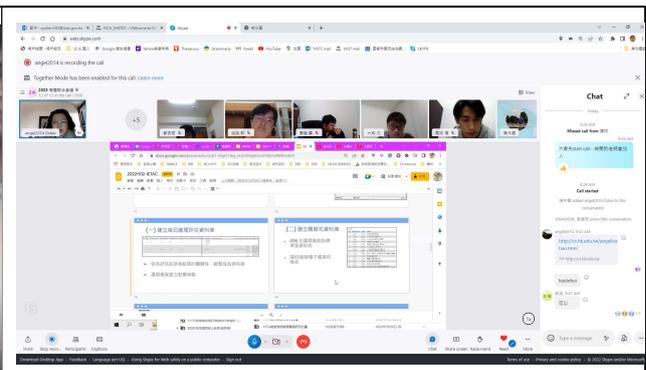
遠距會議：Spring retrospective 會議



遠距會議：與 PO 進行規劃與檢討



專題會議：Sprint review meeting



專聯合會議：Milestone 檢討