

私立東海大學資訊工程與科學研究所

碩士論文

指導教授：周忠信

軟體生命週期管理的快速導入方法

Rapid Modeling of Software Lifecycle Management

研究生：劉駿馨



中華民國九十七年十月二十二日

摘要

在軟體工程領域中，軟體生命週期管理是一個相當重要的議題。然而對於一般軟體團隊，特別是中小型的團隊，導入軟體生命週期管理並不是一件容易的事。而要能夠進一步套用相關工具來落實軟體生命週期的管理工作，則更是一個困難的挑戰。本論文首先提出一個快速軟體生命週期管理的導入方法 – RALM，RALM 主要目的在於能夠幫助中小型的軟體團隊，快速定義其所需的軟體生命週期管理工作。而透過 RALM 所產出的 ALM-XML 定義，可進一步被用來幫助導入軟體生命週期的管理工具。在本論文中，我們以 Microsoft 的 VSTS (Visual Studio 2005 Team System) 工具為例，實際發展出 ALM2VSTS 的自動轉換工具，可將軟體生命週期管理定義 ALM-XML 自動轉換成「VSTS 流程範本 (process template)」、「VSTS 系統設定指引 (configuration guidance)」、以及「工程指引網站 (engineering practice portal)」。透過本論文所提出的 RALM 與 ALM2VSTS，軟體團隊可以以最短的時間，透過 VSTS 有效實踐軟體生命週期的管理工作。

關鍵字：軟體工程、軟體生命週期管理、RALM、VSTS

Abstract

In the domain of software engineering, software lifecycle management is an important issue to study. However, it is not easy for a software development team, especially for those medium and small ones, to build an adequate lifecycle management solution. It is also not a trivial job to implement the solution by just adopting commercial or open source available management tools. Therefore, in this thesis we propose a rapid software lifecycle management modeling and implementing method called RALM (Rapid Application Lifecycle management Modeling) to help software development teams implementing their required management solutions as easily as possible. Based on templates and ALM-XML standard format, a software team can rapidly define its lifecycle management solution. In order to show the superiority of RALM, we further implement a tool called ALM2VSTS such that it can automatically transfer the solution definition into Microsoft ® Visual Studio 2005 Team System (VSTS) process template, configuration guidance, and engineering practice portal.

Keyword: software engineering, software lifecycle management, RALM, VSTS

目錄

摘要	i
Abstract	ii
目錄	iii
表目錄	vi
圖目錄	vii
第一章 研究動機與目的	1
1.1 研究動機	1
1.1.1 導入軟體生命週期管理的難度	2
1.1.2 導入軟體生命週期管理工具的難度	2
1.2 研究目的	3
1.3 論文架構	4
第二章 軟體生命週期管理	5
2.1 軟體模型	5
2.2 軟體生命週期	7
2.3 軟體開發的關鍵	9
2.3.1 團隊	9
2.3.2 技術	10
2.3.3 流程	10
2.3.4 管理	13
2.3.5 紀律	13
2.4 軟體生命週期管理	14
2.4.1 需求管理	15

2.4.2	變更管理	17
2.4.3	議題追蹤管理	18
2.4.4	派工管理	19
2.4.5	測試管理	20
2.4.6	建置與部署管理	21
2.4.7	建構管理	22
2.4.8	專案管理	23
2.5	Visual Studio Team System	25
2.5.1	VSTS 概觀	25
2.5.2	VSTS 的架構	26
2.5.3	VSTS 功能介紹	28
第三章 軟體生命週期管理快速導入方法		34
3.1	RALM	34
3.2	ALM-XML	35
3.3	ALM2VSTS 工具實作	43
3.3.1	VSTS 流程範本	49
3.3.2	工程指引網站	51
3.3.3	VSTS 系統設定指引	52
第四章 案例實作		53
4.1	案例說明	53
4.2	案例實作結果	56
第五章 論文結論與未來研究方向		59
5.1	結論	59
5.2	未來研究方向	59

參考文獻	60
附錄一：ALM-XML的 XML Schema 定義	64
附錄二：工程指引網站基本架構的 HTML文件	70
附錄三：案例-ALM-XML.....	72
附錄四：案例-ALM2VSTS 的 VSTS 流程範本主要產出	94

表目錄

表 2.1 需求管理的活動.....	16
表 2.2 變更管理的活動.....	17
表 2.3 議題追蹤管理的活動.....	19
表 2.4 派工管理的活動.....	20
表 2.5 測試管理的活動.....	21
表 2.6 建置管理的活動.....	21
表 2.7 部署管理的活動.....	22
表 2.8 建構管理的活動.....	22
表 2.9 專案管理的活動.....	24
表 3.1 軟體生命週期管理定義樣板之專案定義說明	36
表 3.2 軟體生命週期管理定義樣板之產出物類型定義說明	37
表 3.3 軟體生命週期管理定義樣板之角色定義說明	38
表 3.4 軟體生命週期管理定義樣板之組織定義說明	39
表 3.5 軟體生命週期管理定義樣板之流程定義說明	39
表 3.6 軟體生命週期管理定義樣板之配置定義說明	40
表 3.7 軟體生命週期管理定義樣板之授權定義說明	41
表 3.8 軟體生命週期管理定義樣板之派工定義說明	42
表 3.9 軟體生命週期管理定義樣板之管理指標定義說明	43
表 3.10 軟體生命週期管理定義樣板之議題定義說明	43
表 3.11 MSF for Agile Software Development v4.0 的資料夾內容	51

圖目錄

圖 2.1 S-system.....	5
圖 2.2 P-system.....	6
圖 2.3 E-system	6
圖 2.4 軟體生命週期.....	7
圖 2.5 軟體生命週期管理.....	15
圖 2.6 Visual Studio Team System 功能與工具關聯圖	29
圖 3.1 RALM 流程	34
圖 3.2 ALM2VSTS 工具的執行步驟活動圖	45
圖 3.3 ALM2VSTS 工具的類別關聯圖.....	46
圖 3.4 ALM2VSTSGUI 類別.....	46
圖 3.5 ALM2VSTS 類別	47
圖 3.6 ExcelToXml 類別	47
圖 3.7 XmlValidator 類別.....	48
圖 3.8 FileCopy 類別.....	48
圖 3.9 XmlTransform 類別.....	49
圖 3.10 XmlTransformXPath 類別	49
圖 3.11 MSF for Agile Software Development v4.0 目錄結構	50
圖 4.1 MobileProject 專案的配置管理目錄結構.....	54
圖 4.2 上載流程範本	56
圖 4.3 工程指引網站	57
圖 4.4 VSTS 系統設定指引的步驟說明	57
圖 4.5 VSTS 系統設定指引的設定內容	58

第一章 研究動機與目的

1.1 研究動機

軟體生命週期是指軟體從尚未存在前的需求發展過程、到進入實際的軟體開發過程、再到軟體發展完成後的上線應用與維護過程、以及最終被汰換而進入下一個生命週期的過程。軟體生命週期通常可被劃分為需求定義、軟體開發、軟體發行、軟體維護及軟體淘汰等五個階段[4][13][14][28][32][42]。當軟體的需求被提出、定義並經過反覆確認之後，即可開始進入軟體開發的階段；軟體開發階段包含系統分析、系統設計、程式實作與軟體測試四個步驟，在這個階段中，軟體會被逐步建構完成，並且進行許多測試，以確保軟體的品質，當軟體通過所有計劃中的測試，就進入軟體發行階段；在軟體發行階段，如果有軟體錯誤或需求變更的情況發生就進入軟體維護的階段，對軟體進行修改而開始「軟體演化」；軟體修改完成後再次進行軟體發行，軟體發行與軟體維護兩個階段會反覆進行，直至軟體不敷使用或無法演化時，軟體就會被淘汰同時進入下一個新生命週期，或者也可以簡單稱之為「舊案翻新」階段。

在整個軟體生命週期中，影響軟體開發成功的五個關鍵要素為（1）團隊、（2）技術、（3）流程、（4）管理、（5）紀律。團隊由兩個以上的成員所組成，而這些成員各擔任個別的角色，每個角色皆有其關聯性，這些角色如何協調、合作以完成軟體開發是團隊的重點；技術影響軟體的架構，使用不同的技術，建構軟體的方式就會有所差異；流程說明了軟體該如何被開發，以及每位團隊成員必須要怎麼做及使用哪些工具才能完成他們被交付的工作，並提供軟體團隊一系列可預期的步驟、關聯每項工作的先後順序、執行步驟以及產出物。管理是對軟體開發過程的一切，包括團隊成員、流程、工作、產出物等進行規劃、監控與協調，透過管理達到在計劃的時間與成本之內完成軟體開發，且保證軟體的品質；紀律是團隊成員在軟體開發過程中，對於自身的工作、活動等，皆能符合流程、步驟等基本原則的一種表現，而這些團隊成員要遵循的紀律必須可被隨時取得參考。而以上各工作的落實，則有賴於落實「軟體生命週期管理」（Software Lifecycle Management, SLM）。

1.1.1 導入軟體生命週期管理的難度

前面已經討論了軟體生命週期管理的重要性，在這裡再次深入說明為何需要軟體生命週期管理的各種原因：

- (1) 為避免產出物混亂的情形發生，軟體生命週期的各種產出物，其關係人、配置、權限、產出物格式以及產出物的演化過程，都必須加以定義與管理。
- (2) 軟體生命週期各角色所擔任的工作必須定義與劃分，而各工作的先後順序、執行限制等流程關係須加以管理，以利軟體開發順利進行。
- (3) 軟體生命週期所發生的議題，其處理過程要加以監控並確保完成，以免造成更大的風險發生。
- (4) 軟體開發過程的健康必須被監控，所以開發過程的各項資料要加以收集與分析。
- (5) 確保軟體開發朝著正確目標發展，並制定基本紀律供團隊成員參考，引導團隊成員遵循一定的準則與紀律。
- (6) 為增進工作分派的精確度與團隊合作，團隊的組織與結構需詳加規劃。

由前面的說明，可以知道軟體生命週期管理是必要且重要的，但軟體的複雜性與易變性，導致軟體生命週期管理相對地複雜，這樣的複雜性使得軟體生命週期管理變得困難；而在導入軟體生命週期管理時，這樣的複雜性可能使得軟體團隊的導入需求不明確，且不清楚導入軟體生命週期管理需要了解、定義哪些東西，更增加導入軟體生命週期管理的難度。

1.1.2 導入軟體生命週期管理工具的難度

要實際落實軟體生命週期管理，通常更需要軟體生命週期管理工具來支援管理工作的進行。軟體生命週期管理工具主要可以協助以下工作：

- (1) 提供工作指派、議題發出，並記錄保存其處理過程與相關資料。
- (2) 提供狀態追蹤，確保工作的執行與議題的處理。

- (3) 提供產出物的管理，包括權限管控、版本歷程與產出物格式的提供。
- (4) 提供管理所需的資料記錄收集、分析並製成報表。
- (5) 提供可供團隊成員隨時查閱團隊的角色、流程、工作等各項定義與活動準則的工程規範。

導入軟體生命週期管理工具之前，需要先行定義軟體生命週期的管理需求，如果需求定義不清楚，工具的導入就有困難；另一方面，大部分軟體生命週期管理工具的導入與運用，通常無法淺顯易懂、輕易上手。因此，要將軟體生命週期管理需求的定義再轉成工具來落實，更是一件困難的挑戰。

1.2 研究目的

本研究提出一個可以協助軟體團隊簡易且快速的軟體生命週期管理導入方法，此方法稱之為「RALM (Rapid Application Lifecycle Management)」。同時，為驗證本方法可以與軟體生命週期管理工具結合，本論文選擇 Microsoft® Visual Studio 2005 Team System (VSTS) 軟體生命週期管理工具[41]，做為案例驗證用。

RALM 主要是由多個「軟體生命週期管理定義樣板」所組成。這些定義樣板使用 XML 文件格式儲存，並以 XML Schema 做為定義與驗證；所有相關 XML 檔案的定義，稱之為 ALM-XML。在本論文中，另外配合 VSTS 發展出 ALM2VSTS 工具，可將 ALM-XML 自動轉換產出「VSTS 流程範本」(process template)、「VSTS 系統設定指引」(configuration guidance)、以及「工程指引網站」(engineering practice portal)。其中的 VSTS 流程定義是套用 VSTS 隨附的流程範本- MSF for Agile Software Development v4.0，進一步修改而得。

本研究可以解決以下幾個問題：

- (1) 使用軟體生命週期管理定義樣板，引導出軟體團隊對於導入軟體生命週期管理的需求，使其更了解導入軟體生命週期管理的要素。
- (2) 減低導入軟體生命週期管理的複雜性，使導入軟體生命週期管理變得簡易且快速。

- (3) 軟體生命週期管理工具設定指引，可以降低導入軟體生命週期管理工具的困難度。
- (4) 工程指引提供軟體團隊的共同活動準則，以作為專案成員認知自己的工作職責、上下游關係、以及必須遵循的紀律。

1.3 論文架構

本論文第二章將介紹軟體生命週期管理及 Microsoft® Visual Studio 2005 Team System；第三章介紹 RALM 方法、ALM-XML 與 ALM2VSTS；第四章為 VSTS 的實際案例；第五章為本研究的結論。

第二章 軟體生命週期管理

2.1 軟體模型

當一個軟體開始發展，通常會被評估將來是否改變，再決定軟體的架構應該如何設計才能因應軟體的改變，讓軟體易於維護。但如何知道軟體是否容易改變，改變的程度又有多少，Lehman 根據這些問題，提出三種軟體模型：S-system、P-system 與 E-system[24]。Lehman 認為，一個軟體與真實世界的環境愈有關聯就愈容易改變，且軟體會改變與環境因素息息相關；Lehman 分類軟體模型的方式即是建立在軟體與環境的關聯度及差異程度的基礎之上。以下為 S-system、P-system 與 E-system 三種軟體模型的特性。

在真實世界的環境中，如果一個問題的定義清楚明確，而解決這個問題的正確方法可能有一至多種，並且在軟體的實作呈現上都可行，這類軟體就屬於 S-system。S-system 為靜態且最不易改變的模型，由於問題的定義明確，需求規格可以輕易地訂定出來，之後再依照需求規格將軟體實作出來即可，軟體實作完成後，可直接將問題的執行結果和真實世界的環境做比較，以驗證軟體是否做對，這樣的關聯如圖 2.1 所示。一般來說，如果 S-system 的系統按照需求規格開發但結果錯誤，都是因為對於問題的描述錯誤，進而造成解決方法與軟體錯誤，所以要修正錯誤只要針對其問題描述更正，再根據更正後的問題訂定需求規格並開發新軟體即可。

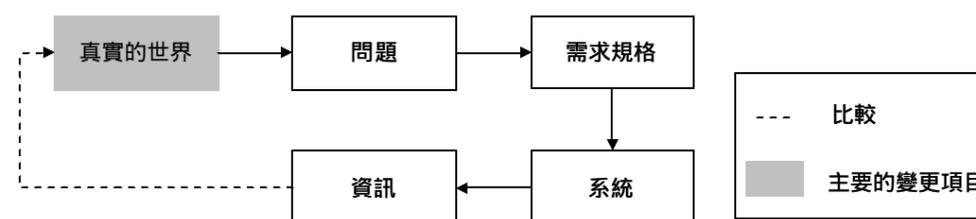


圖 2.1 S-system

P-system 與 S-system 在特性上頗為相似，P-system 的問題與解決方法在真實世界中的定義也非常清楚明確，與 S-system 不同的是，P-system 的問題解決方法在軟體實作上不可行，換句話說，也就是以目前的技術而言，沒辦法以軟體實現問題解決的方法。對於這類的問題，首先會將解決方法簡化，也就是轉

換為抽象模型，使其在軟體實作上可行，需求規格的制定便根據這個抽象模型而來，而軟體所執行的結果通常都是「近似」正確解答的結果，如圖 2.2。抽象模型的解決方法要如何設計，會根據分析者的思維與演算法而有所不同，對一個 P-system 的問題來說，更好的解法抽象模型可能會不斷的被發展出來，當抽象模型變更之後，需求規格與軟體都會跟著被修改。在 P-system 的系統中，如果解答並非滿意的結果，針對相同的問題，只要修改問題的抽象模型軟體實作部份即可。與 S-system 相較，P-system 是更動態的模型，針對同一解決方法，可能有多種抽象模型。

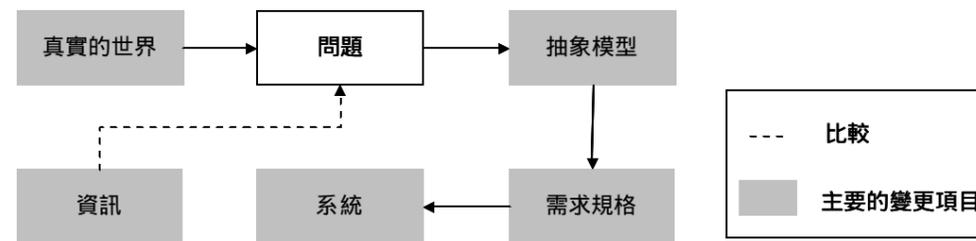


圖 2.2 P-system

S-system 與 P-system 的問題與其解決方法都有明確的定義，但第三種軟體模型，也就是 E-system，這類的問題與解決方法都沒有一個詳細且明確的定義，E-system 就像真實世界的一部份，會隨著環境一起變動，如圖 2.3 所示。E-system 的問題及解決方法都是有可能會變動的，例如企業的營運方式，會根據變動的市場趨勢而有不同的策略，對於相同的市場趨勢，策略可能就有好幾種，策略本身也會不停的修正、變動，當策略變動，支援營運的軟體系統就必須跟著改變。

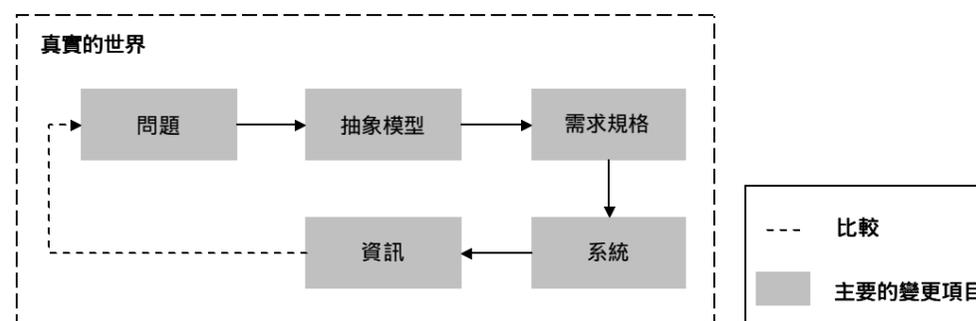


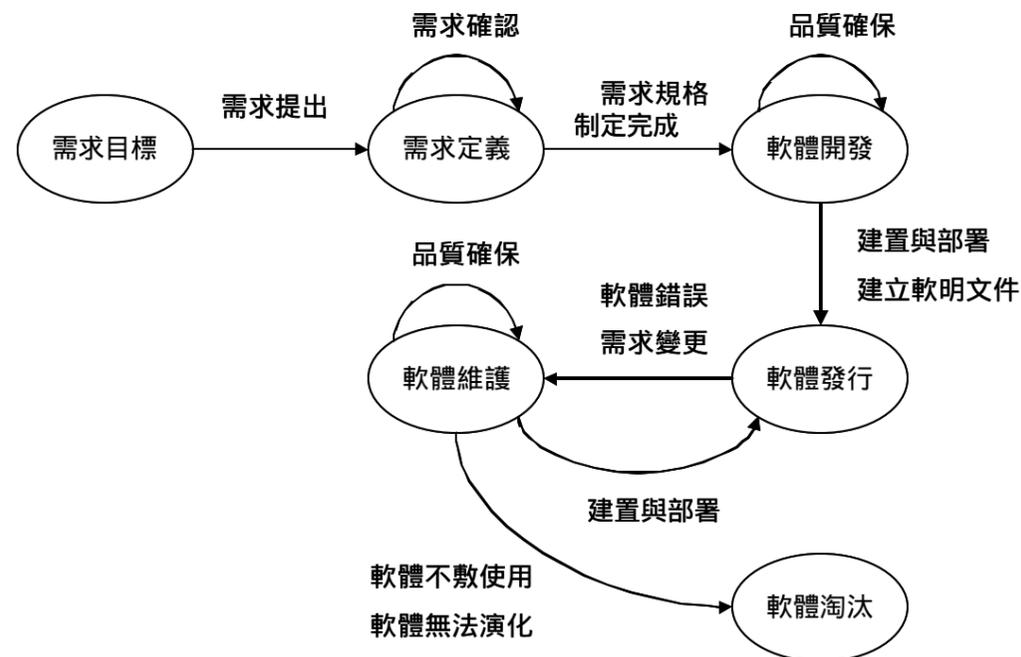
圖 2.3 E-system

由前面的介紹不難發現，E-system 是一種經常改變的軟體模型，而目前在

企業中大部份的應用軟體正是屬於 E-system，隨著企業的成長、營運而改變，正因為容易改變的因素，使得 E-system 在維度與發展上非常麻煩。

2.2 軟體生命週期

軟體生命週期是指軟體從尚未存在前的需求發展過程、到進入實際的軟體開發過程、再到軟體發展完成後的上線應用與維護過程、以及最終被汰換而進入下一個生命週期的過程[4][13][14][28][32][42]。軟體生命週期通常可被分為以下五個階段：需求定義、軟體開發、軟體發行、軟體維護及軟體淘汰，如圖 2.4。



軟體的生產，通常都是由人們的需求與想法所觸發而來，這樣的一個需求與想法可能是為了某個目標，當它被提出並決定開發軟體後，就進入了需求定義的階段。在與關係人（stakeholder）進行需求定義之前，通常開發團隊必須預先進行「領域分析（domain analysis）」，領域分析主要目的是讓開發團隊了解要開發軟體有哪些知識是必須的，這些知識就是所謂的「領域知識（domain knowledge）」，領域知識的了解，可以有效減少開發團隊與關係人之間的鴻溝（gap）。接下來關係人與開發團隊進行訪談，首先要做的就是需求提取與分析

的工作，關係人提出需求、開發團隊記錄並進一步了解需求內容，在進行需求定義的過程中，需求規格書會被一步步的構築完成，一般來說，需求可分為功能性需求及非功能性需求：

- (1) 功能性需求：用來描述軟體必須提供什麼樣的服務，且使用者使用這個軟體的時候，會有怎麼樣的回應。例如，一個 ATM 必須具備餘額查詢、提領現金、修改密碼和跨行轉帳等的服務，這些都是功能性的需求。
- (2) 非功能性需求：這一類的需求通常和具體表現出來的功能沒有直接關係。以 ATM 的例子來說，回應使用者的時間必須在 1 秒之內，操作時必須有好用的介面等，這些都是非功能性需求的例子。

在需求定義時，確定軟體的範圍（scope）也很重要，了解範圍才不會讓軟體開發的目標發散、模糊且變得不確定。因為需求定義階段是確認軟體「作對」的階段，所以需求提取完畢後必須以嚴格的程序對需求規格書的內容進行確認，確定制定的需求符合使用者的想法，確認需求規格書的內容無誤，即可開始軟體開發的階段。

軟體開發是軟體生命週期中，人力需求最多的階段，它的步驟包含系統分析、系統設計、程式實作與軟體測試。在分析步驟中，首先必須了解需求規格書的內容，也就是使用者所「想要」的是什麼，分析必須根據需求，定義「what」的部分，它主要呈現軟體的整體描繪，也就是這個軟體的各項功能、相關角色、所應達成任務、人機介面以及操作流程等。系統設計則根據系統分析結果，實際設計系統「how」的部分。所謂 how 的部份是指資料模型、物件規格、物件互動關係、演算法與步驟等。系統設計完成後，就可開始程式實作的步驟，這個步驟的主要工作就是依照系統設計的內容撰寫程式，每個完成的程式必須進行單元測試，以確認程式是否符合設計，如果單元測試失敗，就進行程式除錯的工作，並再次進行單元測試，直至程式通過測試。軟體測試階段主要是做程式糾錯的工作，這是一個確認軟體「作好」的階段。要做的測試工作有整合測試、壓力測試、效能測試、系統測試、安全測試等，以某種角度來看，軟體測試人員是以挑戰開發團隊的心態來測試軟體，他們必須想盡各種可

能造成軟體失效的方式來測試軟體。當前述的測試都通過之後，就必須進行使用者接受度測試，這項測試由使用者或是使用者代表來進行；測試的最大目的就是提升使用者對產品的信心以及滿意度。所有測試完成之後即可進入軟體發行的階段。

在軟體發行階段，除了開發完成的軟體產品之外，還必須有一份軟體說明文件被產生出來，這份文件的內容必須包含軟體的安裝與部署、使用說明、注意事項等，讓使用者可以順利使用軟體。至此，軟體就真正開始被使用了，開發團隊可能必須提供軟體的訓練與支援服務，並且建立與使用者之間的溝通管道，以利接受使用者回饋，這些回饋可能包含軟體的錯誤提報與建議等。

軟體發行之後，可能還是會有錯誤被使用者發現，或是關係人提出需求變更等情況，此時就進入軟體維護階段，在這裡必須依照各種情況進行評估、對軟體進行修改，並再次進行軟體測試，測試通過即可再次發行，軟體就有可能產生新版本。

雖然完成的軟體沒有和硬體一樣有隨著時間消逝而日漸毀壞的情況發生，但隨著使用者的想法與需求變更，軟體可能變得不敷使用，而修改舊軟體所花的成本高於重新開發新軟體時，舊軟體就會面臨被汰除的情況，被淘汰的舊軟體就結束了他的生命週期，接下來取代的新軟體就是另一個生命週期的開始。

2.3 軟體開發的關鍵

本節探討軟體開發五個不可或缺的關鍵要素：（1）團隊；（2）技術；（3）流程；（4）管理；（5）紀律。現今軟體開發大都需要由一個團隊來執行，軟體開發加上團隊成員，就可構成一個軟體專案，在軟體開發的過程中，小組裡的各種角色各自需要使用不同技術，來執行他們所負責的工作，這些工作之間或有先後順序、或有執行限制等流程關係；管理是對軟體生命週期的一切進行規劃、監控與協調的工作，雖然管理是軟體開發成功的一大要素，但團隊成員的個人紀律卻是最根本的因素。以下就這五個要素一一探討。

2.3.1 團隊

一個團隊組成的基本要素就是「人」，這些人所指的是軟體生命週期進行

過程中所有的角色與關係人，一般來說，參與軟體生命週期的角色可能有專案經理、系統分析師、系統設計師、程式設計師與測試人員，甚至顧客也算是一員。就像各種細胞漸漸成長發育我們的身體一般，這些團隊成員也各司其職、一步步地孕育出欲開發的軟體。團隊成員之間的溝通、協調與合作非常重要，良好的團隊溝通可以免除許多不必要的工作，促進軟體的品質；由於現今的軟體開發規模日益龐大、複雜，相對的開發團隊的成員也會增多，團隊人數愈多，溝通的效果就愈難保持，所以團隊溝通必須透過管理手段來加以維持。團隊的組織方式也影響著軟體開發的品質，Mills 提出的「外科手術團隊」就是一個可以維持軟體整體性並大量減少團隊溝通必須付出的成本的團隊組織方式，Mills 建議大型軟體可被切割許多小部分，並分別交由一個團隊負責，而每個團隊中由一個人扮演設計的角色，其他人則是扮演支援的角色。

2.3.2 技術

技術關係著軟體的架構與整體風貌，在軟體開發初期，評估採用何種技術實現軟體非常重要，不同的技術之間對於軟體的開發方式會有所差異；當技術決定之後，團隊成員即可預期他們必須產出哪些結果，也決定了軟體開發中的工作必須要如何做，需求制定、系統分析、系統設計、程式實作、軟體測試，甚至是文件的編寫對於不同的軟體技術可能都會有不同的做法。通常技術都有工具支援，以增進軟體開發的工作速度與準確性。軟體技術與各種工具不斷創新，其開發、產出、建置與部署等方式也不盡相同，這對於軟體團隊是一個挑戰。

2.3.3 流程

就像一般工廠有其生產流程，軟體生命週期也有「流程」的概念，軟體的開發步驟在流程中會有其規則順序，遵循流程會使軟體系統開發得更快速。流程就像路標一樣，提供開發團隊一系列可預期的步驟，流程的步驟定義可以使軟體開發的活動具有穩定性且可被控制[12]。流程是透過「流程定義」來生成，並使用「流程模塑」來表達流程，接下來介紹這兩個部份。

流程定義說明了團隊成員的角色定位，並讓每個人了解他們的工作為何、

表現每個人在整個組織的溝通以及工作關聯，它也包含可被用來管理、品質保證、建構管理、稽核與開發的資訊；流程定義的最終目標就是要增進團隊的生產力以及軟體產品的品質[16]。

綜觀各個研究，流程定義可以從軟體生命週期的各個觀點切入探討[3][5][6][10][15][21][22][31]：

- (1) 職務 (functional)：描述流程中有什麼「工作」需要被執行。在各個研究中，activity、task 與 process step 都是代表流程中的一項基本「工作」。
- (2) 行為 (behavioral)：描述流程中的工作何時被執行，以及如何被執行。何時要被執行可以用 event 以及 behavior 來描述。如何被執行則有 tool 描述工作要用什麼工具執行、context 描述工作在什麼 work space 執行，constraint 描述執行工作的各種限制。
- (3) 組織 (organizational)：描述工作被什麼人所執行。role、actor、agent 都可以被用來描述執行工作的人與物；role 代表一個角色、一個職責，而 actor 則是代表一個實際的人物，也就是專案的成員，一個 role 可以被多個 actor 擔任，相反的，一個 actor 也可以擔任多種 role。agent 除了可以表示一個人之外，也可以代表一個執行工作的軟體。
- (4) 資訊 (information)：描述工作的產出物，各研究所使用的名詞有 artifact 與 resource 兩種。
- (5) 關聯 (relationship)：描述流程中的人、工具、工作與產出物之間的關聯，例如 has、create、use 與 depend 等。

流程可以用 global 與 local 兩個角度來討論[18]，global 的角度可以輕易的串連各個流程步驟，但對於各流程步驟的內容不提供詳細說明，例如執行者、使用工具、限制與產出物等，而 local 則和 global 的觀點相反，僅對流程各部份進行詳細說明，所以這兩者交互連結使用可以更詳盡的描述流程。

綜觀流程定義所帶來的好處有下列七點[3][6][8][12][20][21]：

- (1) 促進團隊的了解與溝通：流程定義可說明一個流程最基本所需的要

素，說明專案團隊成員的角色定位，讓每個人了解他們的工作為何，並表現每個人在整個組織的溝通以及工作關聯；使用流程模塑語言表達流程定義會比僅僅只用文字表達更讓人容易了解，且讓團隊成員有相同的語言可以互相溝通，進而讓團隊成員相互合作更有效率、提供足夠的資訊讓個人或團隊執行工作。

- (2) 流程改善：流程定義是流程改善不可或缺的一部份；正式的流程定義可以識別一個良好的軟體流程所需的要素、捕捉流程中的資訊，促進組織學習有效的軟體流程以支援管理流程的演化。了解、制定與測量軟體流程的最終目的是改善流程並創造更好的軟體品質。當組織的變動，如人員流動、應用領域及技術的轉移，沒有正式流程定義的缺點與問題就會浮現出來；經過連續改善，軟體流程就會產生更有品質的產品，成本也會降低。
- (3) 流程管理：流程定義可被用來預測、監督、排程、管理與協調流程步驟，並支援專案計劃的制定。提供流程測量的基準點，例如，在流程中定義一個測量點（checkpoint）。
- (4) 流程分析、模擬與評估：只有正式的流程定義才有可能被拿來分析。在執行流程改變之前，可先評估有哪些潛在的影響。更可以比較兩個流程、協助流程中工具的選擇；在流程實行之前可以先使用所定義的流程預先模擬，新增或修改不適合的部份。
- (5) 流程指引與執行：流程定義可以指引正在專案團隊成員清楚了解他們所要執行的工作的一切所需資訊。
- (6) 流程重用：在未來的專案中，良好的流程定義可再重用，或再由此流程定義裁適出適合目前專案的流程定義。
- (7) 教育與訓練：正式的流程定義可以讓新加入的專案團隊成員快速了解他們正在使用的流程。

在真實的世界中，排程、成本、契約、需求的限制都會影響流程的生成 [12]。流程定義可以使用文字以及流程塑模語言來描述，由於一般的文字描述

容易產生模糊、不完整與不一致的情況發生，所以使用流程塑模語言定義流程有其必要性，一般會以流程塑模語言為主、文字描述為輔的方式來定義流程。流程塑模語言的分類如下[5][8][23]：

- (1) 圖形化 (Diagrammatic formalism) 的表示方式：使用正式定義的結構化圖形來表示流程，這一類的表示方式易於了解與溝通，IDEF0、IDEF3[17][34]、UML[2][9][40]與 Little-JIL[1]皆是這一類的流程塑模語言。
- (2) 程式語言 (coding language) 的表示方式：這種方式將軟體流程也當成一個軟體，也有和軟體相同的生命週期，使用 programming-like language 來定義流程；這一類的支援流程執行，但較不易於了解與溝通。例如 DRAGOON[16]與 APPL/A[30]。
- (3) Petri Net：使用以數學為基礎的圖形表示方式來表現動態與分散式的軟體流程[37]，這種方式可用輔助的測量工具來分析流程，但較不容易被了解。

流程的導入對於軟體開發是一項重要的工作，軟體流程品質與軟體品質有相當密切的關係，好的流程可以提升軟體品質、降低成本並縮短軟體開發時程。

2.3.4 管理

由於現今的軟體開發規模日益龐大、複雜，所以管理的問題已經超越技術的問題了，「有好的流程與管理，不一定做得出好軟體；但沒有好的流程與管理，一定做不出好軟體」，這一段話顯示流程與管理的重要性，但比起流程，管理又更是重要，透過管理手段可將軟體生命週期各階段的輸入與產出明確關聯，進而使得軟體開發的全部過程得以有效進行。團隊成員是否遵循開發流程，要怎麼做才能降低軟體開發的成本、縮短軟體開發時間並且提升軟體品質，這些都需要透過妥適的「軟體生命週期管理」才能達成；本論文將在第 2.4 節深入討論軟體生命週期管理。

2.3.5 紀律

軟體開發是否成功，最根本的要素是團隊成員的行為，如果團隊成員沒有紀律，其責任心、向心力日漸消散，可能間接造成軟體開發失敗。紀律記述每位團隊成員必須執行與遵守的基本事項。現今的社會潮流使得個人的自主意識高漲，所以傳統的集權管理方式自然漸漸轉變成授權管理的方式，相對地，團隊成員的自我管理也就愈加重要，培養團隊紀律會讓團隊成員提升責任心、向心力、歸屬感，團隊溝通的有效性也會接著提升，軟體品質自然會更好、軟體專案時程與成本也會更容易在控制範圍內。SEI 所提出的人員能力成熟度模型（People Capability Maturity Model, P-CMM）也是由建立團隊成員的基礎紀律開始，進而管理員工紀律，可見得紀律在團隊中的重要性。

2.4 軟體生命週期管理

目前有關軟體生命週期管理的相關研究頗多[4][7][11][19][25][26][32][42]。軟體生命週期管理主要由以下幾個部份所構成[13][28][42]，分別是：

- (1) 需求管理
- (2) 變更管理
- (3) 議題追蹤管理
- (4) 派工管理
- (5) 測試管理
- (6) 建置與部署管理
- (7) 建構管理
- (8) 專案管理

軟體生命週期各階段的管理，是由此八項工作所組成，如圖 2.5 所示。在圖 2.5 中，所有的箭頭方向代表的涵意為資料流動及交換的方向，並非流程步驟的關係。在這裡，我們將啟動軟體系統生命週期管理的觸發事件分為兩大類：「需求」與「上線服務」；「需求」為軟體系統最初始的服務需求，而開始進行軟體開發之後所發生的事件，如需求變更、系統維護等事件，皆歸類為「上線服務」。

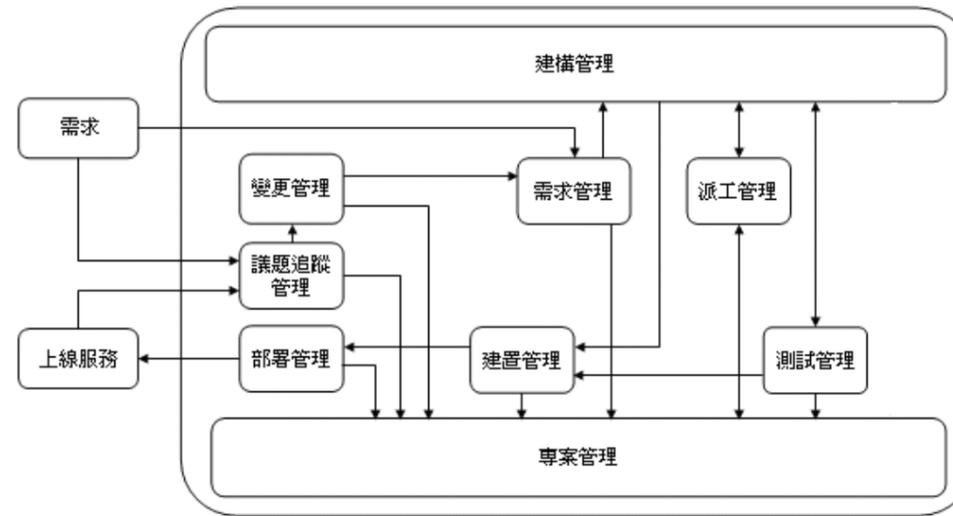


圖 2.5 軟體生命週期管理

2.4.1 需求管理

軟體是需求的呈現，所以需求決定了軟體的全貌，由此觀之，需求是形成一個軟體最重要的要素；需求規格一經確定，開發團隊即可依照需求規格的內容實作出使用者心中所想的軟體。但事實上，大部份的需求很難清楚明確的呈現，這是因為以下的需求本質所造成：

- (1) 抽象；需求是一種想法，沒辦法看見，故必須以各種方式來表達、呈現它的風貌。
- (2) 複雜；現今的軟體大多龐大，不僅軟體本身複雜，和環境之間也有複雜的關聯。
- (3) 具備動態的行為；需求包含靜態與動態行為的需求，其中動態行為很難表達、解釋得清楚。

由於這些本質，致使需求可能不明確、不詳細、不一致，甚至錯誤的情況發生；或者，在需求確認之後，顧客有了新的需求，需求就必須被新增、修改或刪除；這些需求的改變我們稱之為「演化」，透過演化，需求可以更接近我們心中所想的全貌，正因為需求演化的本質，了解如何管控需求演化勢必是重要的一個事項，所以「需求管理」的施行對於一個軟體的發展非常重要。

表 2.1 需求管理的活動

活動名稱	活動說明
需求識別與分類	每一項需求必須能夠被識別，並具唯一性，以利追蹤與參考的進行；依照需求的類型，對需求進行分類。
需求審查	審查需求的可行性，包括是否和其他需求衝突以及實現成本是否在合理範圍等，以決定被提出的需求是否要進行實作。
需求重要性排序	排序需求的重要性，以利實作時可將重要的需求儘早實現；在解決需求之間的衝突時也可提供參考。
解決需求之間的衝突	如果需求之間發生衝突，必須對需求進行取捨。
需求追蹤	記錄需求之間的關聯，此記錄通常透過可追蹤矩陣（traceability matrix）表示，更進一步地，需求和分析之間、分析和設計之間的關聯也可進行記錄。
需求確認	確認目前已定義完成的需求文件內容是否符合顧客所想要的，這是很重要的一個程序，因為需求文件錯誤，將會造成軟體錯誤，在軟體開發階段才發現這些錯誤，可能要花費大量成本才能修正。

表 2.1 表列出需求管理的一般活動[13][38][42]，關於需求變更的部份將在「變更管理」小節討論。需求管理是一個瞭解與管制軟體需求變更的程序，它必須與需求工程程序一起執行。一般來說，需求和需求之間會有某種程度的關聯性。在眾多的需求之中，難免會有相互衝突的需求，對需求必須有所取捨；而哪些需求已被實做出來，哪些還沒審核，都是需求管理要注意的。需求管理包含需求分類、需求審查、需求衝突解決、需求重要性排序、需求追蹤及需求檢查等作業。當需求被提出後，需求必須被加入列表中並儲存起來，將這些需求有條理的分門別類，以利管理，當新的需求被提出時須經過審查，以確定此需求是否可行。需求之間的關聯性及重要性等級必須被記錄，因為需求之間可能會互相衝突，即可依需求之間的關係與重要性程式決定如何解決衝突，或對需求進行取捨；而在進行軟體實作時，重要性較高的需求也可被較早實作出

來，在需求被處理的過程中，各個需求的處理狀態必須被記錄。需求之間互相關聯性，變動一個需求後，可經由需求追蹤找出哪些需求可能需要跟著變動，或是哪些對應的設計與變動的需求相關聯，如此一來可避免混亂的發生。由於軟體規格是由需求所分析、設計出來，所以需求的變化和軟體規格也有關聯。當一個需求改變時，就必須找出有哪一些軟體分析與設計受變動的需求影響，進而確保這些分析與設計會被檢驗。當分析與設計必須變動時，其所對應的程式也必須跟著修訂。唯有透過管理需求、分析和設計之間的關聯，軟體的開發才能夠確保順暢無礙。

2.4.2 變更管理

在軟體模型一節中，我們討論到了在現今的企業軟體中，大部份皆屬於會隨著環境演化的「E-system」，而在演化的過程中，如何避免混亂的發生，並且保持軟體開發順利進行，就必須透過「變更管理」與「建構管理」的施行。表 2.2 為變更管理的一般活動[13][38][42]，在這裡我們討論變更管理，建構管理留待後面章節討論。

表 2.2 變更管理的活動

活動名稱	活動說明
變更審核	審核提出的變更是否合法、變更的影響範圍以及變更的成本預估。
變更決定	審核通過之後，決定變更的實作。
變更記錄	變更的歷程必須被記錄下來，以利追蹤以及稽核的進行。

軟體生命週期中，隨著使用時間的演變，軟體可能會因為錯誤、需求新增或變更而必須修改，這些修改的影響則可能涵蓋需求、設計或程式。當變更發生時，成本與變動影響範圍的評估，是衡量同意變更與否的關鍵因素，而這正是變更管理的首要任務。變更管理首先必須設計一套變更管理程序，以確保所有的變更都可以在控制中進行，一般的變更管理程序如下：

- (1) 提出變更；可能使用變更要求單（change request form, CRF）提出，CRF 除了要求變更的項目之外，也記錄了日期、關係人、處理狀態等

資訊。

- (2) 分析變更要求；審核 CRF 的內容，以決定變更是否有效，如果無效必須回報提出變更要求者。如變更要求有效就進行下一步驟。
- (3) 評估變更；要評估的部份有變更的範圍以及成本等，評估完成後就送交給變更管制委員會（change control board, CCB），讓 CCB 決定是否接受這個變更，如果 CCB 拒絕變更要求，則變更無效並退回提報者。如果 CCB 接受變更要求就進行下一步驟。
- (4) 進行變更；變更後的軟體或文件等，必須被再次進行確認，以確保變更被正確進行。

在變更程序中的所有歷程都應該被記錄下來，以利日後稽核的進行。在軟體生命週期中最常發生變更的項目就是需求。當軟體需求變更時，就必須透過需求變更管理來進行變更的控制；一般來說，新舊需求間通常會有很大的關聯性，此時管理者必須回頭看之前的需求，以了解要變更或新增的需求是否和其他舊有的需求相衝突；而新的需求是否值得採用，或者衡量需求之間的取捨，達到最大的利益，都是需求變更時所要注意的。

2.4.3 議題追蹤管理

在軟體的生命週期中，系統所發生的任何事件，例如程式錯誤、規格錯誤、需求新增或變更、資源調度等，皆需要被有效追蹤與管理，進而確認問題是否獲得解決，議題追蹤管理，主要即在負責追蹤、紀錄與管理所有事件的相關資訊。不同類型的議題發起者與處理流程也不相同，所以議題追蹤管理首先要做的便是議題分類，使議題追蹤過程不會產生混亂；當議題被提出後，必須經過管理者的審核，確定其重要性以及它所造成的風險，並及早進行應對措施，減低議題造成的負面影響；當議題被處理的過程中，它必須可以被追蹤，包括其發起者、目前處理者、進度與狀態等資料。表 2.3 是議題追蹤管理的活動說明[42]。

表 2.3 議題追蹤管理的活動

活動名稱	活動說明
議題識別與分類	每一項議題必須可被識別，並具唯一性，以利追蹤；依照議題的類型對議題進行分類。
議題審核	審核議題對軟體開發所造成的影響，以及其重要性與風險，再以這些資訊對議題訂定重要性。
議題記錄	每個議題的過程都要被記錄下來，包括日期、關係人、狀態等。
議題追蹤	議題從發起至關閉的處理過程，甚至是關閉之後，都必須被可被追蹤。

2.4.4 派工管理

如表 2.4，派工管理主要活動是工作排程、工作分派及工作進度的監控與追蹤[13][42]。首先工作排程首先必須將工作進行分解，也就是將所有工作分解成幾個獨立的細項工作，並為這些工作排定優先權、估算完成工作所需的時間以及所需的資源；在軟體專案的初期，工作排程的計畫會先擬訂一個涵蓋整個軟體專案的大範圍排程，這個大範圍的排程會粗略的說明每個階段的完成時間，以及哪些產出必須完成；而在軟體專案進行中，就會針對每個階段進行詳細的排程，在詳細的排程中，每個工作的開始時間、完成時間等，都必須被詳細定義。完成工作排程後，即進行工作分派，根據工作排程的結果將這些工作分配給團隊中適合的成員。

當工作開始進行之後，管理者就對工作的進度進行監控與追蹤，避免工作時程延遲，並適當的提供團隊成員支援，讓工作可以順利完成；工作進行的情況也可當做擬定下一階段排程計畫的參考。關於追蹤工作進度的方式有：

- (1) 透過小組會議，讓成員報告他們的工作進度。
- (2) 檢核專案里程碑是否都在期限內完成。
- (3) 檢核專案中的實際開始工作時間是否和排程的開始工作時間相同。
- (4) 當面和團隊成員會談了解。

表 2.4 派工管理的活動

活動名稱	活動說明
工作排程	進行工作分解、排序工作優先權，並估算完成工作所需的時間以及所需的資源，再參考這些資料進行工作排程。
工作分派	將工作分派給團隊中適合的成員。
工作進度的監控與追蹤	工作進行的狀況必須加以監控與追蹤，避免時程延遲的情形發生。

2.4.5 測試管理

軟體生命週期中必定會經歷多次、多種的測試，測試的目標是為了發現軟體錯誤，而成功的測試就是發現尚未發現的軟體錯誤，經過不斷的測試，可以提高軟體的品質、增加顧客對軟體的信心。測試管理的活動如表 2.5 所示 [13][38][42]。

測試在進行之前必須進行規劃，包括測試的策略、測試的方法、測試案例以及測試時程等都必須預先規劃，並擬定一份測試計畫書，測試計畫書指引軟體開發團隊與測試團隊一個對於測的正確指引，它的內容包含：

- (1) 對於測試程序的描述，包括測試的範圍以及測試完成條件的制訂。
- (2) 對需求追蹤，確保每個需求都被測試到。
- (3) 明確指出哪些軟體執行的產出需要被測試。
- (4) 測試的時程以及資源的配置。
- (5) 測試結果的記錄以及檢查程序。
- (6) 說明執行測試的硬體與軟體需求。
- (7) 影響測試程序的限制條件。

測試管理除了對測試計畫書的內容進行制訂之外，還必須對測試的過程加以監控，稽核測試是否確實執行。

表 2.5 測試管理的活動

活動名稱	活動說明
測試規劃與設計	測試的策略、測試的方法、測試案例以及測試時程等進行規劃與設計。
測試結果記錄	所有的測試結果都必須被記錄下來，並撰寫測試報告書。
測試監控	對測試的過程進行監控，並對測試結果進行稽核。

2.4.6 建置與部署管理

建置是對所有軟體元件進行編譯與連結動作，將它們轉換成可執行在特定目標組態的可執行檔，所以建置管理要做的是決定所需的元件及其版本、所需的資料檔案、使用的編譯器或其他工具等。在軟體生命週期中，同一支程式會有多種版本，一個上線的軟體，必須由正確版本的程式所組成，而此工作乃透過建置管理維護之；軟體部署的主要目的是要讓軟體可以正常上線運作，故部署管理要決定部署環境、決定軟體部署版本，而在軟體交付時，必須提供與軟體相關的資料與文件、軟體的相關支援、與使用者建立溝通與回饋機制等。表 2.6 與表 2.7 是建置與部署管理的活動說明[13][38][42]。

表 2.6 建置管理的活動

活動名稱	活動說明
說明建置的軟體版本	新建置的軟體版本必須被決定，並加入建構管理進行版本控制。
說明建置所需的元件、模組及其版本	決定建置軟體所需的元件及模組，包括其版本也必須指定。
說明建置所需的資料檔案及其版本	決定建置軟體所需的資料檔案，包括其版本也必須指定。
說明建置所需的編譯器或其他軟體工具	建置所需的編譯器或其他軟體工具都要詳細說明其名稱以及版本。

表 2.7 部署管理的活動

活動名稱	活動說明
說明部署環境	軟體的部署環境，包括硬體及軟體環境都必須說明。
指定部署的軟體版本	被部署的軟體版本必須被詳細指定。
產品交付	當軟體產品交付給顧客時，必須提供與軟體相關的資料與文件、軟體的相關支援、建立顧客溝通與回饋機制等

2.4.7 建構管理

建構管理的主要目的是定義如何記錄與處理軟體的變更，並對軟體加以識別、組織與控制的技術；建構管理的活動如表 2.8 所示[13][38][42]。

表 2.8 建構管理的活動

活動名稱	活動說明
建構識別	識別建構項目、制訂基準線與版本號碼格式。
建構變更控制	變更管理以及建構項目的版本控制。
建構稽核	確保建構項目符合即定標準。
建構狀況報告	針對建構項目的進度與變更提供一份完整的資訊，供團隊成員參考。

建構管理的主要任務為以下四項：

- (1) 建構識別 (configuration identification)。
- (2) 建構變更控制 (configuration change control)。
- (3) 建構稽核 (configuration auditing)。
- (4) 建構狀況報告 (configuration reporting accounting)。

建構識別的主要工作是識別建構項目 (software configuration items, SCIs)，也就是識別必須管理並保持其完整性與一致性的各項產出，這些產出包括程式碼、文件及資料等；制訂基準線也是這項任務的主要工作之一，基準線的正式定義為「一種已經過正式審查並同意接受的規格說明或產品，將其做為一個未來開發的基準，並只能在經由正式改變控制程序之下進行改變」，簡單來說，基準線就是受控制的規格說明或產品。對於版本號碼的格式制訂也是

在這個任務之下進行。

在軟體開發過程中，建構項目變更的管理也是建構管理的重點之一，變更的過程以及這些變更與軟體版本的關聯及變更的原因必須要被記錄，並可被追蹤，關於變更的部份，我們已經在「變更管理」小節介紹過了，這些就不詳加討論了；當需求、設計和程式變更時，這些產出物會跟著產生許多不同版本，因應不同版本的產生，測試也要隨之改變，當系統愈加龐大時，版本也就愈複雜。當軟體開發工作被派出之後，都必須對各個階段的產出物加以管理，並有系統的整理對應版本，保持產出物的完整性與一致性。

建構稽核是一個確保建構項目符合即定標準的活動，這裡所指的標準是經過嚴格程序所制訂的基準線（baseline）；建構狀況報告的主要目的是針對建構項目的進度與變更提供一份完整的資訊，這些資訊包括建構項目的日期、關係人、狀態等，讓團隊成員對一個已經過時的設計文件進行程式實作等類似事件不會發生，可增團隊成員的溝通。

2.4.8 專案管理

專案管理必須負責規劃與排程，對所有工作進行監督，並控制所有的工作可以在標準內進行，使專案可以在制訂的成本與時間內開發出高品質的軟體，就是專案管理的主要目的。專案管理的活動主要有專案規劃、專案排程、專案成本估計、專案監控、團隊管理與風險管理等六項，如表 2.9 所示 [13][38][42]。

專案規劃是針對整個專案的進行提供一個資源、成本與時程的估計，並列出一系列可執行的方案。軟體專案規劃的基本特性如下：

- (1) 規劃工作必須由團隊成員共同參與；軟體專案由許多角色組成，每位角色對於本身的工作所花費的時間、成本等會較為清楚，共同規劃可以減少誤差的發生；由另一方面來看，參與規劃的人員最好具有實務工作經驗，否則規劃的結果可能會與實際產生較大的誤差。
- (2) 軟體規劃涉及軟體開發的技術、工具與設計方法的選擇。
- (3) 軟體規劃工作可能涉及系統整合的問題。

- (4) 軟體需求不易事先估計，當需求不確定時，規劃將會難以進行，所以需求的確認非常重要。

表 2.9 專案管理的活動

活動名稱	活動說明
專案規劃	針對整個專案的進行提供一個資源、成本與時程的估計，並列出一系列可執行的方案。
專案排程	對於專案的工作進行排程與分派。
專案成本估計	預估出完成一個活動需投入的工作量、時間以及總花費成本。
專案監控	對專案不斷監督，包括其進度、成本與品質等，都必須控制在規劃與估計的範圍之內。
團隊管理	團隊管理的議題包含團隊成員的溝通與協調、授權、工作分派等。
風險管理	風險管理的主要工作是預期可能影響專案的風險，並建立、規劃風險發生之後的因應對策，或是採取行動以降低、避免這些風險發生的機會。

專案排程在「派工管理」小節有加以說明，這裡再加以補充；對於軟體專案來說，因為需求演化本質的影響，專案排程變成一項更困難的工作，當需求變更時，可能會有新的工作加入，或是有些工作被取消，造成既定的專案排程必須加以修改；另一方面，如果專案延遲了，對專案增加人力以趕上進度，在一般情況下，對於軟體專案並不適用，這是因為增加人力必須對其進行訓練，而且會造成溝通與協調的增加，勢必會再拉低專案進行的速度。

專案成本估計必須預估出完成一個活動需投入的工作量、時間以及總花費成本；專案成本估計與專案規劃幾乎是同時進行，且是專案排程的基礎。在專案進行的過程中，如果實際花費的成本比估計的還多，專案經理就必須進行因應對策，以降低成本。

在專案進行的過程中，專案經理必須對專案不斷監督，包括其進度、成本與品質等，都必須控制在規劃與估計的範圍之內。專案經理應事先規劃一套機制，當專案有異常現象出現，專案經理即可參考這套機制立即採取行動。

軟體專案通過需要很多人合力完成，此時團隊管理就顯得重要許多，團隊管理的議題包含團隊成員的溝通與協調、授權與工作分派等；團隊管理的首要

步驟就是選擇或設計一個適合的組織結構，好的組織結構對於團隊的溝通、協調與合作都會有所幫助。

風險管理的主要工作是預期可能影響專案的風險，並建立、規劃風險發生之後的因應對策，或是採取行動以降低、避免這些風險發生的機會。風險管理的程序如下：

- (1) 風險識別 (Risk identification)：識別可能發生的風險。
- (2) 風險分析 (Risk analysis)：分析風險發生的可能性以及發生的後果。
- (3) 風險規劃 (Risk planning)：制訂能夠降低或避免風險對專案影響的計劃。
- (4) 風險監督 (Risk monitoring)：持續評估風險，如果風險發生則立刻採取相應的行動；如果評估後確定風險不會發生則停止監督此項風險。

2.5 Visual Studio Team System

本研究將以 Visual Studio 2005 Team System (VSTS) 展現 RALM 的案例實作，本節對 VSTS 的整體概念及主要功能做一個概括性的介紹[27][29][37][41]。

2.5.1 VSTS 概觀

在軟體生命週期中有許多不同的角色，他們使用不同的工具來完成各自的工作，也必須遵循一個流程使軟體開發更順利。微軟推出的 VSTS 融合了軟體生命週期中的角色、工具及流程，使每個角色都能在同一平台上工作，讓溝通更順暢、資料的整合更快速且順利；微軟對 VSTS 所設立的目標如下：

- (1) 提供軟體工程所需完整的管理系統與工具，使團隊中每個成員的價值最大化：VSTS 的導入，可在整個軟體生命週期中依照各成員所擔任的角色選擇適當的工具，並運用 Visual Studio Team Foundation Server (TFS) 進行團隊協同作業。
- (2) 減少開發過程的複雜度：讓專案的完成是可被預測、成果是可被累積且可被管理的。VSTS 透過 Microsoft Solutions Framework (MSF) 來簡化和引導開發工作流程，以達到提高軟體生命週期的能見性及可追蹤性，並使得軟體開發更趨向標準化；VSTS 更可以自訂流程以適合

各開發團隊。

- (3) 增進開發團隊的溝通與協同運作的效能：明確且不過度繁雜的文件是了解專案的要件，TFS 會自動蒐集相關軟體生命週期中的相關資訊並分析製成報表，達成簡化報表的功能。TFS 將蒐集到的資料存放在資料倉儲中，團隊中的相關成員都能由報表了解專案健康狀況。
- (4) 提供自訂和可擴充性：VSTS 可自訂的項目有：流程指引、工作項目類型、版本控制、報表與專案入口網站，如此一來 VSTS 將更靈活，不同的使用者皆可依照不同的需求自訂所需的環境，這樣的自訂功能可透過修改流程範本（process template）來達成[33][38]。VSTS 可自行擴充或使用其他合作廠商所開發的產品進行擴充，使 VSTS 更符合使用者的需求。

VSTS 可讓各角色個別安裝他們所需的工具，但如果要達到團隊開發的目標，就必須安裝 VSTS 的核心：Team Foundation Server（TFS），安裝 TFS 前有幾個軟體是必備的，這些軟體安裝後，TFS 才能安裝成功：

- (1) Microsoft SQL Server 2005：每個成員的產出物，包括所有的工作項目、原始程式碼和建置資料都儲存在 SQL Server 中。Analysis Services 和 Reporting Services 可支援專案入口網站和各種報表。
- (2) Microsoft SQL Server 2005 Hotfix：更新 SQL Server Analysis Services，更有效率地支援報表功能。
- (3) .NET Framework 2.0 Hotfix：更新 ASP.NET，使大型檔案的處理更有效率。
- (4) Windows SharePoint Portal Services：專案入口網站所使用的軟體。

此外，安裝 VSTS 的作業系統版本必須為 Windows Server 2003，並設定為應用程式伺服器（安裝 Windows 元件 - Internet Information Services（IIS）），並啟用 ASP.NET。

2.5.2 VSTS 的架構

VSTS 的是一個多層式的架構，它的邏輯架構由資料層、應用程式層和用戶層組成，其中的資料層與應用程式層就是組成 Team Foundation Server (TFS) 的主要元素。

資料層裝載在 SQL Server 2005 上，即可儲存工作項目、版本控制原始程式檔和測試結果等專案度量資訊，TFS 並使用了 SQL Server2005 的資料庫引擎和分析服務兩項功能，在安裝 TFS 時，系統會自動將下列資料庫安裝到資料層中：

- (1) TFSIntegration：TFS 核心服務。
- (2) TFSWarehouse：Team Foundation 資料倉儲。
- (3) TFSWorkItemTracking：TFS 工作項目追蹤資料庫。
- (4) TFSWorkItemTrackingAttachments：TFS 工作項目附件資料庫。
- (5) TFSVersionControl：TFS 版本控制資料庫。
- (6) TFSActivityLogging：TFS 版本控制事件紀錄檔。
- (7) TFSBuild：Team Build 資料。

將 TFS 裝載在 SQL Server 2005 上的優點有下列幾點：

- (1) TFS 成為維護和備份工作項目、報表資料、程式碼等專案資產。
- (2) 可利用 SQL Server 2005 的其他功能。
- (3) SQL Server 2005 內含 Reporting Services，是 TFS 內建報表功能的基礎，使用 SQL Server 2005 即可用來編輯、新增報表。

應用程式層由 SQL Server 2005 Reporting Services、Team Foundation 核心服務和 Windows SharePoint Services (WSS) 所呼叫的 ASP.NET Web 服務所組成。此外，應用程式層還包含 Team Foundation Build 和 Team Foundation Server Proxy，前者提供建置自動化服務，後者會快取之前存取過的原始檔控制檔案，以改善使用低頻寬連接時的原始檔控制作業效能。

用戶層由 Team 總管 (Team Explorer) 所組成，這是一個和 TFS 連結的簡

易使用介面，能夠被用來建立和管理專案等，Team 總管除了可獨立使用之外，還能和 Visual Studio 2005 整合使用（除了 Visual Studio 2005 Express Edition 以外）；用戶層還可經由 Visual Studio 2005 Tools for the Microsoft Office System 與 Microsoft Office Excel 及 Microsoft Office Project 整合。透過 Team 總管與 Visual Studio 2005 提供給各角色的工具之間的整合，使用 VSTS 的各角色可以快速取得與發佈專案的各項產出物與資訊。

2.5.3 VSTS 功能介紹

圖 2.6 為 VSTS 所提供的功能與工具之間的關係。由 VSTS 的架構面來看，VSTS 可分為 TFS（圖 2.6 下方）和用戶端應用程式工具（圖 2.6 上方）兩大部份，圖中的灰色部份即代表工具所擁有的功能，有些功能並非專屬某些工具，例如「Unit Testing」就是 Team Developer 和 Team Tester 兩個工具皆有的功能。

其中 TFS 可說是 VSTS 的核心，因為 TFS，整個團隊成員會有更具效率的合作和溝通；用戶端應用程式工具分別有 Visual Studio Team Architect、Visual Studio Team Developer 和 Visual Studio Team Tester 三個部份，這三個部份可依團隊中的角色需要，分別當成獨立的工具使用，TFS 透過 Team 總管和用戶端工具整合在一起，即可支援整個軟體開發團隊。

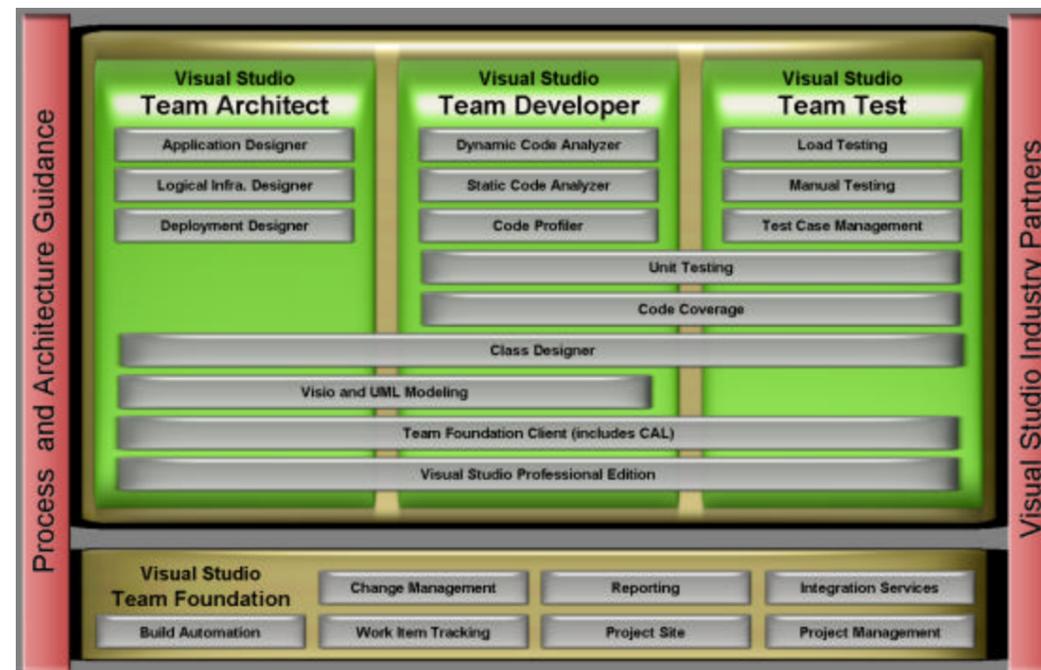


圖 2.6 Visual Studio Team System功能與工具關聯圖

(資料來源：Microsoft®)

2.5.3.1 Team Architect、Team Developer與 Team Tester

首先介紹圖 5.1 上方，也就是用戶端應用程式工具的部份。對架構人員來說，VSTS 提供一套直覺的設計工具給架構設計人員，稱為「分散式系統設計工具」，這個工具內含邏輯 DataCenter 設計工具、應用程式設計工具、系統設計工具及部署設計工具。這四個工具所建立出來的皆是圖表，透過這些圖表可以更清楚的表達架構設計的說明，增進團隊成員對架構的了解，進而提高專案的品質。

邏輯 DataCenter 設計工具用來建立部署環境的邏輯表示。它可以定義伺服器類型、主機間的通訊路徑（伺服器端點）、區域、區域端點、已啟用的服務類型、應用程式服務的組態及在邏輯伺服器和端點加入或移除設定資源。其中區域表示界限，例如網域，或是任何主機的邏輯集合。邏輯 DataCenter 設計工具也提供了自訂邏輯伺服器、伺服器群組、區域、區域群組的功能，並存放起來使用。

應用程式設計工具用來定義應用程式的架構。它提供在一或多個邏輯 DataCenter 中進行測試部署，確認架構可以順利部署，達到早期發現問題，早期解決，降低成本。應用程式設計工具另外提供了產生程式碼和部署報告的功能；也可以自訂應用程式、應用程式群組、端點及端點群組。

系統設計工具可以把應用程式圖表合成為一個「系統」。部署設計工具使用邏輯 DataCenter 圖表來測試應用程式圖表的部署。驗證成功會產生一份報告，驗證失敗會在錯誤清單視窗中顯示發生哪些錯誤。

接下來介紹開發人員的工具；開發人員主要的產出物是程式碼，對程式碼的開發品質和版本管理都是重要的一環，VSTS 在這方面提供了類別設計、單元測試、程式碼涵蓋範圍、靜態分析和程式碼剖析等。

類別設計工具提供以圖形建立程式碼的方式來寫程式。使用圖形方式有很

多的優點，例如可以快速了解類別之間的關聯性，以及類別的繼承關係。類別圖表還提供了「物件測試工作台」，它可以使用圖形方式建立類別的實體，然後再以圖形方式來決定要呼叫哪個方法。另外類別設計工具也可以在圖形介面中進行重整作業。

開發人員可以在程式碼中直接產生單元測試專案，再來只要自己加入測試程式碼即可，抑或是先寫好單元測試程式碼，再由其中產生程式碼也可。要確認是否已撰寫足夠的單元測試，就要使用程式碼涵蓋範圍工具，這個工具可以驗證單元測試對程式碼的涵蓋範圍有多少，涵蓋率高的單元測試是優良的。

靜態分析工具的作用是尋找程式的缺失和不良的程式設計作法。它提供自訂規則，靜態分析工具啟動之後會檢查程式碼是否符合規則，結果會顯示在「錯誤」視窗中，並儲存在 CodeAnalysisLog.xml 的檔案之中。

程式碼剖析工具讓開發人員了解程式碼的效能，並知道是哪一部份的程式碼最花費執行時間，以此為根據來修改程式碼，使其達到良好的效能。

測試人員有許多測試工具可以使用：單元測試、Web 測試、負載測試、手動測試、一般測試及已排序的測試，除此之外，VSTS 提供了一個「測試管理員」的工具介面，讓測試管理人員可以更有效率地管理測試。「測試管理員」提供建立新測試、執行測試的功能，還可將為測試依屬性分組及篩選測試，只顯示符合設定篩選屬性的測試。單元測試在開發人員部份已介紹過了，接下來介紹其他類型的測試。

Web 測試可對網頁進行測試，開始執行 Web 測試時，「Web 測試錄製器」就會被啟動，它會記錄傳送到 Web 伺服器的每個 HTTP 請求。測試完成之後檢視詳細的資料，包括驗證狀態、HTTP 狀態、回應時間和傳輸位元組。

負載測試的目的是為了找出應用程式的效能極限和瓶頸，所以負載測試工具提供模擬多位使用者同時使用一個應用程式的情境，包括瀏覽器種類、網路頻寬種類的做用百分比等設定；此外，負載測試工具還可以儲存測試結果，並且詳細分析。

無法使用自動測試的部份就必須使用手動測試。手動測試使用純文字檔或

Microsoft Office Word 來記錄手動測試的步驟和說明，測試結果可以記錄下來，VSTS 會將其包含在報告之中。

一般測試用來整合現有的自動化測試技術或是其他沒有整合到 VSTS 之中的技術。它可以叫用外部資源。

VSTS 的測試工具還提供一個容器，稱為「已排序的測試」，它可以保存其他測試，並讓這些測試依照順序進行，另外已排序測試中還可以包含其他的已排序測試。

除了 Team Architect、Team Developer 與 Team Tester 之外，微軟還為 Visual Studio 增加了一個資料人員的工具：Team Edition for Database Professionals，在這裡就不多加介紹了。

2.5.3.2 Team Foundation Server

如前所述，TFS 為 VSTS 的核心，TFS 最重要的功能即是將團隊中每個成員原本各自獨立的工作統合起來，使成員之間的關聯更密切，進而提高團隊的溝通與協同合作的品質。我們可以把 TFS 想像成中心機構，它做為每個團隊成員的橋梁，更進一步的，它還會把專案團隊的各種資料整理、分析並製成報表，讓相關成員更清楚專案的相關資訊。Team Foundation Server (TFS) 所提供的功能如下：

- (1) 建置伺服器：Team Foundation Build 提供了一個可擴充並建置自動化的工具，它可以透過介面精靈輕易設定並規劃建置流程。執行建置之後，Team Foundation Build 會產生一份報告，由這份報告可以了解建置的狀況。
- (2) 建構與變更管理：對專案中必須管理的項目提供版本控制。在版本控制方面，簽出程式碼的時候，可以選定鎖定類型：允許共同簽出、不允許其他使用者簽出與簽入或允許其他使用者簽出但不允許簽入三項選擇；簽入程式碼時，可以事先建立好版本控制簽入原則，這些原則是開發人員要簽入時必須要遵守的，在簽入的時候也可和工作項目產生關聯。如果程式碼還沒達到簽入原則的要求，但必須存放到版本控

制伺服器上，這時就可以使用擱置（Shelving）的功能，擱置可以將程式碼放在伺服器端而不必簽入。開發人員還可以對程式碼做分支和合併的動作，在將變更簽入伺服器端之前，對新建立的分支程式碼所做的任何變更都是在本機工作區。合併可以比較分支程式碼的所有變更，之後由使用者選擇要合併哪些變更，合併後的資訊會被記錄下來，日後可以追蹤。

- (3) 工作項目追蹤：建立和管理風險、Bug、工作、案例、服務品質等工作項目（工作項目可依各團隊的需求進行自訂）。專案管理或相關人員可建立工作項目並指派給其他團隊成員，工作項目是一個用來追蹤指派和狀態的「資料庫記錄」；VSTS 提供工作項目的定義，以 MSF for Agile Software Development 為例，它包含了五種工作項目：風險（risk）、Bug、工作（task）、案例（scenario）及服務需求品質（QoS）。VSTS 提供 Microsoft Office Excel 及 Microsoft Office Project 的整合，這 2 個工具可以和 TFS 連結，對工作項目做查詢和編輯的動作，讓慣用這兩個工具的專案管理人員使用上更得心應手。
- (4) 專案管理：建立和管理專案。建立新專案的時候，專案管理人員必須定義專案、選擇方法、版本控制的設定和專案入口網站的設定。這些步驟可由 VSTS 一步步的輕鬆設定完成。專案的流程可選擇使用 VSTS 內附的 MSF for Agile Development 和 MSF for CMMI Process Improvement 兩種流程方法，如果團隊中有自訂的流程方法，也可以從中選取。專案建立完成之後，專案管理人員必須設定安全性、建立區域和反覆項目、設定簽入原則等；這裡的安全性設定是指給定每個成員權限。在 VSTS 中，可以定義「伺服器」及「專案」兩個群組，「伺服器」是針對 TFS 整體的權限設定，而「專案」的設定則是限於目前所指定的專案。建立區域可邏輯地將角色或是工作項目分類，更進一步的，還可以設定各區域的安全性，除了區域之外，還可以建立反覆項目，各工作項目可和區域與反覆項目做關聯。
- (5) 專案網站：專案入口網站使用 Windows SharePoint Services（WSS）建

立，在這個入口網站中，只要團隊成員有足夠的權限，即可上傳、編輯和檢視在網站上的各式文件。

- (6) 整合服務：可和多個軟體工具整合，如 VSTS 的用戶端應用程式工具、Microsoft Office Excel、Microsoft Office Project 等。
- (7) 報表：這是 VSTS 不可或缺的報表功能。在整個專案的過程中，VSTS 會自動蒐集專案資訊至資料倉儲中，並產生報表，這些資訊包含了所有工作項目的資訊、建置工作的資訊、測試工作的資訊等，由這些資訊即可生成各種不同的報表；VSTS 內建了許多報表，還可以自訂與新增其他報表。

第三章 軟體生命週期管理快速導入方法

3.1 RALM

為減低軟體生命週期管理導入的困難，本研究提出一個名為 RALM 的軟體生命週期管理導入方法。本方法的整體導入流程如圖 3.1 所示，RALM 包括「軟體生命週期管理導入」與「軟體生命週期管理工具導入」兩大部份。

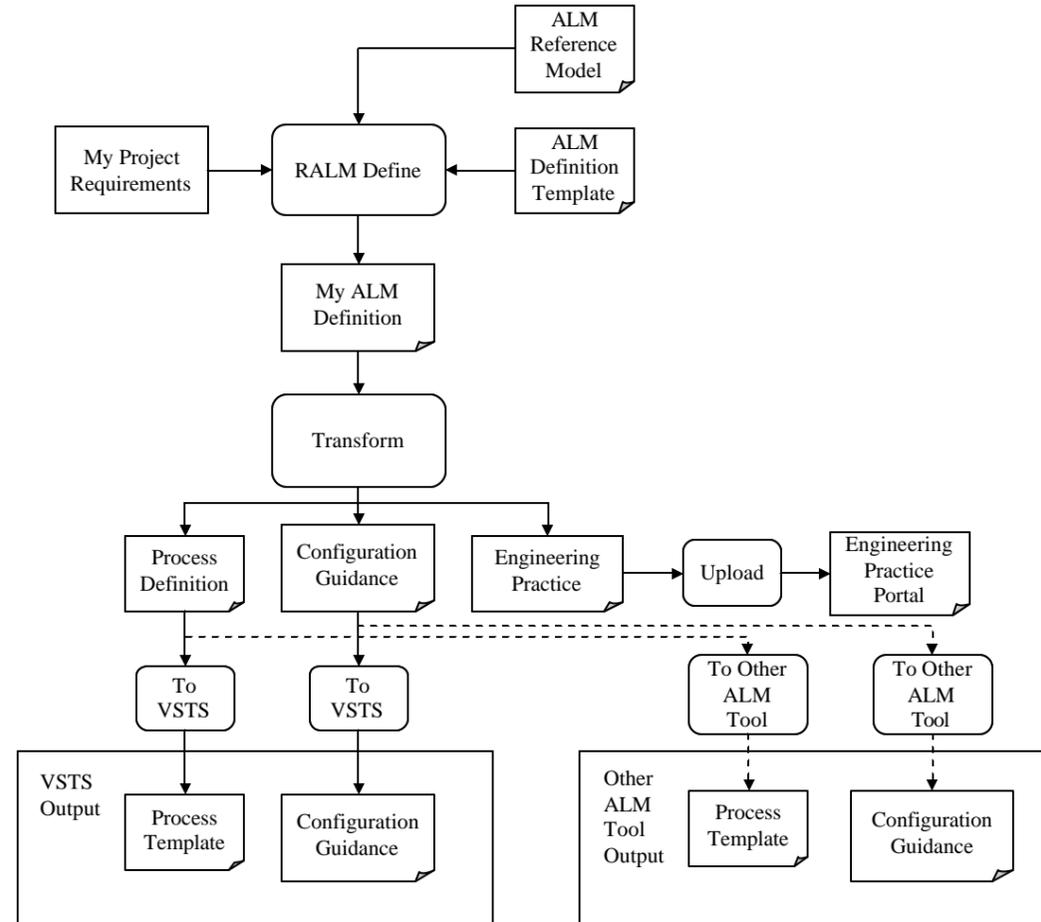


圖 3.1 RALM 流程

- (1) 軟體生命週期管理導入：根據軟體團隊的軟體專案需求（my project requirement），利用軟體生命週期管理參考模型（ALM reference model）及其相對應的軟體生命週期管理定義樣板（ALM Definition template），定義符合專案目標的軟體生命週期管理需求，產出軟體團隊所需的軟體生命週期管理定義（my ALM definition），其產出物為

ALM-XML。

- (2) 軟體生命週期管理系統導入：軟體生命週期管理定義經轉換，將會輸出工程指引（engineering practice）、流程定義（process definition）及軟體生命週期管理工具設定指引（configuration guidance）三個部份，這三個部份再經由對應各軟體生命週期管理工具的轉換工具，工程指引將被上載至工程指引網站（engineering practice portal），而流程定義與管理系統設定指引的內容則是對應至各種軟體生命週期管理工具的設定方式。本研究以 VSTS 做為軟體生命週期管理工具做為導入案例說明，透過 ALM2VSTS 工具，工程指引會被上載至 VSTS 的專案入口網站中、流程定義會被轉換為 VSTS 流程範本、軟體生命週期管理工具設定指引的內容則是針對 VSTS 的規劃與設定。

本章接下來的內容將對軟體生命週期管理定義樣板、ALM-XML 定義以及 ALM2VSTS 的實作等進行詳細的說明與介紹。

3.2 ALM-XML

ALM-XML 是使用軟體生命週期管理定義樣板記錄軟體生命週期管理需求所得到的產出，ALM-XML 使用 XML 文件格式為基礎，可以靈活地將軟體生命週期管理定義轉換為工程指引、流程定義與軟體生命週期管理工具設定指引三項產出，更進一步地，還可利用 XML 文件的彈性，將其對應至各種軟體生命週期管理工具的導入，關於 ALM-XML 的 XML Schema 定義請參考附件一。

每個軟體生命週期管理定義樣板的內容都是導入軟體生命週期管理必須具備的要素，也是軟體團隊的導入需求；本研究提出十個軟體生命週期管理定義樣板，各個軟體生命週期管理定義樣板定義以及與 ALM-XML 的元素對應說明如下：

- (1) 專案定義：記錄專案的基本資訊，如表 3.1；其中專案目標與專案描述提供軟體團隊一個可見的記錄，輔助它們朝著正確的目標發展軟體；專案準則與專案紀律則讓每一位團隊成員了解專案的核心原則，進而培養團隊紀律。與專案定義相對應的元素為<ProjectDef>其子元素有 <Goals> 、 <Descriptions> 、 <Principles> 與

<Disciplines>。 <Goals>定義專案目標，其子元素為<Goal>，每個<Goal>記錄一項專案目標； <Description>記錄專案描述； <Principles>元素定義專案準則，其子元素為<Principle>，每個<Principle>記錄一項專案準則； <Disciplines>元素定義專案紀律，其子元素為<Discipline>，每個<Discipline>記錄一項專案紀律。

表 3.1 軟體生命週期管理定義樣板之專案定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Name>	軟體專案的名稱
<Goals>	軟體專案的目標
<Description>	軟體專案的描述
<Principles>	團隊成員的工作準則
<Disciplines>	團隊成員須遵循的紀律

- (2) 產出物類型定義：記錄軟體開發過程所有產出物的資訊，如表 3.2。產出物類型的定義使軟體團隊了解軟體生命週期的進行中，預計會有哪此些產出，且這些產出必須使用何種格式完成。團隊有共同的產出格式，可以增進溝通與合作的效率。與產出物類型定義相對應的元素為 <ArtifactDef>，其子元素為 <ArtifactType>，一個 <ArtifactType>元素定義一種產出物類型，<ArtifactType>子元素有 <Name>、 <Description>、 <Styling>、 <Url> 與 <Comment>； <Name>記錄產出物類型名稱、 <Description>記錄產出物的說明、 <Styling>記錄所產出物所使用的格式、 <Url>記錄產出物的放置位置，以及 <Comment>是其他補充說明。

表 3.2 軟體生命週期管理定義樣板之產出物類型定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<ArtifactType>	軟體生命週期各產出物類型，其子元素<Name>定義產出物類型名稱
<Description>	說明產出物的內容與用途等
<Styling>	說明產出物遵照何種共同格式
<Url>	文件規範的放置位址
<Comment>	其他說明

- (3) 角色定義：記錄開發團隊各角色的資訊，如表 3.3 所示。角色定義最主要的部份是各角色所負責的工作類型定義，這些工作類型定義說明在工作被執行前有哪些前置條件必須完成、執行工作的步驟為何、工作必須達到哪些後置條件才算是將工作完成，以及工作完成會有什麼產出物。這裡描述了流程中各工作的細節，讓每個角色對於流程的工作執行有所參考。與角色定義相對應的元素為<RoleDef>，其子元素為<Role>，每個<Role>定義一個角色。<Role>的子元素有<Name>、<Description>與<Task>，<Name>記錄角色名稱、<Description>對角色加以說明，<Task>是角色所需執行的工作。在<Task>底下的子元素有<EntryCriterias>、<Steps>、<ExitCriterias>、<ArtifactType>與<Comment>；<EntryCriterias>定義工作的允入準則，其子元素為<EntryCriteria>，每個<EntryCriteria>記錄一項允入準則；<Steps>定義工作的執行步驟，其子元素為<Step>，每個<Step>記錄一個工作執行步驟；<ExitCriterias>定義工作的允出準則，其子元素為<ExitCriteria>，每個<ExitCriteria>記錄一項允出準則；<ArtifactType>記錄工作的產出物類型；<Comment>是其他補充說明。

表 3.3 軟體生命週期管理定義樣板之角色定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Role>	軟體生命週期的角色，其子元素<Name>定義角色名稱
<Description>	說明角色所擔任的職務為何
<Task>	角色在軟體生命週期中所負責的工作類型，其子元素<Name>定義工作類型名稱
<EntryCriterias>	各工作的允入準則
<Steps>	各工作的執行步驟
<ExitCriterias>	各工作的允出準則
<ArtifactType>	各工作的產出物類型
<Comment>	其他說明

- (4) 組織定義：記錄開發團隊的組織資訊，如表 3.4 所示。這裡記錄了軟體團隊的組織結構以及組織中每位成員的帳號、名稱、角色、技能、水平等資料，這些資料可提供專案規劃、派工、人員管理以及軟體生命週期管理系統的規劃與設定等參考。與組織定義相對應的元素為<OrgDef>，其子元素為<Location>，定義一個單位所在地，<Location>的子元素有<Name>與<Section>，<Name>記錄所在地的名稱，<Section>定義單位。<Section>的子元素有<Name>與<Group>，<Name>記錄單位的名稱，<Group>定義單位內的群組。<Group>的子元素有<Name>與<Account>，<Name>記錄群組的名稱，<Account>定義群組內的人員帳號。<Account>的子元素有<Name>、<Actor>、<Role>、<Skills>與<Comment>，<Name>記錄帳號的名稱、<Actor>記錄帳號所對應的人員名稱、<Role>記錄人員所擔任的角色，<Skills>定義此人員的技能。<Skills>的子元素有<Name>與<SkillLevel>，<Name>記錄技能的名稱，<SkillLevel>記錄技能的水平。<Comment>是其他補充說明。

表 3.4 軟體生命週期管理定義樣板之組織定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Location>	單位的所在地，其子元素<Name>定義所在地名稱
<Section>	參與軟體開發的單位，其子元素<Name>定義單位名稱
<Group>	各單位所包含的群組，其子元素<Name>定義群組名稱
<Account>	各群組所包含的人員帳號，其子元素<Name>定義帳號名稱
<Actor>	與人員帳號對應的人員名稱
<Role>	各人員所擔任的角色
<Skills>	各人員所具備的技能
<SkillLevel>	技能的熟練度
<Comment>	其他說明

- (5) 流程定義：記錄開發流程的相關資訊，如表 3.5 所示。相對於角色定義，這裡所表達的是工作與流程的整體性，說明了流程中的工作類型屬於哪些週期階段，提供各角色另一個整體性的流程說明。與流程定義相對應的元素為<ProcessDef>，子元素為<Phase>，定義一個週期階段。<Phase>的子元素有<Name>與<Task>，<Name>記錄週期名稱，<Task>定義此週期所包含的工作類型。<Task>的子元素有<Name>、<IterationPoint>與<Comment>，<Name>記錄工作類型名稱、<IterationPoint>定義工作的反覆點，<Comment>是其其他補充說明。

表 3.5 軟體生命週期管理定義樣板之流程定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Phase>	流程各階段的週期，其子元素<Name>定義週期名稱
<Task>	各週期所包含的工作類型，其子元素<Name>定義工作類型名稱
<IterationPoint>	工作的反覆點定義
<Comment>	其他說明

- (6) 配置定義：記錄原始檔控制的資料夾名稱及其目錄結構，如表 3.6 所

示。規劃原始檔控制資料夾的配置，有利於產出物的分類以及授權定義的進行，原始檔控制可避免產出物混亂的發生，並確保產出物的歷程資料隨時可被存取。與配置定義相對應的元素為<CMDef>，其子元素為<Folder>，用來定義資料夾。<Folder>的子元素有<Name>、<Comment>與<Children>，<Name>記錄資料夾名稱，<Comment>是其他補充說明，<Children>則是子資料夾的定義。<Children>的子元素再次參考<Folder>，以遞迴參考的方式完成目錄結構。

表 3.6 軟體生命週期管理定義樣板之配置定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Folder>	資料夾的配置，其子元素<Name>定義資料夾名稱
<Children>	子資料夾的配置
<Comment>	其他說明

- (7) 授權定義：記錄原始檔控制各資料夾的群組權限設定，如表 3.7 所示。此處以配置定義為基礎，再對配置定義所建的資料夾進行授權定義；授權定義管控各項產出物不會被任意讀取與簽出，讓團隊成員專注於他們所負責的部份。與授權定義相對應的元素為<AADef>，其子元素為<Folder>，定義要被設定權限的資料夾。<Folder>的子元素有<Name>與<Group>，<Name>記錄資料夾名稱，<Group>是被授權的群組定義。<Group>的子元素有<Name>、<Read>、<CheckIn>、<CheckOut>、<Lock>與<Comment>，<Name>記錄群組名稱，<Read>、<CheckIn>、<CheckOut>與<Lock>各自表示讀取、簽出、簽入與鎖定四個權限，皆以「Allow 屬性」定義是否被授與這些權限。<Comment>是其他補充說明。

表 3.7 軟體生命週期管理定義樣板之授權定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Folder>	設定授權的資料夾，其子元素<Name>定義資料夾名稱
<Group>	被設定此資料夾授權的群組，其子元素<Name>定義群組名稱
<Read>	資料夾內容的讀取權限
<CheckIn>	資料夾內容的簽出權限
<CheckOut>	資料夾內容的簽入權限
<Lock>	資料夾內容的鎖定權限
<Comment>	其他說明

- (8) 派工定義：記錄專案初始已規劃好的派工資訊，如表 3.8 所示，工作的技能與困難度資料可對應至組織定義中各成員的技能與水平，提供派工時一項重要的參考。與派工定義相對應的元素為<TaskDef>，其子元素為<Task>，定義一項工作。<Task>的子元素有<Name>、<Phase>、<Area>、<TaskType>、<Skill>、<Level>、<Actor>、<Ref>、<Artifact>、<CheckIn>、<StartTime>、<EndTime>、<Cost>與<Comment>。<Name>記錄工作名稱、<Phase>記錄工作所在的週期階段、<Area>記錄工作所在的區域、<TaskType>記錄工作的類型、<Skill>記錄工作所需的技能、<Level>記錄工作的困難度、<Actor>記錄負責工作的人員帳號、<Ref>記錄工作所需的參考附件、<CheckIn>記錄工作產出的放置位置、<StartTime>與<EndTime>記錄工作的開始與結束時間，<Cost>是完成工作的預估時間，<Comment>是其他補充說明。

表 3.8 軟體生命週期管理定義樣板之派工定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Task>	工作，其子元素<Name>定義工作名稱
<Phase>	工作所屬的週期名稱
<Area>	工作所屬的組織區域
<TaskType>	工作的類型
<Skill>	執行工作所需的技能
<Level>	工作的困難度
<Actor>	負責此工作的人員帳號
<Ref>	執行工作所需的參考附件名稱及所在資料夾名稱
<Artifact>	執行工作的產出物名稱
<CheckIn>	執行工作的產出物放置資料夾名稱
<StartTime>	工作開始執行的排程時間
<EndTime>	工作完成執行的排程時間
<Cost>	預估完成工作所花的時數
<Comment>	其他說明

- (9) 管理指標定義：記錄專案管理所需的重點指標資訊，如表 3.9 所示。對於軟體專案健康的監控是完成軟體開發的一項重要課題，了解必須由什麼角度來觀察專案健康才能切到重點，是管理指標定義最主要的目的。與管理指標定義相對應的元素為 <PMDef>，其子元素為 <Indicator>，定前管理指標。<Indicator> 的子元素為 <Name>、<Goal>、<Description>與<Commnet>，<Name>記錄管理指標的名稱、<Goal>記錄管理指標的管理目的為何、<Description>記錄管理指標的各項說明，如指標的意義，<Comment>是其他補充說明。

表 3.9 軟體生命週期管理定義樣板之管理指標定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<Indicator>	管理指標，其子元素<Name> 定義管理指標名稱
<Goal>	管理指標對於軟體專案管理可以做到的管理目的
<Description>	說明此指標所代表的軟體專案情況與狀態
<Commnet>	其他說明

- (10) 議題定義：記錄有哪些議題類型定義，如表 3.10 所示。定義議題類型，可讓團隊成員了解有哪些議題類型，以及這些議題的處理步驟、關係人。與議題定義相對應的元素為<IssueTrackingDef>，其子元素為<IssueTracking>，定義議題類型。<IssueTracking>的子元素有<Name>、<Steps>、<Field>與<Comment>，<Name>記錄議題類型名稱、<Steps>定義議題處理步驟，<Field>定義議題的欄位，<Comment>是其他補充說明。<Steps>的子元素為<Step>，記錄議題處理的一個步驟。<Field>的子元素有<Name>與<Description>，<Name>記錄議題欄位名稱，<Description>記錄議題欄位的說明。

表 3.10 軟體生命週期管理定義樣板之議題定義說明

元素名稱	與軟體生命週期管理定義樣板的對應
<IssueTracking>	議題類型，其子元素<Name> 定義議題名稱
<Steps>	議題的處理步驟
<Field>	議題包含哪些欄位，其子元素<Name> 定義議題欄位名稱
<Description>	說明議題欄位所代表的意義
<Comment>	其他說明

3.3 ALM2VSTS 工具實作

在實作部份，本研究所採用的應用程式、語言和 API 為：

- Microsoft Office 2003 Excel

- Microsoft Office 2003 XML Reference Schemas[36]。
- Java SE 6.0。
- Java Excel API 2.6.6[35]。
- Xalan-Java 2.7.0[39]。

本研究在軟體生命週期管理定義樣板實作部份使用 Microsoft® Office 2003 Excel 文件做為記錄 RALM 定義的工具，Microsoft® Office 2003 Excel 的優點為簡單、易用；我們將以一個 Excel 文件做為軟體生命週期管理參考模型，並在此 Excel 文件創建多個表單做為相對應的軟體生命週期管理定義樣板，再利用 Java Excel API 撰寫程式將 Excel 文件轉換為 ALM-XML；Excel 轉換為 ALM-XML 的部份我們將它納入 ALM2VSTS 工具中。ALM-XML 的轉換使用 Java 語言與 XSLT 技術實作，其中 XSLT 技術使用 Xalan-Java 2.7.0 API 實作 XSL 轉換。

圖 3.2 是將 ALM2VSTS 工具以活動圖 (activity diagram) 展現的整體執行流程。各流程步驟說明如下：

- (1) 啟動 ALM2VSTS 工具，ALM2VSTS 工具即會要求選擇記錄軟體生命週期管理定義的 Excel 文件，以及輸出所要放置的目錄，選擇完成按下「Start」即可開始進行轉換步驟；如按下「Exit」即關閉 ALM2VSTS 工具。
- (2) 透過 Java Excel API 實作的程式碼將軟體生命週期管理定義的 Excel 文件轉換為 ALM-XML。
- (3) 使用 XML Schema 對 ALM-XML 進行驗證，如驗證成功則進行下一步驟；驗證失敗則回到步驟 (1)。
- (4) 複製 VSTS 流程範本、工程指引網站及 VSTS 系統設定指引所需的各種檔案至輸出資料夾；複製的檔案內容將在 3.3.1 至 3.3.3 說明。
- (5) 將 ALM-XML 轉換輸出 VSTS 流程範本、工程指引網站以及 VSTS 系統設定指引，至此整個轉換步驟完成，回到步驟 (1)。

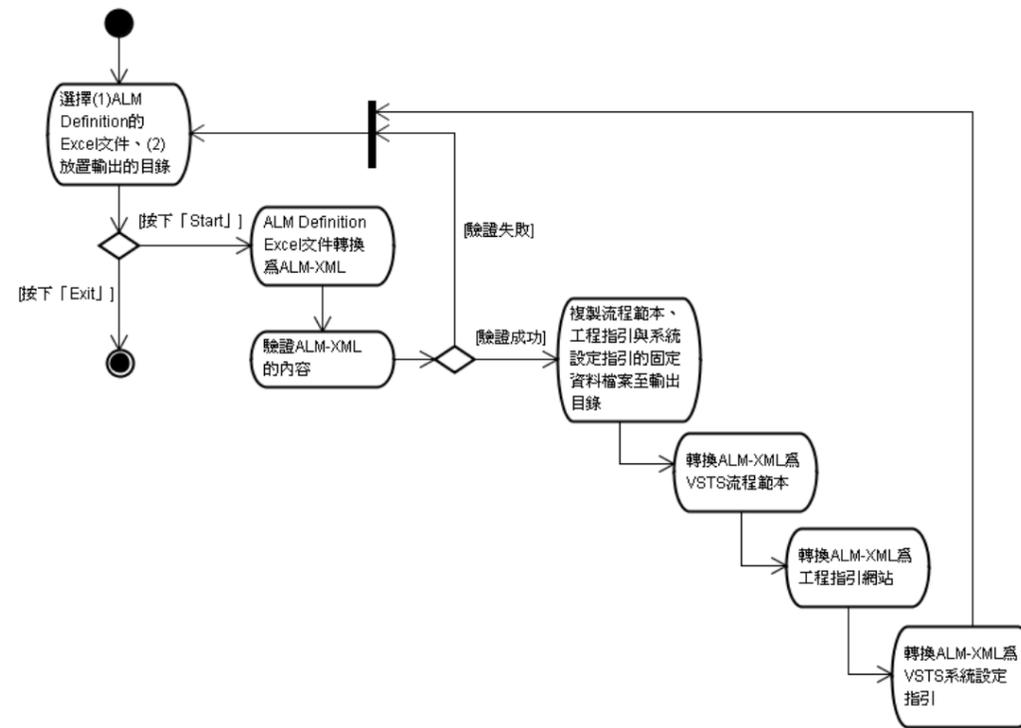


圖 3.2 ALM2VSTS 工具的執行步驟活動圖

接下來說明 ALM2VSTS 工具的設計結構。圖 3.3 是 ALM2VSTS 工具的類別關聯圖，下面將以圖 3.3 為基礎，再搭配各類別的類別圖，說明 ALM2VSTS 工具的設計。

圖 3.4 是 ALM2VSTSGUI 類別的類別圖，這個類別包含圖形使用界面與主程式，它含有四個內部類別（Inner Class），主要做為各按鈕的事件處理，ALM2VSTSGUI.SrcButtonHandler 是選擇軟體生命週管理定義的按鈕事件處理；ALM2VSTSGUI.DestButtonHandler 是選擇放置輸出目錄的按鈕事件處理；ALM2VSTSGUI.StartButtonHandler 是開始進行轉換的按鈕事件處理，在此會建立一個 ALM2VSTS 類別的物件，並呼叫它處理軟體生命週管理定義轉換程序的方法；ALM2VSTSGUI.CancelButtonHandler 是關閉 ALM2VSTS 工具的按鈕事件處理。

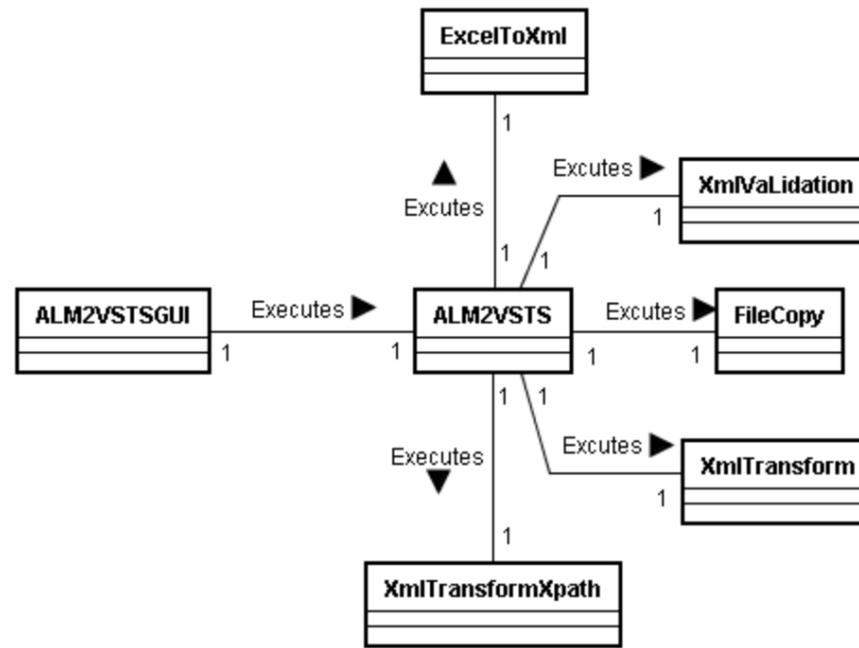


圖 3.3 ALM2VSTS 工具的類別關聯圖

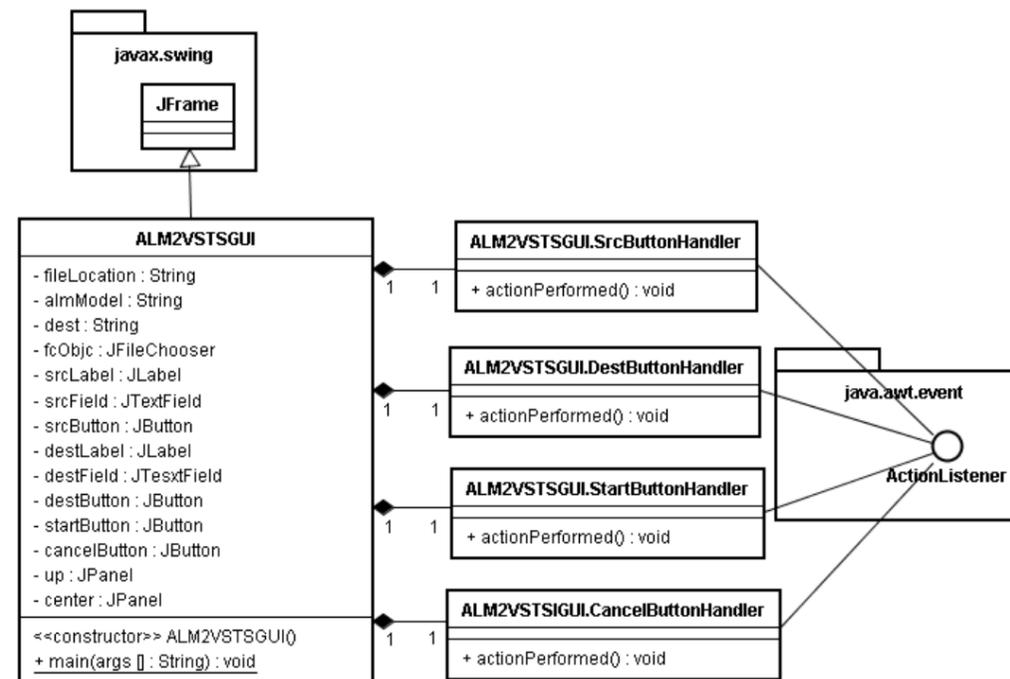


圖 3.4 ALM2VSTSGUI 類別

圖 3.5 是 ALM2VSTS 類別的類別圖。flag 屬性用來記錄 ALM-XML 是否通過驗證。此類別的 execute()方法含有兩個參數，almModel 參數必須傳入軟體生

命週期管理定義 Excel 文件的檔案名稱；dest 參數必須傳入欲放置輸出的目錄路徑；execute() 將會建立 ExcelToXml、XmlValidation、FileCopy、XmlTransform 與 XmlTransformXPath 的物件，並透過這些物件呼叫轉換流程所需的方法。

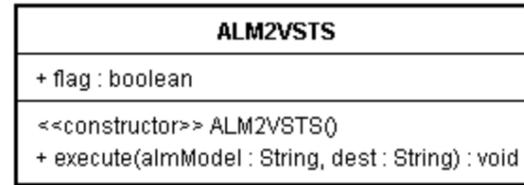


圖 3.5 ALM2VSTS 類別

圖 3.6 是 ExcelToXml 類別的類別圖，ExcelToXml 主要是做軟體生命週期管理定義的 Excel 文件轉換為 ALM-XML 的處理。建立 ExcelToXml 類別的物件需傳入兩個參數，excelFile 是欲轉換的 Excel 文件檔案名稱，而 xmlFile 則是轉換完成的 ALM-XML 檔案名稱；ProjectDef()、ArtifactDef()、RoleDef()、OrgDef()、ProcessDef()、CMDef()、AADef()、TaskDef()、PMDef() 及 IssueTrackingDef() 等十個方法，是對應至各軟體生命週期管理定義樣板的轉換方法。

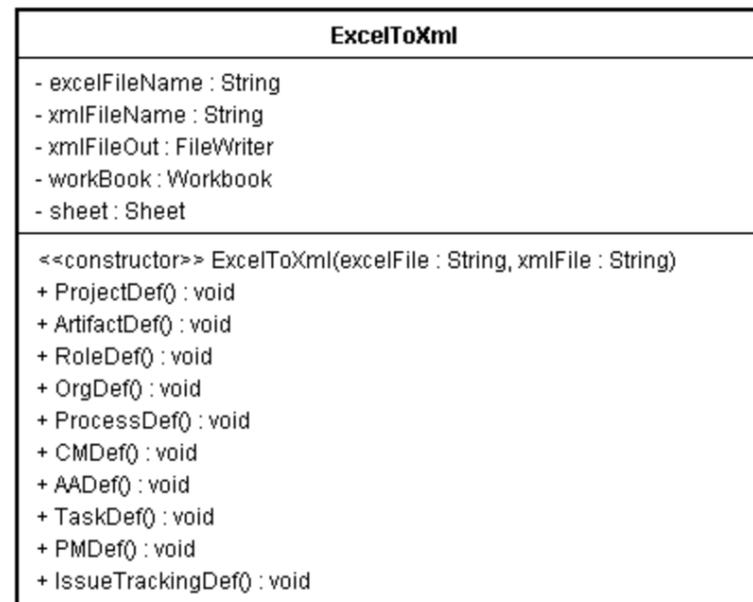


圖 3.6 ExcelToXml 類別

圖 3.7 是 XmlValidation 類別的類別圖，xmlValidator()方法執行 ALM-XML 的驗證，xmlSchema 參數必須傳入驗證的 XML Schema 定義；xmlDoc 是要被驗證的 ALM-XML 檔案。

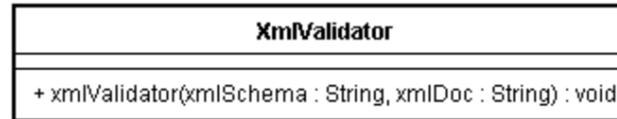


圖 3.7 XmlValidator 類別

圖 3.8 是 FileCopy 類別的類別圖，fileCopy()方法可把來源檔案複製至目的檔案。src 參數是來源檔案，dest 參數則是目的檔案。

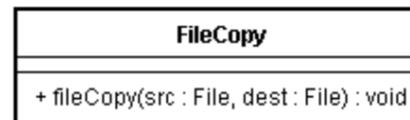


圖 3.8 FileCopy 類別

圖 3.9 與圖 3.10 是 XmlTransform 類別與 XmlTransformXpath 類別的類別圖。這兩個類別是做 ALM-XML 轉換的處理，兩者的屬性與方法相似，不同的地方主要在於 XmlTransformXpath 類別是用來轉換角色定義、產出物類型定義、議題定義這類內容的數量不定，且必須被展示在工程指引裡的軟體生命週期管理定義樣板；原因在於工程指引會使用 HTML 文件展示，而這些定義必須可被連結，但其定義內容的數量不定，導致網頁名稱無法固定，所以就必須在其命名中加上編號來加以區別，例如各角色說明網頁的命名方式為 Role1.html、Role2.html Rolen.html。這兩個類別皆有 xslProcess 與 cmd 兩個屬性，xslProcess 是一個 Xalan-Java 所提供的 XSLT 處理物件，只要將 XSLT 處理器所需的參數存入 cmd 字串陣列，再傳給 xslProcess 即可進行軟體生命週期管理定義 XML 文件的各項轉換。transform()方法就是執行轉換的方法，xmlSrc 參數為軟體生命週期管理定義 XML 文件；xslSrc 參數為轉換所需的 XSL 文件；outDest 參數則是輸出要放置的目錄；XmlTransformXpath 類別的 transform()方法多加一個 xmlPath 參數做為命名編號使用。

XmlTransform
- cmd : String[] - xslProcess : org.apache.xalan.xslt.Process
<<constructor>> XmlTransform() + transform(xmlSrc : String, xslSrc : String, outDest : String) : void

圖 3.9 XmlTransform 類別

XmlTransformXpath
- cmd : String[] - xslProcess : org.apache.xalan.xslt.Process
<<constructor>> XmlTransformXpath() + transform(xmlSrc : String, xslSrc : String, outDest : String, xmlPath : String) : void

圖 3.10 XmlTransformXpath 類別

接著對 ALM2VSTS 的輸入以及輸出做一個詳細的說明。ALM2VSTS 的輸出有 VSTS 流程範本、工程指引網站與 VSTS 系統設定指引，我們將以這三個輸出做為分類的方式說明 ALM2VSTS 的除了 ALM-XML 之外還需要哪些輸入，並對各輸出做詳細的說明。

3.3.1 VSTS 流程範本

本論文的 VSTS 流程範本內容是以 MSF for Agile Software Development v4.0 為基礎修改而來的，MSF for Agile Software Development v4.0 是 VSTS 預設的流程範本之一，其結構如圖 3.11 所示。

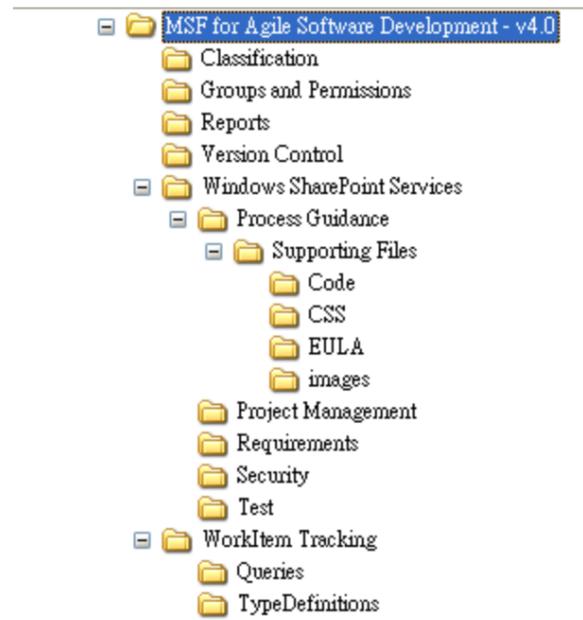


圖 3.11 MSF for Agile Software Development v4.0 目錄結構

MSF for Agile Software Development v4.0 流程範本各資料夾檔案如表 3.11 所示，各資料檔的內容說明可參考 MSDN2 Library 的線上說明文件[33][38]。ALM2VSTS 對於 VSTS 流程範本轉換所需的輸入幾乎和圖 3.12 所展現的相同，不同的地方在於：

- (1) 在表 3.11 的資料夾內容中，檔案名稱後面有 * 符號的檔案不需要包含在 ALM2VSTS 的輸入，因為這些檔案會透過 ALM2VSTS 轉換輸出。
- (2) 除了 FileMapping.xml、ReportsTask.xml、VersionControl.xml 與工作項目定義檔之外，其他的 XML 文件內容都把與 MSF for Agile Software Development v4.0 相關的內容清除。
- (3) 移除 MSF for Agile Software Development v4.0 的流程指引，但放入工程指引網站基本架構的 HTML 文件。預設的文件庫全部移除，只留下 Project Management 的專案計畫文件。

表 3.11 MSF for Agile Software Development v4.0 的資料夾內容

資料夾名稱	資料夾內容
Classification	(1) Classification.xml* (2) FileMapping.xml
Groups and Permissions	(1) GroupsandPermissions.xml*
Reports	(1) ReportsTasks.xml (2) 流程範本預設的報表定義檔 (* .rdl)
Version Control	(1) VersionControl.xml
Windows SharePoint Services	(1) WssTasks.xml (2) 流程指引網頁及其相關檔案 (3) 流程範本預設的文件庫
WorkItem Tracking	(1) workitems.xml* (2) Queries 資料夾下放置流程範本預設的工作項目查詢檔案 (* .wiq) (3) TypeDefinitions 資料夾下放置流程範本預設的工作項目定義檔 (* .xml)

而 ALM2VSTS 對於 VSTS 流程範本的輸出基本上也和圖 3.11 所展現的結構相同，但沒有包含其流程指引和文件庫的部份。在 VSTS 流程範本中，ALM2VSTS 主要的轉換輸出為：

- (1) Classification.xml：ALM-XML 的組織定義與流程定義內容會被轉換為這個檔案的「區域結構」與「反覆項目」兩項設定。
- (2) GroupsandPermissions.xml：ALM-XML 的角色定義內容的群組資訊會被轉換為這個檔案的「群組設定」，將群組的安全性會被賦予預設值。
- (3) workitems.xml：ALM-XML 的派工定義的部份內容會被轉換為這個檔案的「預設工作」。

3.3.2 工程指引網站

工程指引網站使用 HTML 文件完成，它需要的輸入是網站架構的 HTML 文件，這些 HTML 文件就等同於工程指引網站的骨架；檔案內容請參考附錄二。

當工程指引網站被 ALM2VSTS 轉換輸出之後，會直接將其放在 VSTS 流程範本的 Windows SharePoint Services\Supporting Files 資料夾中，上載 VSTS 流程範本時，即可同時上載至 VSTS 的專案入口網站中。工程指引網站的主要內容如下：

- (1) 專案說明：ALM-XML 的專案定義內容將被轉換到這裡。
- (2) 開發流程說明：ALM-XML 的流程定義內容將被轉換到這裡。展開各工作的連結，可以連結至與工作相關的角色與產出說明。
- (3) 角色說明：ALM-XML 的角色定義內容將被轉換到這裡。展開各角色的連結，可以連結至與角色相關的工作與產出說明。
- (4) 產出說明：ALM-XML 的產出類型定義內容將被轉換到這裡。展開各產出物的連結，可以連結並下載產出的範本。
- (5) 議題管理說明：ALM-XML 的議題定義內容將被轉換到這裡。展開各議題的連結，可看到議題的處理步驟及欄位說明。
- (6) 專案入口網站：工程指引網站上載至 VSTS 之後，使用 VSTS 開啟工程指引文件，即可透過這裡連結至 VSTS 所提供的專案入口網站。

3.3.3 VSTS 系統設定指引

此部份包含「VSTS 系統設定步驟說明」與「VSTS 系統設定內容」兩個文件。VSTS 系統設定步驟說明文件是 Microsoft® Office Word 的文件格式，這個文件內容會預先編輯完成，因為系統設定的步驟是固定的，對於 ALM2VSTS 來說，VSTS 系統設定步驟說明文件是一項輸入。

VSTS 系統設定內容文件是 ALM2VSTS 利用 ALM-XML 所產生的輸出，它使用 Microsoft® Office 2003 XML Reference Schemas 中的 WordprocessingML Schemas 編輯而成，是一個 XML 文件，但可使用 Microsoft® Office Word 開啟與編輯，使用上與一般的 Word 文件相同。

第四章 案例實作

4.1 案例說明

本案例是一家軟體公司，預計開發一個軟體產品，專案名稱為 **MobileProject**，其劇情說明如下。

此案例的「品保部」**Q** 與「開發部」**T**。欲開發的軟體有兩個模組，分別是 **N1** 與 **N2** 所構成，其中 **N1** 有三支程式 **n11**、**n12**、**n13**，**N2** 有四支程式 **n21**、**n22**、**n23**、**n24**，由 **T** 負責分析、設計與程式撰寫；**Q** 負責此產品的品質保證工作。

此案例中的角色與人員，分別說明如下。**T** 有 3 位人員，**a1** 擔任 SA (系統分析) 的角色，負責需求定義、系統分析工作；**d1** 擔任 SD (系統設計) 角色，負責系統設計；**p1** 與 **p2** 為 PR (程式設計) 角色，負責程式撰寫與單元測試兩個工作。**Q** 有 1 位 IR (整合與測試) 角色人員 **q1**，負責整合測試、系統測試、壓力測試、負載測試、安全性測試、回歸測試與 bug 提報等工作。至於本案例中的產出主要包括：需求規格表、系統分析書、程式規格書、原始程式碼、編譯程式碼、單元測試程式碼與測試報告書。本案例開發期間，需要一個專案入口網站。此專案入口網站，除了做為專案管理外，同時需要提供專案執行的準則說明。

本案例的開發流程遵循 waterfall 模型，進行方式說明如下：首先，**a1** 必須先行提交需求規格表；再來，由 **a1** 進行分析，產出 **N1**、與 **N2** 的分析書；**d1** 發展 **N1** 與 **N2** 的設計書；再來，由 **p1** 與 **p2** 分別發展 **N1** 與 **N2** 的程式撰寫。當程式完成後，由 **Q** 的 **q1** 進行相關測試並產出測試報告書。任何錯誤或修改，必須退回對應人員，直至專案完成。

在此專案中，為了管理相關產出文檔的存取授權以及版本變動控管，所有人員，只能存取他們所負責的程式、相關文件與共用元件。其配置管理規劃如下：

1. 目錄結構：

如圖 4.1，資料夾 N1 與 N2，分別放置與共用模組 N1 與 N2 的相關產出；資料夾 Test，放置開發完成與測試相關的產出；資料夾 Release，放置需求規格表以及已通過測試的軟體產品。

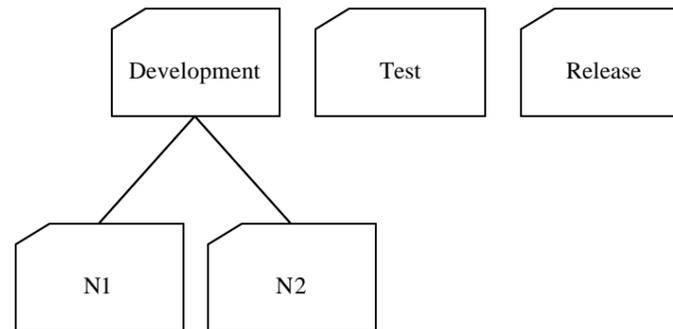


圖 4.1 MobileProject 專案的配置管理目錄結構

2. 認證與存取授權 (Authentication and Authorization) 管理：

1. 資料夾 N1：a1、d1、p1 具有簽入、簽出、鎖定、讀取的權限；q1 具有簽出、讀取的權限；p2 在此資料夾沒有任何權限。
2. 資料夾 N2：a1、d1、p2 具有簽入、簽出、鎖定、讀取的權限；q1 具有簽出、讀取的權限；p1 在此資料夾沒有任何權限。
3. 資料夾 Test：a1 與 q1 具有簽入、簽出、鎖定、讀取的權限；d1、p1 與 p2 在此資料夾只有「限制下的簽入」權限。
4. 資料夾 Release：a1 具有簽入、簽出、鎖定、讀取的權限；d1、p1、p2 與 q1 在此資料夾沒有任何權限。

另外，此案例開發過程中的產出物必須版本管理，讓產出物保有完整性與一致性，其需求如下：

1. 以前面的目錄結構為基礎，在每次簽入檔案時，新的版本皆被正確建立；同時，檔案的舊版本也必須可以順利取得。
2. 每個檔案版本與工作、產出或 bug 的關聯可被正確建立，使其可以雙向追蹤。

另外，在此案例中，為確保需求變更、bug 提報後的確實處理，定義議題

追蹤管理如下：

1. 需求議題：

- i. SA 與 SD 可提出需求變更。
- ii. SA 審核此變更並決定是否接受。若接受此需求變更，則定義完成期限並提報派工議題；若不接受，SA 擱置或否決此變更議題。
- iii. SA 根據變更需求，決定是否變更配置管理。當配置管理變更後，PM 通知所有相關人員。
- iv. SA 關閉此議題。

2. Bug 議題：

- i. SA、SD 與 IR 可提報 Bug 議題。
- ii. SA 審核此 Bug 並決定是否接受。若確認為 Bug，則在專案管理上，新增工作並派工給原開發人員，同時將此議題轉給該人員；若不接受為 Bug，SA 回覆理由給原提辦人員。
- iii. 原開發人員收到 SA 轉送之 Bug 議題，確認自己職責。如並非本人造成，退回 Bug 議題給 SA，並說明原因與建議作法；如是本人造成，則修復 Bug 後，回覆該 Bug 議題給 SA。
- iv. SA 回覆 Bug 議題給原提報人員，要求重新進行測試。
- v. 如果原提報人員確認修正完成，則關閉此議題；如仍無法同意，原提報人員回報 SA，SA 再重複上述步驟，直到能夠順利關閉此議題為止。

4.2 案例實作結果

本節針對上一節所說明的案例，使用本研究所提出的軟體生命週期管理的快速導入方法 - RALM，規劃各個軟體生命週期管理樣版內容；案例的 ALM-XML 定義的輸出請參考附錄三。

VSTS 流程範本的主要產出請參考附錄四。當 ALM2VSTS 將 VSTS 流程範本轉換出來，並且將流程所需的工作項目、文件以及報告都加入流程範本之後即可上載流程，如圖 4.2；流程上載完畢即可開始新增專案。圖 4.3 為上載至 VSTS 的工程指引網站，由此圖也可看出 MobileProject 專案建立成功。

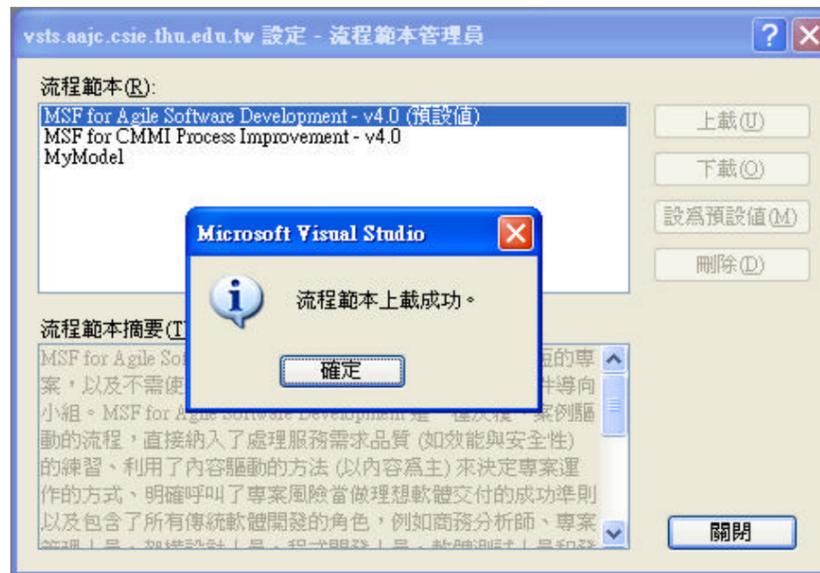


圖 4.2 上載流程範本

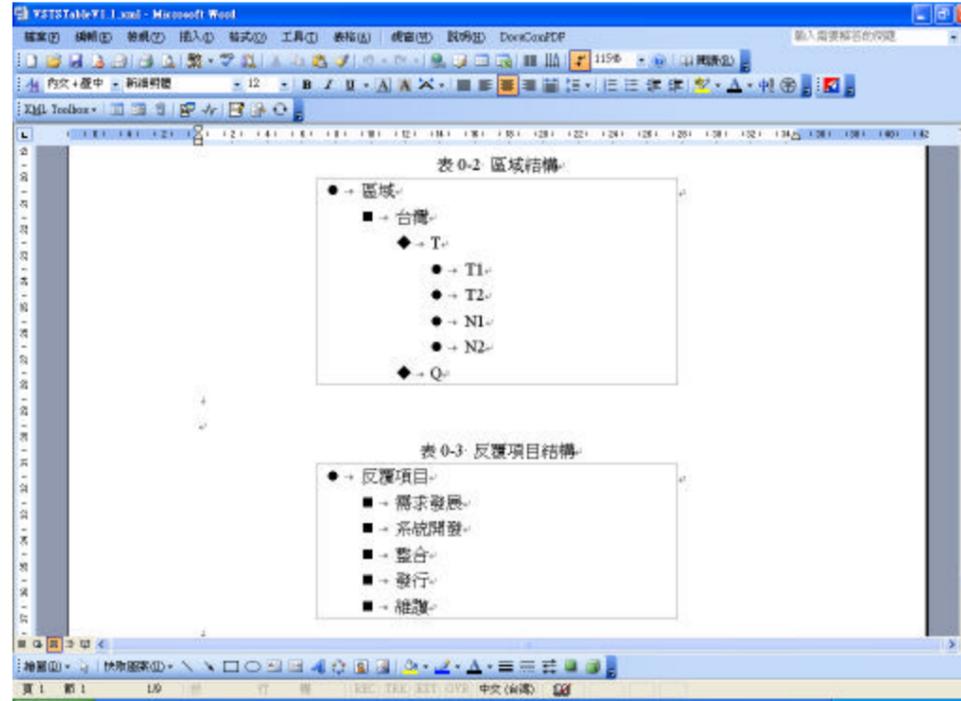


圖 4.5 VSTS 系統設定指引的設定內容

第五章 論文結論與未來研究方向

5.1 結論

在本論文中，我們首先探討軟體生命週期管理導入的重要性，也探討了軟體生命週期管理與相關工具導入的難度。本論文所提的軟體生命週期管理快速導入方法 – RALM，即是以簡易、快速且重用性高的特性，幫助降低軟體生命週期管理導入的難度，並快速定義所需之軟體生命週期管理需求。在案例實作中，本論文以 VSTS 為例，實際發展出 ALM2VSTS 工具，將 ALM-XML 自動轉換成 VSTS 流程範本、VSTS 系統設定指引以及工程指引網站。除了提供軟體生命週期管理工具導入初期的設定說明外，更提供軟體團隊在軟體開發的過程中，有一個可隨時查閱的共同工程準則，輔助軟體團隊紀律的增進。透過 RALM 簡易、快速的特性，以及 ALM2VSTS 的各項產出，軟體團隊將大大降低軟體生命週期管理導入的難度。

5.2 未來研究方向

本論文未來研究方向有以下三項：

- (1) 經由 RALM 的實際應用，更加精鍊其軟體生命週期管理定義樣板與 ALM-XML 的內容，使其更適合各式軟體團隊使用。
- (2) 更深入探討 VSTS 的其他軟體生命週期管理功能，以及自訂與擴充的功能，使 ALM2VSTS 工具的輸出更豐富。
- (3) 探討並比較各種軟體生命週期管理工具，發展更一般化、可對應至多種軟體生命週期管理工具的自動轉換工具。

參考文獻

- [1] Aaron G., Cass, Barbara Staudt Lerner, Stanley M. Sutton, Eric K. McCall, Alexander Wise, Leon J. Osterweil, “Little-JIL/Juliette: a process definition language and interpreter”, *Proceedings of the 22nd International Conference on Software Engineering ICSE '00*, June, 2000, pp:754 – 757.
- [2] Acuna S.T., Giandini R., Lasserre C.M., Quincoces V., “Capacities-centered integral software process formalization”, *Proceedings of the XX International Conference of the Chilean Computer Science Society, SCCC '00*, November, 2000, pp:152 – 161.
- [3] Alfonso Fuggetta, “Software process: a roadmap”, *Proceedings of the Conference on The Future of Software Engineering ICSE '00*, May, 2000, pp:25 – 34.
- [4] Alvin W. Yeo, “Global-software development lifecycle: an exploratory study”, *Proceedings of the SIGCHI conference on Human factors in computing systems*, March, 2001.
- [5] Armitage J.W., Kellner M.I., “A conceptual schema for process definitions and models”, *Proceedings of the third International Conference on the Software Process and Applying the Software Process*, October, 1994, pp:153 – 165.
- [6] Bill Curtis, Marc I. Kellner, Jim Over, “Process modeling”, *Communications of the ACM, Volume 35, Issue 9*, September, 1992, pp:75 – 90.
- [7] Carlshamre, P., Regnell, B, “Requirements lifecycle management and release planning in market-driven requirements engineering processes”, *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, September, 2000, pp: 961 – 965.
- [8] Carol Diane Klingler, “A case study in process definition”, *Proceedings of the Conference on TRI-Ada '93*, October, 1993, pp:65 – 79.
- [9] Dirk Jäger, Ansgar Schleicher, Bernhard Westfechtel, “Using UML for software process modeling”, *Proceedings of the 7th European Software Engineering Conference held jointly with the 7th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Volume 24, Issue 6*, October 1999, pp: 91 –

108.

- [10] Dowson M., Nejme B., Riddle W., “Concepts For Process Definition And Support”, *Proceedings of the 6th International on Software Process Workshop and support for the Software Process*, October, 1990, pp:87 – 90.
- [11] Harold Ossher, William Harrison, Peri Tarr, “Software engineering tools and environments: a roadmap”, *Proceedings of the Conference on the Future of Software Engineering*, May, 2000.
- [12] Henry, J.; Blasewitz, B, “Process definition: theory and reality”, *IEEE Software*, Volume 9, Issue 6, November, 1992, pp:103 – 105.
- [13] Ian Sommerville, “Software Engineering, 8/e”, Addison Wesley.
- [14] IEEE, “IEEE standard for developing software life cycle processes”, *IEEE Std 1074 – 1995* , April, 1996.
- [15] Jacobs D., Marlin C., “Software process representation to support multiple views”, *Proceedings of the First Asia-Pacific Software Engineering Conference*, December, 1994, pp:197 – 205.
- [16] John D. Riley, “An object-oriented approach to software process modeling and definition”, *Proceedings of the conference on TRI-Ada '94*, November 1994, pp: 16 – 22.
- [17] J. Mogilensky, D. Stipe , “Applying reusability to software process definition”, *Proceedings of the Conference on Tri-Ada '89: Ada technology in context: application, development, and deployment TRI-Ada '89*, January 1989, pp: 355 – 366.
- [18] Kaiser G.E., Popovich S. S., Ben-Shaul I.Z., “A bi-level language for software process modeling”, *Proceedings of the 15th International Conference on Software Engineering*, May, 1993, pp:132 – 143.
- [19] Lattanze, A.J., Rosso-Llopart, M., “Managing cyclical software development”, *Proceedings of the International Conference on Engineering and Technology Management*, October, 1998, pp: 62 – 70.
- [20] Martin Verlage, “About views for modeling software processes in a role-specific manner”, *Proceedings of the second International Software Architecture Workshop*

- (ISAW-2) and International Workshop on Multiple Perspectives in Software Development (Viewpoints '96) on SIGSOFT '96 Workshops, October 1996, pp:280 – 284.
- [21] M.I. Kellner, “Representation formalisms for software process modeling”, *Proceedings of the 4th International Software Process Workshop on Representing and enacting the Software Process, Volume 14, Issue 4*, April 1988, pp:93 – 96.
- [22] Nguyen M.N., Conradi R., “Classification of meta-processes and their models”, *Proceedings of the third International Software Process Conference*, October, 1994, pp:167 – 175.
- [23] Osterweil L., Heimbigner D., “An alternative to software process languages”, *Proceedings of the ninth International Software Process Workshop*, October, 1994, pp:129 – 131.
- [24] Pfleeger S.L., “The nature of system change”, *IEEE Software, Volume 15, Issue 3*, May-June 1998, pp:87 – 90.
- [25] Pillai, K., Sukumaran Nair, V.S., “A model for software development effort and cost estimation”, *IEEE Transactions on Software Engineering*, Volume 23, Issue 8, August, 1997 pp: 485 – 49.
- [26] Ricardo A. Falbo, Giancarlo Guizzardi, Ana C. C. Natali, Gleidson Bertollo, Fabiano F. Ruy, Paula G. Mian, “Software process modelling: Towards semantic software engineering environments”, *Proceedings of the 14th international conference on Software engineering and knowledge engineering SEKE '02*, July, 2002.
- [27] Richard Hundhausen, “Working with Microsoft Visual Studio 2005 Team System”, MicroSoft.
- [28] Roger S. Pressman, “Software Engineering: A Practitioner's Approach, 6/e”, McGraw-Hill.
- [29] Sam Guckenheimer, Juan J. Perez, “Software Engineering with Microsoft Visual Studio Team System”, Addison Wesley.
- [30] Stanley M. Sutton, Dennis Heimbigner, Leon J. Osterweil, “APPL/A: a language for software process programming”, *ACM Transactions on Software Engineering*

and Methodology (TOSEM), Volume 4, Issue 3, July 1995, pp:221 – 286.

- [31] Tarhan A., Demirors E., Demirors O., “A distributed tool for commitment specification and management”, *Proceedings of the 25th EUROMICRO Conference, Volume 2, September, 1999, pp:210 – 217.*
- [32] Turpin, R., “A progressive software development lifecycle”, *Proceedings of the Second IEEE International Conference on Engineering of Complex Computer Systems, October, 1996, pp:208 – 211.*
- [33] Customizing Process Templates : [http://msdn2.microsoft.com/en-us/library/ms243782\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms243782(VS.80).aspx).
- [34] IDEF, <http://www.idef.com/>.
- [35] Java Excel API, <http://jexcelapi.sourceforge.net/>.
- [36] Microsoft Office 2003 XML Reference Schemas,
<http://www.microsoft.com/downloads/details.aspx?familyid=FE118952-3547-420A-A412-00A2662442D9&displaylang=en>
- [37] Petri nets, <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>.
- [38] Process Template, <http://msdn2.microsoft.com/en-us/teamsystem/aa718801.aspx>.
- [39] The Apache Xalan Project, <http://xalan.apache.org/>.
- [40] UML, <http://www.uml.org/>.
- [41] Visual Studio Team System 2005, [http://msdn2.microsoft.com/en-us/library/fda2bad5\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/fda2bad5(VS.80).aspx).
- [42] 周忠信, 劉駿馨, 陳皆曲, 吳偉旭, 鄭有進, “企業 e 化的生命週期管理系統”, *Proceedings of the Second Taiwan Conference on Software Engineering, June, 2006, pp:68 – 72.*

附錄一：ALM-XML 的 XML Schema 定義

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Project">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProjectDef">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name" type="xs:string"/>
              <xs:element name="Goals">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Goal" type="xs:string"
                      minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Description" type="xs:string"/>
              <xs:element name="Principles">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Principle" type="xs:string"
                      minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Disciplines">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Discipline" type="xs:string"
                      minOccurs="0" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="ArtifactDef">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ArtifactType" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Name" type="xs:string" />
                    <xs:element name="Description" type="xs:string"/>
                    <xs:element name="Styling" type="xs:string" />
                    <xs:element name="Url" type="xs:string" />
                    <xs:element name="Comment" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RoleDef">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Role" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Name" type="xs:string"/>
          <xs:element name="Description" type="xs:string"/>
          <xs:element name="Task" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Name"
                  type="xs:string"/>
                <xs:element name="EntryCriteria">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="EntryCriteria"
type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="Steps">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Step"
type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="ExitCriteria">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="ExitCriteria"
type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="ArtifactType"
                  type="xs:string"/>
                <xs:element name="Comment"
                  type="xs:string"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="OrgDef">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Location" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" type="xs:string"/>
            <xs:element name="Section" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>

```



```

<xs:element name="Task" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="IterationPoint"
                  type="xs:string"/>
      <xs:element name="Comment"
                  type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="CMDef">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Folder" minOccurs="0"
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="AADef">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Folder" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" type="xs:string"/>
            <xs:element name="Group" minOccurs="0"
                          maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Name" type="xs:string"/>
                  <xs:element name="Read">
                    <xs:complexType>
                      <xs:attribute name="Allow"
                                      type="xs:boolean" use="required"/>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="CheckOut">
                    <xs:complexType>
                      <xs:attribute name="Allow"
                                      type="xs:boolean" use="required"/>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="CheckIn">
                    <xs:complexType>
                      <xs:attribute name="Allow"
                                      type="xs:boolean" use="required"/>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="Lock">
                    <xs:complexType>

```



```

<xs:complexType>
  <xs:sequence>
    <xs:element name="IssueTracking" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Name" type="xs:string"/>
          <xs:element name="Steps">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Step" minOccurs="0"
                  maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Field" minOccurs="0"
            maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Name" type="xs:string"/>
                <xs:element name="Description"
                  type="xs:string"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Folder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="Comment" type="xs:string"/>
      <xs:element name="Children" minOccurs="0"
        maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="Folder" minOccurs="0"
              maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

附錄二：工程指引網站基本架構的 HTML 文件

EngineeringPracticePortal.html :

```
<html>
  <body onload="javascript:window.location =
    'Supporting%20Files\\
EngineeringPracticePortal.html';">
  </body>
</html>
```

Supporting Files\Artifact.html :

```
<html>
  <head>
    <title></title>
  </head>
  <frameset cols = "20,80">
    <frame src = "ArtifactLeft.html" name = "left">
    <frame src = "Disciplines.html" name = "right">
  </frameset>
</html>
```

Supporting Files\Issue.html :

```
<html>
  <head>
    <title></title>
  </head>
  <frameset cols = "20,80">
    <frame src = "IssueLeft.html" name = "left">
    <frame src = "Disciplines.html" name = "right">
  </frameset>
</html>
```

Supporting Files\Process.html :

```
<html>
  <head>
    <title></title>
  </head>
  <frameset cols = "20,80">
    <frame src = "ProcessLeft.html" name = "left">
    <frame src = "Disciplines.html" name = "right">
  </frameset>
</html>
```

Supporting Files\ EngineeringPracticePortal.html :

```
<html>
  <head>
    <title>Engineering Practice Portal</title>
  </head>
  <frameset rows = "20,80">
    <frame src = "TopLink.html" name = "top">
    <frame src = "ProjectIn.html" name = "down">
  </frameset>
</html>
```

Supporting Files\ProjectIn.html :

```
<html>
  <head>
    <title></title>
  </head>
  <frameset cols = "20,80">
    <frame src = "ProjectLeft.html" name = "left">
    <frame src = "ProjectDefault.html" name = "right">
  </frameset>
</html>
```

Supporting Files\Role.html :

```
<html>
  <head>
    <title></title>
  </head>
  <frameset cols = "20,80">
    <frame src = "RoleLeft.html" name = "left">
    <frame src = "Disciplines.html" name = "right">
  </frameset>
</html>
```

附錄三：案例-ALM-XML

```
<?xml version="1.0" encoding="big5" ?>
<Project xsi:noNamespaceSchemaLocation="ALMModel.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ProjectDef>
    <Name>MobileProject</Name>
    <Goals>
      <Goal>在兩個月內，以本專案五個成員，順利發展完成並發行。</Goal>
      <Goal>產品生命週期管理，需要系統化支援。</Goal>
    </Goals>
    <Description>
      MobileProject 預計開發一套行動數位相簿管理系統。由 SA 負責定義與確認產品需求。需求確認後，經由系統分析，確立系統架構、資料模型與測試劇情。在系統分析確認完成後，進行系統設計，產出每隻程式的細部規格與測試設計。程式設計完成後由整合人員則整合與所有測試。
    </Description>
    <Principles>
      <Principle>進度第一</Principle>
      <Principle>注意品質</Principle>
      <Principle>所有人員必須熟悉軟體工程管理，以及相關軟體生命週期管理系統的使用</Principle>
    </Principles>
    <Disciplines>
      <Discipline>必須遵循工作規範，確實執行份內負責的工作</Discipline>
      <Discipline>必須遵循工作規範，套用對應之軟體生命週期管理系統</Discipline>
      <Discipline>必須注意上下游合作關係與時程，即時完成工作並交付至正確位置</Discipline>
    </Disciplines>
  </ProjectDef>
  <ArtifactDef>
    <ArtifactType>
      <Name>需求規格表</Name>
      <Description>記錄所有功能需求以及非功能需求</Description>
      <Styling>遵照公司內部需求規格表格式</Styling>
      <Url>
        http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/Requirement/ReqDocExmample.doc
      </Url>
      <Comment></Comment>
    </ArtifactType>
    <ArtifactType>
      <Name>專案定義書</Name>
      <Description>記錄專案的估算、組織、人力、進度、管理指標與報表等</Description>
      <Styling>遵照公司內部專案定義文件格式</Styling>
      <Url>
        http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/ProjectManagement/ProjectDefinitionExample.doc
      </Url>
      <Comment></Comment>
    </ArtifactType>
    <ArtifactType>
      <Name>派工表</Name>
      <Description>記錄人員與時程的派工資訊</Description>
```

```

        <Styling>遵照公司內部派工表格式</Styling>
        <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/ProjectManagement/Development
    Project Plan.mpp
    </Url>
    <Comment></Comment>
</ArtifactType>
<ArtifactType>
    <Name>版本配置書</Name>
    <Description>記錄新版本配置關係的定義</Description>
    <Styling>遵照公司內部版本配置書格式</Styling>
    <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/ProjectManagement/ProductVersion.doc
    </Url>
    <Comment></Comment>
</ArtifactType>
<ArtifactType>
    <Name>配置規劃書</Name>
    <Description>記錄配置資料夾及其人員身分認證與授權內容的定義
</Description>
    <Styling>遵照公司內部配置規劃書格式</Styling>
    <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/ProjectManagement/CMDefinition.doc
    </Url>
    <Comment></Comment>
</ArtifactType>
<ArtifactType>
    <Name>系統分析書</Name>
    <Description>各模組與作業內容定義</Description>
    <Styling>遵照公司內部 SA 文件格式</Styling>
    <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/SystemAnalysis/SADocExmample.doc
    </Url>
    <Comment></Comment>
</ArtifactType>
<ArtifactType>
    <Name>程式規格書</Name>
    <Description>各程式細部規格定義</Description>
    <Styling>遵照公司內部 SD 文件格式</Styling>
    <Url>

http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/SystemDesign/SDDocExmample.doc
    </Url>
    <Comment></Comment>
</ArtifactType>
<ArtifactType>
    <Name>原始程式碼、編譯程式碼</Name>
    <Description>實作程式碼</Description>
    <Styling>遵照標準程式 coding style</Styling>
    <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/Program/ProgramExmample.doc

```

```

    </Url>
    <Comment></Comment>
  </ArtifactType>
  <ArtifactType>
    <Name>單元測試碼</Name>
    <Description>實作程式碼的單元測試碼</Description>
    <Styling>遵照測試碼 coding style</Styling>
    <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/Program/ProgramExmaple.doc
    </Url>
    <Comment></Comment>
  </ArtifactType>
  <ArtifactType>
    <Name>測試報告書</Name>
    <Description>測試規劃、測試結果與分析</Description>
    <Styling>遵照公司內部測試報告書格式</Styling>
    <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/Testing/TestReportExmaple.doc
    </Url>
    <Comment></Comment>
  </ArtifactType>
  <ArtifactType>
    <Name>客戶服務紀錄</Name>
    <Description>人、事、時、地、問題紀錄</Description>
    <Styling>遵照公司服務單格式</Styling>
    <Url>
http://vsts.aajc.csie.thu.edu.tw/sites/MobileProject/Service/SRExmaple.doc
    </Url>
    <Comment></Comment>
  </ArtifactType>
</ArtifactDef>
<RoleDef>
  <Role>
    <Name>SA</Name>
    <Description>系統分析師</Description>
    <Task>
      <Name>需求定義</Name>
      <EntryCriterias>
        <EntryCriteria>無</EntryCriteria>
      </EntryCriterias>
      <Steps>
        <Step>訪談關係人，建立功能需求與非功能需求。</Step>
        <Step>撰寫需求規格表。</Step>
        <Step>審核需求間的一致性、完整性以及可完成性，如有不合適的需求，重複上述步驟 1。</Step>
        <Step>需求規格表簽入對應資料夾。</Step>
      </Steps>
      <ExitCriterias>
        <ExitCriteria>需求規格表已簽入對應資料夾。</ExitCriteria>
      </ExitCriterias>
      <ArtifactType>需求規格表</ArtifactType>
      <Comment></Comment>
    </Task>
  </Role>

```

```

<Name>需求確認</Name>
<EntryCriterias>
  <EntryCriteria>需求定義已完成</EntryCriteria>
</EntryCriterias>
<Steps>
  <Step>說明需求規格表予相關之關係人。</Step>
  <Step>確認需求規格表內容無誤，各方簽名確認。</Step>
  <Step>如有不符需求處，修改需求規格表內容，重複上述步驟，直至各方簽名
確認。</Step>
  <Step>需求規格表簽入對應資料夾。</Step>
</Steps>
<ExitCriterias>
  <ExitCriteria>需求規格表完成確認。</ExitCriteria>
  <ExitCriteria>需求規格表已簽入對應資料夾。</ExitCriteria>
</ExitCriterias>
<ArtifactType>需求規格表</ArtifactType>
<Comment></Comment>
</Task>
<Task>
  <Name>需求變更</Name>
  <EntryCriterias>
    <EntryCriteria>無</EntryCriteria>
  </EntryCriterias>
  <Steps>
    <Step>說明需求變更予相關之關係人。</Step>
    <Step>確認需求變更為駁回、擱置或變更。定義變更優先順序，各方簽名確
認。</Step>
    <Step>修正需求規格表，同時簽入對應資料夾。</Step>
  </Steps>
  <ExitCriterias>
    <ExitCriteria>完成變更確認。</ExitCriteria>
    <ExitCriteria>需求規格表已簽入對應資料夾。</ExitCriteria>
  </ExitCriterias>
  <ArtifactType>需求規格表</ArtifactType>
  <Comment></Comment>
</Task>
<Task>
  <Name>系統分析</Name>
  <EntryCriterias>
    <EntryCriteria>需求確認已完成</EntryCriteria>
  </EntryCriterias>
  <Steps>
    <Step>簽出需求規格表。</Step>
    <Step>依照需求內容，進行分析，定義模組。</Step>
    <Step>針對各模組撰寫系統分析書。</Step>
    <Step>系統分析書簽入對應資料夾。</Step>
  </Steps>
  <ExitCriterias>
    <ExitCriteria>系統分析書已簽入對應資料夾。</ExitCriteria>
  </ExitCriterias>
  <ArtifactType>系統分析書</ArtifactType>
  <Comment></Comment>
</Task>
</Role>
<Role>
  <Name>SD</Name>

```

```

<Description>系統設計師</Description>
<Task>
  <Name>系統設計</Name>
  <EntryCriterias>
    <EntryCriteria>對應模組之系統分析已完成</EntryCriteria>
  </EntryCriterias>
  <Steps>
    <Step>簽出對應之系統分析書。</Step>
    <Step>依照系統分析書內容，進行系統細部設計。</Step>
    <Step>撰寫各程式規格書。</Step>
    <Step>將規格書簽入對應資料夾。</Step>
  </Steps>
  <ExitCriterias>
    <ExitCriteria>程式規格書已簽入對應資料夾。</ExitCriteria>
  </ExitCriterias>
  <ArtifactType>程式規格書</ArtifactType>
  <Comment></Comment>
</Task>
</Role>
<Role>
  <Name>PR</Name>
  <Description>程式設計師</Description>
  <Task>
    <Name>程式設計</Name>
    <EntryCriterias>
      <EntryCriteria>系統設計工作已完成</EntryCriteria>
    </EntryCriterias>
    <Steps>
      <Step>至對應資料夾簽出程式規格書。</Step>
      <Step>依照程式規格書，撰寫程式。</Step>
      <Step>進程式編譯。</Step>
      <Step>程式簽入對應資料夾。</Step>
    </Steps>
    <ExitCriterias>
      <ExitCriteria>程式通過編譯。</ExitCriteria>
      <ExitCriteria>程式已簽入對應資料夾。</ExitCriteria>
    </ExitCriterias>
    <ArtifactType>原始程式碼、編譯程式碼</ArtifactType>
    <Comment></Comment>
  </Task>
  <Task>
    <Name>單元測試</Name>
    <EntryCriterias>
      <EntryCriteria>程式設計已完成</EntryCriteria>
    </EntryCriterias>
    <Steps>
      <Step>撰寫單元測試程式碼。</Step>
      <Step>確認單元測試程式碼正確，如不正確，進行修正。</Step>
      <Step>至對應資料夾簽出程式碼。</Step>
      <Step>進行單元測試。</Step>
      <Step>測試成功，將通過測試的程式碼與單元測試程式碼同時簽入對應資料
夾。</Step>
      <Step>測試失敗，進程式修正，直至通過測試。</Step>
      <Step>規畫測試方式、內容、環境、人力與時程。</Step>
      <Step>簽出所有程式碼。</Step>
    </Steps>
  </Task>

```

```

        <Step>進行整合測試。 </Step>
        <Step>如果測試失敗，記錄測試結果，發起 Bug 議題，直至通過整合測試。
</Step>
        <Step>進行系統測試。 </Step>
        <Step>進行壓力測試。 </Step>
        <Step>進行負載測試。 </Step>
        <Step>進行安全測試。 </Step>
        <Step>進行迴歸測試。 </Step>
        <Step>記錄上述所有測試結果。 </Step>
        <Step>撰寫測試報告書。 </Step>
    </Steps>
    <ExitCriteria>
        <ExitCriteria>程式通過單元測試。 </ExitCriteria>
        <ExitCriteria>
            通過測試的程式碼與單元測試程式碼已簽入對應資料夾。
        </ExitCriteria>
        <ExitCriteria>通過測試程式碼已簽入對應資料夾。 </ExitCriteria>
        <ExitCriteria>測試報告書撰寫完成並已簽入對應資料夾。
    </ExitCriteria>
    </ExitCriteria>
    </ExitCriteria>
    </ExitCriteria>
    <ArtifactType>單元測試碼</ArtifactType>
    <Comment></Comment>
</Task>
</Role>
<Role>
    <Name>IR</Name>
    <Description>整合與測試人員</Description>
    <Task>
        <Name>測試管理</Name>
        <EntryCriteria>
            <EntryCriteria>符合版本配置之程式碼已簽入至對應的資料夾。
        </EntryCriteria>
    </EntryCriteria>
    </EntryCriteria>
    <Steps>
        <Step>規畫測試方式、內容、環境、人力與時程。 </Step>
        <Step>簽出所有程式碼。 </Step>
        <Step>進行整合測試。 </Step>
        <Step>如果測試失敗，記錄測試結果，發起 Bug 議題，直至通過整合測試。
    </Step>
        <Step>進行系統測試。 </Step>
        <Step>進行壓力測試。 </Step>
        <Step>進行負載測試。 </Step>
        <Step>進行安全測試。 </Step>
        <Step>進行迴歸測試。 </Step>
        <Step>記錄上述所有測試結果。 </Step>
        <Step>撰寫測試報告書。 </Step>
    </Steps>
    <ExitCriteria>
        <ExitCriteria>通過測試程式碼已簽入對應資料夾。 </ExitCriteria>
        <ExitCriteria>測試報告書撰寫完成並已簽入對應資料夾。
    </ExitCriteria>
    </ExitCriteria>
    </ExitCriteria>
    <ArtifactType>測試報告書</ArtifactType>
    <Comment></Comment>
</Task>

```

```

</Role>
</RoleDef>
<OrgDef>
  <Location>
    <Name>台灣</Name>
    <Section>
      <Name>T</Name>
      <Group>
        <Name>T1</Name>
        <Account>
          <Name>a1</Name>
          <Actor>a1</Actor>
          <Role>SA</Role>
          <Skills>
            <Skill>
              <Name>Analysis</Name>
              <SkillLevel>3</SkillLevel>
            </Skill>
          </Skills>
        </Account>
      </Group>
      <Group>
        <Name>T2</Name>
        <Account>
          <Name>d1</Name>
          <Actor>d1</Actor>
          <Role>SD</Role>
          <Skills>
            <Skill>
              <Name>Design</Name>
              <SkillLevel>3</SkillLevel>
            </Skill>
          </Skills>
        </Account>
      </Group>
      <Group>
        <Name>N1</Name>
        <Account>
          <Name>p1</Name>
          <Actor>p1</Actor>
          <Role>PR</Role>
          <Skills>
            <Skill>
              <Name>C#</Name>
              <SkillLevel>1</SkillLevel>
            </Skill>
            <Skill>
              <Name>JAVA</Name>
              <SkillLevel>2</SkillLevel>
            </Skill>
          </Skills>
        </Account>
      </Group>
      <Group>
        <Name>N2</Name>
        <Account>
          <Name>p2</Name>
          <Actor>p2</Actor>
          <Role>PR</Role>
          <Skills>

```

```

        <Skill>
          <Name>C#</Name>
          <SkillLevel>2</SkillLevel>
        </Skill>
        <Skill>
          <Name>JAVA</Name>
          <SkillLevel>3</SkillLevel>
        </Skill>
      </Skills>
    </Account>
  </Group>
</Section>
<Section>
  <Name>Q</Name>
  <Group>
    <Name>Q</Name>
    <Account>
      <Name>q1</Name>
      <Actor>q1</Actor>
      <Role>IR</Role>
      <Skills>
        <Skill>
          <Name>Testing</Name>
          <SkillLevel>4</SkillLevel>
        </Skill>
      </Skills>
    </Account>
  </Group>
</Section>
</Location>
</OrgDef>
<ProcessDef>
  <Phase>
    <Name>需求發展</Name>
    <Task>
      <Name>需求定義</Name>
      <IterationPoint></IterationPoint>
      <Comment></Comment>
    </Task>
    <Task>
      <Name>需求確認</Name>
      <IterationPoint></IterationPoint>
      <Comment></Comment>
    </Task>
  </Phase>
  <Phase>
    <Name>系統開發</Name>
    <Task>
      <Name>系統分析</Name>
      <IterationPoint></IterationPoint>
      <Comment></Comment>
    </Task>
    <Task>
      <Name>系統設計</Name>
      <IterationPoint></IterationPoint>
      <Comment></Comment>
    </Task>
    <Task>
      <Name>程式設計</Name>

```

```

        <IterationPoint></IterationPoint>
        <Comment></Comment>
    </Task>
    <Task>
        <Name>單元測試</Name>
        <IterationPoint></IterationPoint>
        <Comment></Comment>
    </Task>
</Phase>
<Phase>
    <Name>整合</Name>
    <Task>
        <Name>測試管理</Name>
        <IterationPoint></IterationPoint>
        <Comment></Comment>
    </Task>
</Phase>
<Phase>
    <Name>發行</Name>
    <Task>
        <Name>產品管理</Name>
        <IterationPoint></IterationPoint>
        <Comment></Comment>
    </Task>
</Phase>
<Phase>
    <Name>維護</Name>
    <Task>
        <Name>產品服務</Name>
        <IterationPoint></IterationPoint>
        <Comment></Comment>
    </Task>
</Phase>
</ProcessDef>
<CMDef>
    <Folder>
        <Name>Release</Name>
        <Comment>null</Comment>
        <Children></Children>
    </Folder>
    <Folder>
        <Name>Test</Name>
        <Comment>null</Comment>
        <Children></Children>
    </Folder>
    <Folder>
        <Name>Development</Name>
        <Comment>null</Comment>
        <Children>
            <Folder>
                <Name>N1</Name>
                <Comment></Comment>
                <Children></Children>
            </Folder>
            <Folder>
                <Name>N2</Name>
                <Comment></Comment>
                <Children></Children>
            </Folder>
        </Children>
    </Folder>

```

```

    </Children>
  </Folder>
</CMDef>
<AADef>
  <Folder>
    <Name>Development\N1</Name>
    <Group>
      <Name>T1</Name>
      <Read Allow="true" />
      <CheckOut Allow="true" />
      <CheckIn Allow="true" />
      <Lock Allow="true" />
    </Group>
    <Group>
      <Name>T2</Name>
      <Read Allow="true" />
      <CheckOut Allow="true" />
      <CheckIn Allow="true" />
      <Lock Allow="true" />
    </Group>
    <Group>
      <Name>Q</Name>
      <Read Allow="true" />
      <CheckOut Allow="true" />
      <CheckIn Allow="false" />
      <Lock Allow="false" />
    </Group>
    <Group>
      <Name>N1</Name>
      <Read Allow="true" />
      <CheckOut Allow="true" />
      <CheckIn Allow="true" />
      <Lock Allow="true" />
    </Group>
  </Folder>
  <Folder>
    <Name>Development\N2</Name>
    <Group>
      <Name>T1</Name>
      <Read Allow="true" />
      <CheckOut Allow="true" />
      <CheckIn Allow="true" />
      <Lock Allow="true" />
    </Group>
    <Group>
      <Name>T2</Name>
      <Read Allow="true" />
      <CheckOut Allow="true" />
      <CheckIn Allow="true" />
      <Lock Allow="true" />
    </Group>
    <Group>
      <Name>Q</Name>
      <Read Allow="true" />
      <CheckOut Allow="true" />
      <CheckIn Allow="false" />
      <Lock Allow="false" />
    </Group>
    <Group>
      <Name>N2</Name>

```

```

        <Read Allow="true" />
        <CheckOut Allow="true" />
        <CheckIn Allow="true" />
        <Lock Allow="true" />
    </Group>
</Folder>
<Folder>
    <Name>Test</Name>
    <Group>
        <Name>T1</Name>
        <Read Allow="true" />
        <CheckOut Allow="true" />
        <CheckIn Allow="true" />
        <Lock Allow="true" />
    </Group>
    <Group>
        <Name>Q</Name>
        <Read Allow="true" />
        <CheckOut Allow="true" />
        <CheckIn Allow="true" />
        <Lock Allow="true" />
    </Group>
</Folder>
<Folder>
    <Name>Release</Name>
    <Group>
        <Name>T1</Name>
        <Read Allow="true" />
        <CheckOut Allow="true" />
        <CheckIn Allow="true" />
        <Lock Allow="true" />
    </Group>
</Folder>
</AADef>
<TaskDef>
    <Task>
        <Name>A1</Name>
        <Phase>系統開發</Phase>
        <Area>台灣\T\N1</Area>
        <TaskType>系統分析</TaskType>
        <Skill>Analysis</Skill>
        <Level>2</Level>
        <Actor>a1</Actor>
        <Ref>Release\ReqDoc</Ref>
        <Artifact>N1SADoc</Artifact>
        <CheckIn>N1</CheckIn>
        <StartTime>2008/4/5 上午 09:00:00</StartTime>
        <EndTime>2008/4/5 下午 02:00:00</EndTime>
        <Cost>4</Cost>
        <Comment></Comment>
    </Task>
    <Task>
        <Name>A2</Name>
        <Phase>系統開發</Phase>
        <Area>台灣\T\N2</Area>
        <TaskType>系統分析</TaskType>
        <Skill>Analysis</Skill>
        <Level>2</Level>
        <Actor>a1</Actor>

```

```

<Ref>Release\ReqDoc</Ref>
<Artifact>N2SADoc</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/5 下午 02:00:00</StartTime>
<EndTime>2008/4/6 上午 10:00:00</EndTime>
<Cost>5</Cost>
<Comment></Comment>
</Task>
<Task>
  <Name>D11</Name>
  <Phase>系統開發</Phase>
  <Area>台灣\T\N1</Area>
  <TaskType>系統設計</TaskType>
  <Skill>Design</Skill>
  <Level>2</Level>
  <Actor>d1</Actor>
  <Ref>N1\N1SADoc</Ref>
  <Artifact>n11DesignDoc</Artifact>
  <CheckIn>N1</CheckIn>
  <StartTime>2008/4/5 下午 02:00:00</StartTime>
  <EndTime>2008/4/6 上午 11:00:00</EndTime>
  <Cost>6</Cost>
  <Comment></Comment>
</Task>
<Task>
  <Name>D12</Name>
  <Phase>系統開發</Phase>
  <Area>台灣\T\N1</Area>
  <TaskType>系統設計</TaskType>
  <Skill>Design</Skill>
  <Level>2</Level>
  <Actor>d1</Actor>
  <Ref>N1\N1SADoc</Ref>
  <Artifact>n12DesignDoc</Artifact>
  <CheckIn>N1</CheckIn>
  <StartTime>2008/4/6 上午 11:00:00</StartTime>
  <EndTime>2008/4/7 上午 11:00:00</EndTime>
  <Cost>8</Cost>
  <Comment></Comment>
</Task>
<Task>
  <Name>D13</Name>
  <Phase>系統開發</Phase>
  <Area>台灣\T\N1</Area>
  <TaskType>系統設計</TaskType>
  <Skill>Design</Skill>
  <Level>2</Level>
  <Actor>d1</Actor>
  <Ref>N1\N1SADoc</Ref>
  <Artifact>n13DesignDoc</Artifact>
  <CheckIn>N1</CheckIn>
  <StartTime>2008/4/7 上午 11:00:00</StartTime>
  <EndTime>2008/4/8 下午 12:00:00</EndTime>
  <Cost>9</Cost>
  <Comment></Comment>
</Task>
<Task>

```



```

<Name>D21</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N2</Area>
<TaskType>系統設計</TaskType>
<Skill>Design</Skill>
<Level>2</Level>
<Actor>d1</Actor>
<Ref>N2\N2SADoc</Ref>
<Artifact>n21DesignDoc</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/8 下午 01:00:00</StartTime>
<EndTime>2008/4/9 下午 03:00:00</EndTime>
<Cost>10</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>D22</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N2</Area>
<TaskType>系統設計</TaskType>
<Skill>Design</Skill>
<Level>2</Level>
<Actor>d1</Actor>
<Ref>N2\N2SADoc</Ref>
<Artifact>n22DesignDoc</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/9 下午 03:00:00</StartTime>
<EndTime>2008/4/10 下午 03:00:00</EndTime>
<Cost>8</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>D23</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N2</Area>
<TaskType>系統設計</TaskType>
<Skill>Design</Skill>
<Level>2</Level>
<Actor>d1</Actor>
<Ref>N2\N2SADoc</Ref>
<Artifact>n23DesignDoc</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/10 下午 03:00:00</StartTime>
<EndTime>2008/4/11 下午 03:00:00</EndTime>
<Cost>8</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>D24</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N2</Area>
<TaskType>系統設計</TaskType>
<Skill>Design</Skill>
<Level>2</Level>
<Actor>d1</Actor>
<Ref>N2\N2SADoc</Ref>
<Artifact>n24DesignDoc</Artifact>

```

```

<CheckIn>N2</CheckIn>
<StartTime>2008/4/11 下午 03:00:00</StartTime>
<EndTime>2008/4/12 下午 02:00:00</EndTime>
<Cost>7</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>P11</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N1</Area>
<TaskType>程式設計單元測試</TaskType>
<Skill>C#</Skill>
<Level>2</Level>
<Actor>p1</Actor>
<Ref>N1\n11DesignDoc</Ref>
<Artifact>n11</Artifact>
<CheckIn>N1</CheckIn>
<StartTime>2008/4/6 上午 11:00:00</StartTime>
<EndTime>2008/4/8 下午 03:00:00</EndTime>
<Cost>19</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>P12</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N1</Area>
<TaskType>程式設計單元測試</TaskType>
<Skill>C#</Skill>
<Level>2</Level>
<Actor>p1</Actor>
<Ref>N1\n12DesignDoc</Ref>
<Artifact>n12</Artifact>
<CheckIn>N1</CheckIn>
<StartTime>2008/4/8 下午 03:00:00</StartTime>
<EndTime>2008/4/10 上午 11:00:00</EndTime>
<Cost>13</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>P13</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N1</Area>
<TaskType>程式設計單元測試</TaskType>
<Skill>C#</Skill>
<Level>2</Level>
<Actor>p1</Actor>
<Ref>N1\n13DesignDoc</Ref>
<Artifact>n13</Artifact>
<CheckIn>N1</CheckIn>
<StartTime>2008/4/10 上午 11:00:00</StartTime>
<EndTime>2008/4/12 下午 02:00:00</EndTime>
<Cost>18</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>P21</Name>
<Phase>系統開發</Phase>

```

```

<Area>台灣\T\N2</Area>
<TaskType>程式設計單元測試</TaskType>
<Skill>C#</Skill>
<Level>2</Level>
<Actor>p2</Actor>
<Ref>N2\n21DesignDoc</Ref>
<Artifact>n21</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/9 下午 03:00:00</StartTime>
<EndTime>2008/4/11 下午 12:00:00</EndTime>
<Cost>14</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>P22</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N2</Area>
<TaskType>程式設計單元測試</TaskType>
<Skill>C#</Skill>
<Level>2</Level>
<Actor>p2</Actor>
<Ref>N2\n22DesignDoc</Ref>
<Artifact>n22</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/11 下午 01:00:00</StartTime>
<EndTime>2008/4/12 下午 05:00:00</EndTime>
<Cost>12</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>P23</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N2</Area>
<TaskType>程式設計單元測試</TaskType>
<Skill>C#</Skill>
<Level>2</Level>
<Actor>p2</Actor>
<Ref>N2\n23DesignDoc</Ref>
<Artifact>n23</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/12 下午 05:00:00</StartTime>
<EndTime>2008/4/14 下午 04:00:00</EndTime>
<Cost>15</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>P24</Name>
<Phase>系統開發</Phase>
<Area>台灣\T\N2</Area>
<TaskType>程式設計單元測試</TaskType>
<Skill>C#</Skill>
<Level>2</Level>
<Actor>p2</Actor>
<Ref>N2\n24DesignDoc</Ref>
<Artifact>n24</Artifact>
<CheckIn>N2</CheckIn>
<StartTime>2008/4/14 下午 04:00:00</StartTime>

```

```

    <EndTime>2008/4/16 下午 06:00:00</EndTime>
    <Cost>18</Cost>
    <Comment></Comment>
</Task>
<Task>
  <Name>T1</Name>
  <Phase>整合</Phase>
  <Area>台灣\Q</Area>
  <TaskType>整合測試</TaskType>
  <Skill>Testing</Skill>
  <Level>2</Level>
  <Actor>q1</Actor>
  <Ref>All codes</Ref>
  <Artifact>TestReport</Artifact>
  <CheckIn>Test</CheckIn>
  <StartTime>2008/4/17 上午 09:00:00</StartTime>
  <EndTime>2008/4/18 下午 06:00:00</EndTime>
  <Cost>16</Cost>
  <Comment></Comment>
</Task>
<Task>
  <Name>T2</Name>
  <Phase>整合</Phase>
  <Area>台灣\Q</Area>
  <TaskType>整合測試</TaskType>
  <Skill>Testing</Skill>
  <Level>2</Level>
  <Actor>q1</Actor>
  <Ref>All codes</Ref>
  <Artifact>TestReport</Artifact>
  <CheckIn>Test</CheckIn>
  <StartTime>2008/4/20 上午 09:00:00</StartTime>
  <EndTime>2008/4/22 上午 11:00:00</EndTime>
  <Cost>18</Cost>
  <Comment></Comment>
</Task>
<Task>
  <Name>T3</Name>
  <Phase>整合</Phase>
  <Area>台灣\Q</Area>
  <TaskType>系統測試</TaskType>
  <Skill>Testing</Skill>
  <Level>2</Level>
  <Actor>q1</Actor>
  <Ref>All codes</Ref>
  <Artifact>TestReport</Artifact>
  <CheckIn>Test</CheckIn>
  <StartTime>2008/4/23 上午 09:00:00</StartTime>
  <EndTime>2008/4/24 下午 06:00:00</EndTime>
  <Cost>16</Cost>
  <Comment></Comment>
</Task>
<Task>
  <Name>T4</Name>
  <Phase>整合</Phase>
  <Area>台灣\Q</Area>
  <TaskType>壓力測試</TaskType>

```

```

<Skill>Testing</Skill>
<Level>2</Level>
<Actor>q1</Actor>
<Ref>All codes</Ref>
<Artifact>TestReport</Artifact>
<CheckIn>Test</CheckIn>
<StartTime>2008/4/25 上午 09:00:00</StartTime>
<EndTime>2008/4/27 上午 11:00:00</EndTime>
<Cost>18</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>T5</Name>
<Phase>整合</Phase>
<Area>台灣\Q</Area>
<TaskType>負載測試</TaskType>
<Skill>Testing</Skill>
<Level>2</Level>
<Actor>q1</Actor>
<Ref>All codes</Ref>
<Artifact>TestReport</Artifact>
<CheckIn>Test</CheckIn>
<StartTime>2008/4/28 上午 09:00:00</StartTime>
<EndTime>2008/4/30 上午 12:00:00</EndTime>
<Cost>19</Cost>
<Comment></Comment>
</Task>
<Task>
<Name>T6</Name>
<Phase>整合</Phase>
<Area>台灣\Q</Area>
<TaskType>安全性測試</TaskType>
<Skill>Testing</Skill>
<Level>2</Level>
<Actor>q1</Actor>
<Ref>All codes</Ref>
<Artifact>TestReport</Artifact>
<CheckIn>Test</CheckIn>
<StartTime>2008/5/1 上午 09:00:00</StartTime>
<EndTime>2008/5/3 上午 10:00:00</EndTime>
<Cost>17</Cost>
<Comment></Comment>
</Task>
</TaskDef>
<PMDef>
<Indicator>
<Name>剩餘工作</Name>
<Goal>專案進度管理、專案人力運用管理</Goal>
<Description>了解還有多少工作沒做、何時可以完成。</Description>
<Comment></Comment>
</Indicator>
<Indicator>
<Name>專案速度</Name>
<Goal>專案進度管理</Goal>
<Description>了解團隊的生產力如何。</Description>
<Comment></Comment>
</Indicator>

```

```

<Indicator>
  <Name>未計畫工作</Name>
  <Goal>專案變動管理、專案人力運用管理</Goal>
  <Description>了解還有多少尚未規劃的工作。</Description>
  <Comment></Comment>
</Indicator>
<Indicator>
  <Name>Bug 率</Name>
  <Goal>專案進度管理、專案品質管理</Goal>
  <Description>了解在尋找、修正、和確認 bug 解決的效率如何。
</Description>
  <Comment></Comment>
</Indicator>
<Indicator>
  <Name>重新啟動</Name>
  <Goal>專案重工管理</Goal>
  <Description>了解做了多少虛工、解決了多少 bug。</Description>
  <Comment></Comment>
</Indicator>
<Indicator>
  <Name>Bug (依優先權)</Name>
  <Goal>專案品質管理</Goal>
  <Description>了解是否有找到正確的 bugs，並將他們適當地分級。
</Description>
  <Comment></Comment>
</Indicator>
</PMDef>
<IssueTrackingDef>
  <IssueTracking>
    <Name>需求議題</Name>
    <Steps>
      <Step>PM、SA、SD 與 SR 可提出需求變更。</Step>
      <Step>
        PM 審核此變更並決定是否接受。若接受此需求變更，則定義完成期限並提報派工
        議題；若不接受，PM 擱置或否決此變更議題。
      </Step>
      <Step>PM 根據變更需求，決定是否變更配置管理。當配置管理變更後，PM 通知所
      有相關人員。</Step>
      <Step>PM 關閉此議題。</Step>
    </Steps>
    <Field>
      <Name>ID</Name>
      <Description>需求議題代號</Description>
    </Field>
    <Field>
      <Name>Name</Name>
      <Description>需求議題名稱</Description>
    </Field>
    <Field>
      <Name>Description</Name>
      <Description>需求議題說明</Description>
    </Field>
    <Field>
      <Name>AssignTo</Name>
      <Description>指派的人員</Description>
    </Field>
    <Field>

```

```

    <Name>Priority</Name>
    <Description>議題的優先順序。1 最高，3 最低</Description>
</Field>
<Field>
    <Name>Response</Name>
    <Description>議題過程中，相關人員的所有回應紀錄</Description>
</Field>
<Field>
    <Name>ResponseBy</Name>
    <Description>對應 Response 的回應人員</Description>
</Field>
<Field>
    <Name>RDate</Name>
    <Description>對應 Response 的日期</Description>
</Field>
<Field>
    <Name>State</Name>
    <Description>
        議題狀態。分 [作用中]、[擱置]、[已解決]或[已關閉]等狀態
    </Description>
</Field>
<Field>
    <Name>DDate</Name>
    <Description>解決需求議題的目標日期</Description>
</Field>
<Field>
    <Name>CreateBy</Name>
    <Description>建立議題的人員</Description>
</Field>
<Field>
    <Name>CreateDate</Name>
    <Description>建立議題的時間</Description>
</Field>
<Field>
    <Name>ChangeBy</Name>
    <Description>目前為止，最後變更議題的人員</Description>
</Field>
<Field>
    <Name>ChangeDate</Name>
    <Description>目前為止，最後變更議題的日期</Description>
</Field>
<Field>
    <Name>ClosedBy</Name>
    <Description>關閉議題的人員</Description>
</Field>
<Field>
    <Name>CloseDate</Name>
    <Description>關閉議題的日期</Description>
</Field>
<Field>
    <Name>ResolvedBy</Name>
    <Description>解決議題的人員</Description>
</Field>
<Field>
    <Name>ResolveDate</Name>
    <Description>解決議題的日期</Description>
</Field>
<Field>

```

```

    <Name>Link</Name>
    <Description>相關工作項目、超連結或原始程式碼檔的連結</Description>
</Field>
<Field>
    <Name>Attachment</Name>
    <Description>議題的相關附件</Description>
</Field>
<Field>
    <Name>History</Name>
    <Description>議題所有欄位的歷史變更紀錄</Description>
</Field>
<Field>
    <Name>Comment</Name>
    <Description>其他說明</Description>
</Field>
</IssueTracking>
<IssueTracking>
    <Name>Bug 議題</Name>
    <Steps>
        <Step>PM、SA、SD、IR 與 SR 可提報 bug 議題。</Step>
        <Step>
            PM 審核此 bug 並決定是否接受。若確認為 bug，則在專案管理上，新增工作並派工給原開發人員，同時將此議題轉給該人員；若不接受為 bug，PM 回覆理由給原提辦人員。
        </Step>
        <Step>
            原開發人員收到 PM 轉送之 bug 議題，確認自己職責。如並非本人造成，退回 bug 議題給 PM，並說明原因與建議作法；如是本人造成，則修復 bug 後，回覆該 bug 議題給 PM。
        </Step>
        <Step>
            如果原提報人員確認修正完成，則通知 PM，PM 關閉此議題；如仍無法同意，原提報人員回報 PM，PM 再重複上述步驟，直到能夠順利關閉此議題為止。
        </Step>
    </Steps>
    <Field>
        <Name>ID</Name>
        <Description>Bug 議題代號</Description>
    </Field>
    <Field>
        <Name>Name</Name>
        <Description>Bug 議題名稱</Description>
    </Field>
    <Field>
        <Name>Description</Name>
        <Description>Bug 議題說明</Description>
    </Field>
    <Field>
        <Name>AssignTo</Name>
        <Description>指派的人員</Description>
    </Field>
    <Field>
        <Name>Priority</Name>
        <Description>議題的優先順序。1 最高，3 最低</Description>
    </Field>
    <Field>
        <Name>Response</Name>

```

```

    <Description>議題過程中，相關人員的所有回應紀錄</Description>
</Field>
<Field>
    <Name>ResponseBy</Name>
    <Description>對應 Response 的回應人員</Description>
</Field>
<Field>
    <Name>RDate</Name>
    <Description>對應 Response 的日期</Description>
</Field>
<Field>
    <Name>State</Name>
    <Description>
        議題狀態。分 [作用中]、[擱置]、[已解決]或[已關閉]等狀態
    </Description>
</Field>
<Field>
    <Name>DDate</Name>
    <Description>解決 Bug 議題的目標日期</Description>
</Field>
<Field>
    <Name>CreateBy</Name>
    <Description>建立議題的人員</Description>
</Field>
<Field>
    <Name>CreateDate</Name>
    <Description>建立議題的時間</Description>
</Field>
<Field>
    <Name>ChangeBy</Name>
    <Description>目前為止，最後變更議題的人員</Description>
</Field>
<Field>
    <Name>ChangeDate</Name>
    <Description>目前為止，最後變更議題的日期</Description>
</Field>
<Field>
    <Name>ClosedBy</Name>
    <Description>關閉議題的人員</Description>
</Field>
<Field>
    <Name>CloseDate</Name>
    <Description>關閉議題的日期</Description>
</Field>
<Field>
    <Name>ResolvedBy</Name>
    <Description>解決議題的人員</Description>
</Field>
<Field>
    <Name>ResolveDate</Name>
    <Description>解決議題的日期</Description>
</Field>
<Field>
    <Name>FoundInBuild</Name>
    <Description>這個欄位會顯示發現 Bug 的組建編號</Description>
</Field>
<Field>
    <Name>ResolvedInBuild</Name>

```

```
    <Description>這個欄位含有已解決 Bug 的組建編號</Description>
</Field>
<Field>
  <Name>TestName</Name>
  <Description>這個欄位可識別與這個 Bug 相關聯的測試名稱</Description>
</Field>
<Field>
  <Name>TestID</Name>
  <Description>這個欄位可識別與這個 Bug 相關聯的測試 ID</Description>
</Field>
<Field>
  <Name>TestPath</Name>
  <Description>這個欄位可識別與這個 Bug 相關聯的測試路徑</Description>
</Field>
<Field>
  <Name>Link</Name>
  <Description>相關工作項目、超連結或原始程式碼檔的連結</Description>
</Field>
<Field>
  <Name>Attachment</Name>
  <Description>議題的相關附件</Description>
</Field>
<Field>
  <Name>History</Name>
  <Description>議題所有欄位的歷史變更紀錄</Description>
</Field>
<Field>
  <Name>Comment</Name>
  <Description>其他說明</Description>
</Field>
</IssueTracking>
</IssueTrackingDef>
</Project>
```

附錄四：案例-ALM2VSTS 的 VSTS 流程範本主要產出

Classification.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<tasks>
  <task completionMessage="已建立 Team 專案結構。"
plugin="Microsoft.ProjectCreationWizard.Classification" name="正在建立
專案結構" id="UploadStructure">
  <taskXml>
    <Nodes>
      <Node Name="反覆項目" StructureType="ProjectLifecycle">
        <Children>
          <Node Name="需求發展" StructureType="ProjectLifecycle" />
          <Node Name="系統開發" StructureType="ProjectLifecycle" />
          <Node Name="整合" StructureType="ProjectLifecycle" />
          <Node Name="發行" StructureType="ProjectLifecycle" />
          <Node Name="維護" StructureType="ProjectLifecycle" />
        </Children>
      </Node>
      <Node Name="區域" StructureType="ProjectModelHierarchy">
        <Children>
          <Node Name="台灣" StructureType="ProjectModelHierarchy">
            <Children>
              <Node Name="T" StructureType="ProjectModelHierarchy">
                <Children>
                  <Node Name="T1"
StructureType="ProjectModelHierarchy" />
                  <Node Name="T2"
StructureType="ProjectModelHierarchy" />
                  <Node Name="N1"
StructureType="ProjectModelHierarchy" />
                  <Node Name="N2"
StructureType="ProjectModelHierarchy" />
                </Children>
              </Node>
              <Node Name="Q" StructureType="ProjectModelHierarchy">
                <Children />
              </Node>
            </Children>
          </Node>
        </Children>
      </Node>
    </Nodes>
    <properties>
      <property isFile="true"
value="Classification\FileMapping.xml" name="MSPROJ" />
    </properties>
  </taskXml>
</task>
</tasks>
```

GroupsandPermissions.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<tasks>
```

```

    <task completionMessage="已建立群組和權限。"
plugin="Microsoft.ProjectCreationWizard.Groups" name="建立群組和權限"
id="GroupCreation1">
    <taskXml>
        <groups>
            <group description="" name="T1">
                <permissions>
                    <permission allow="true" class="PROJECT"
name="GENERIC_READ" />
                </permissions>
            </group>
            <group description="" name="T2">
                <permissions>
                    <permission allow="true" class="PROJECT"
name="GENERIC_READ" />
                </permissions>
            </group>
            <group description="" name="N1">
                <permissions>
                    <permission allow="true" class="PROJECT"
name="GENERIC_READ" />
                </permissions>
            </group>
            <group description="" name="N2">
                <permissions>
                    <permission allow="true" class="PROJECT"
name="GENERIC_READ" />
                </permissions>
            </group>
            <group description="" name="Q">
                <permissions>
                    <permission allow="true" class="PROJECT"
name="GENERIC_READ" />
                </permissions>
            </group>
        </groups>
    </taskXml>
</task>
</tasks>

```

workitems.xml :

```

<?xml version="1.0" encoding="UTF-8"?>
<tasks>
    <task completionMessage="已建立 WorkItemType"
plugin="Microsoft.ProjectCreationWizard.WorkItemTracking"
name="WorkItemType definitions" id="WITs">
    <taskXml>
        <WORKITEMTYPES>
            <WORKITEMTYPE
                fileName="WorkItem Tracking\TypeDefinitions\Bug.xml" />
            <WORKITEMTYPE
                fileName="WorkItem Tracking\TypeDefinitions\Task.xml" />
            <WORKITEMTYPE
                fileName="WorkItem Tracking\TypeDefinitions\Issue.xml" />
            <WORKITEMTYPE
                fileName="WorkItem Tracking\TypeDefinitions\ReqIssue.xml" />
            <WORKITEMTYPE
                fileName="WorkItem Tracking\TypeDefinitions\TaskIssue.xml"
/>

```

```

        <WORKITEMTYPE
            fileName="WorkItem
Tracking\TypeDefinitions\NewVerIssue.xml" />
        </WORKITEMTYPES>
    </taskXml>
</task>
<task completionMessage="已上載工作項目"
    plugin="Microsoft.ProjectCreationWizard.WorkItemTracking"
    name="WorkItems" id="WIs">
    <dependencies>
        <dependency taskId="WITs" />
    </dependencies>
    <taskXml>
        <WORKITEMS>
            <WI type="工作">
                <FIELD value="A1" refname="System.Title" />
                <FIELD value="$$PROJECTNAME$$\系統開發"
                    refname="System.IterationPath" />
                <FIELD value="作用中" refname="System.State" />
                <FIELD value="新增" refname="System.Reason" />
                <FIELD value="否"
                    refname="Microsoft.VSTS.Common.Issue" />
                <FIELD value="否"
                    refname="Microsoft.VSTS.Common.ExitCriteria" />
                <FIELD value="" refname="System.Description" />
                <FIELD value="$$PROJECTNAME$$\台灣\T\N1"
                    refname="System.AreaPath" />
                <FIELD value="4"
                    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
                <FIELD value="a1" refname="System.AssignedTo" />
                <FIELD value="2008/4/5 上午 09:00:00"
                    refname="Microsoft.VSTS.Scheduling.StartDate" />
                <FIELD value="2008/4/5 下午 02:00:00"
                    refname="Microsoft.VSTS.Scheduling.FinishDate" />
            </WI>
            <WI type="工作">
                <FIELD value="A2" refname="System.Title" />
                <FIELD value="$$PROJECTNAME$$\系統開發"
                    refname="System.IterationPath" />
                <FIELD value="作用中" refname="System.State" />
                <FIELD value="新增" refname="System.Reason" />
                <FIELD value="否"
                    refname="Microsoft.VSTS.Common.Issue" />
                <FIELD value="否"
                    refname="Microsoft.VSTS.Common.ExitCriteria" />
                <FIELD value="" refname="System.Description" />
                <FIELD value="$$PROJECTNAME$$\台灣\T\N2"
                    refname="System.AreaPath" />
                <FIELD value="5"
                    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
                <FIELD value="a1" refname="System.AssignedTo" />
                <FIELD value="2008/4/5 下午 02:00:00"
                    refname="Microsoft.VSTS.Scheduling.StartDate" />
                <FIELD value="2008/4/6 上午 10:00:00"
                    refname="Microsoft.VSTS.Scheduling.FinishDate" />
            </WI>
            <WI type="工作">

```

```

<FIELD value="D11" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\系統開發"
  refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N1"
  refname="System.AreaPath" />
<FIELD value="6"
  refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="d1" refname="System.AssignedTo" />
<FIELD value="2008/4/5 下午 02:00:00"
  refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/6 上午 11:00:00"
  refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="D12" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\系統開發"
  refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N1"
  refname="System.AreaPath" />
<FIELD value="8"
  refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="d1" refname="System.AssignedTo" />
<FIELD value="2008/4/6 上午 11:00:00"
  refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/7 上午 11:00:00"
  refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="D13" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\系統開發"
  refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N1"
  refname="System.AreaPath" />
<FIELD value="9"
  refname="Microsoft.VSTS.Scheduling.RemainingWork" />

```

```

<FIELD value="d1" refname="System.AssignedTo" />
<FIELD value="2008/4/7 上午 11:00:00"
  refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/8 下午 12:00:00"
  refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="D21" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
  <FIELD value="" refname="System.Description" />
  <FIELD value="$$PROJECTNAME$$\台灣\T\N2"
    refname="System.AreaPath" />
  <FIELD value="10"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
  <FIELD value="d1" refname="System.AssignedTo" />
  <FIELD value="2008/4/8 下午 01:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
  <FIELD value="2008/4/9 下午 03:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="D22" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
  <FIELD value="" refname="System.Description" />
  <FIELD value="$$PROJECTNAME$$\台灣\T\N2"
    refname="System.AreaPath" />
  <FIELD value="8"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
  <FIELD value="d1" refname="System.AssignedTo" />
  <FIELD value="2008/4/9 下午 03:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
  <FIELD value="2008/4/10 下午 03:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="D23" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />

```

```

<FIELD value="否"
  refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N2"
  refname="System.AreaPath" />
<FIELD value="8"
  refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="d1" refname="System.AssignedTo" />
<FIELD value="2008/4/10 下午 03:00:00"
  refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/11 下午 03:00:00"
  refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="D24" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
  <FIELD value="" refname="System.Description" />
  <FIELD value="$$PROJECTNAME$$\台灣\T\N2"
    refname="System.AreaPath" />
  <FIELD value="7"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
  <FIELD value="d1" refname="System.AssignedTo" />
  <FIELD value="2008/4/11 下午 03:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
  <FIELD value="2008/4/12 下午 02:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="P11" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
  <FIELD value="" refname="System.Description" />
  <FIELD value="$$PROJECTNAME$$\台灣\T\N1"
    refname="System.AreaPath" />
  <FIELD value="19"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
  <FIELD value="p1" refname="System.AssignedTo" />
  <FIELD value="2008/4/6 上午 11:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
  <FIELD value="2008/4/8 下午 03:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="P12" refname="System.Title" />

```

```

<FIELD value="$$PROJECTNAME$$\系統開發"
  refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N1"
  refname="System.AreaPath" />
<FIELD value="13"
  refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="p1" refname="System.AssignedTo" />
<FIELD value="2008/4/8 下午 03:00:00"
  refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/10 上午 11:00:00"
  refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="P13" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\系統開發"
  refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N1"
  refname="System.AreaPath" />
<FIELD value="18"
  refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="p1" refname="System.AssignedTo" />
<FIELD value="2008/4/10 上午 11:00:00"
  refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/12 下午 02:00:00"
  refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="P21" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\系統開發"
  refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
  refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N2"
  refname="System.AreaPath" />
<FIELD value="14"
  refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="p2" refname="System.AssignedTo" />

```

```

<FIELD value="2008/4/9 下午 03:00:00"
  refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/11 下午 12:00:00"
  refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="P22" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
  <FIELD value="" refname="System.Description" />
  <FIELD value="$$PROJECTNAME$$\台灣\T\N2"
    refname="System.AreaPath" />
  <FIELD value="12"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
  <FIELD value="p2" refname="System.AssignedTo" />
  <FIELD value="2008/4/11 下午 01:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
  <FIELD value="2008/4/12 下午 05:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="P23" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
  <FIELD value="" refname="System.Description" />
  <FIELD value="$$PROJECTNAME$$\台灣\T\N2"
    refname="System.AreaPath" />
  <FIELD value="15"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
  <FIELD value="p2" refname="System.AssignedTo" />
  <FIELD value="2008/4/12 下午 05:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
  <FIELD value="2008/4/14 下午 04:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="P24" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\系統開發"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"

```

```

    refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\T\N2"
    refname="System.AreaPath" />
<FIELD value="18"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="p2" refname="System.AssignedTo" />
<FIELD value="2008/4/14 下午 04:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/16 下午 06:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="T1" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\整合"
    refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\Q"
    refname="System.AreaPath" />
<FIELD value="16"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="q1" refname="System.AssignedTo" />
<FIELD value="2008/4/17 上午 09:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/18 下午 06:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="T2" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\整合"
    refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\Q"
    refname="System.AreaPath" />
<FIELD value="18"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="q1" refname="System.AssignedTo" />
<FIELD value="2008/4/20 上午 09:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/22 上午 11:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="T3" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\整合"

```

```

    refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\Q"
    refname="System.AreaPath" />
<FIELD value="16"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="q1" refname="System.AssignedTo" />
<FIELD value="2008/4/23 上午 09:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/24 下午 06:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="T4" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\整合"
    refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\Q"
    refname="System.AreaPath" />
<FIELD value="18"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="q1" refname="System.AssignedTo" />
<FIELD value="2008/4/25 上午 09:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/27 上午 11:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
<FIELD value="T5" refname="System.Title" />
<FIELD value="$$PROJECTNAME$$\整合"
    refname="System.IterationPath" />
<FIELD value="作用中" refname="System.State" />
<FIELD value="新增" refname="System.Reason" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
<FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
<FIELD value="" refname="System.Description" />
<FIELD value="$$PROJECTNAME$$\台灣\Q"
    refname="System.AreaPath" />
<FIELD value="19"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
<FIELD value="q1" refname="System.AssignedTo" />
<FIELD value="2008/4/28 上午 09:00:00"

```

```

        refname="Microsoft.VSTS.Scheduling.StartDate" />
<FIELD value="2008/4/30 上午 12:00:00"
        refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
<WI type="工作">
  <FIELD value="T6" refname="System.Title" />
  <FIELD value="$$PROJECTNAME$$\整合"
    refname="System.IterationPath" />
  <FIELD value="作用中" refname="System.State" />
  <FIELD value="新增" refname="System.Reason" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.Issue" />
  <FIELD value="否"
    refname="Microsoft.VSTS.Common.ExitCriteria" />
  <FIELD value="" refname="System.Description" />
  <FIELD value="$$PROJECTNAME$$\台灣\Q"
    refname="System.AreaPath" />
  <FIELD value="17"
    refname="Microsoft.VSTS.Scheduling.RemainingWork" />
  <FIELD value="q1" refname="System.AssignedTo" />
  <FIELD value="2008/5/1 上午 09:00:00"
    refname="Microsoft.VSTS.Scheduling.StartDate" />
  <FIELD value="2008/5/3 上午 10:00:00"
    refname="Microsoft.VSTS.Scheduling.FinishDate" />
</WI>
</WORKITEMS>
</taskXml>
</task>
<task completionMessage="已上載查詢"
  plugin="Microsoft.ProjectCreationWizard.WorkItemTracking"
  name="預存查詢定義" id="Queries">
  <dependencies>
    <dependency taskId="WIs" />
    <dependency taskId="WITs" />
  </dependencies>
  <taskXml>
    <QUERIES>
      <Query
        fileName="WorkItem Tracking\Queries\MyWorkItems.wiq" name="
我的工作項目" />
      <Query
        fileName="WorkItem Tracking\Queries\ProjectChecklist.wiq"
        name="專案檢查清單" />
      <Query
        fileName="WorkItem Tracking\Queries\AllScenarios.wiq"
name="所有案例" />
      <Query
        fileName="WorkItem
Tracking\Queries\AllQualityOfServiceRequirements.wiq"
        name="所有服務需求品質" />
      <Query fileName="WorkItem Tracking\Queries\AllTasks.wiq"
        name="所有工作" />
      <Query
        fileName="WorkItem Tracking\Queries\AllWorkItems.wiq"
        name="所有工作項目" />
      <Query
        fileName="WorkItem Tracking\Queries\ActiveBugs.wiq"

```

```
        name="作用中的 Bug" />
    <Query
        fileName="WorkItem Tracking\Queries\ResolvedBugs.wiq"
        name="已解決的 Bug" />
    <Query
        fileName="WorkItem
Tracking\Queries\MyWorkItemsAllTeamProjects.wiq"
        name="所有 Team 專案我的工作項目" />
    <Query
        fileName="WorkItem Tracking\Queries\AllIssues.wiq" name="所
有問題" />
    <Query
        fileName="WorkItem Tracking\Queries\UntriagedBugs.wiq"
        name="未分級的 Bug" />
    </QUERIES>
</taskXml>
</task>
</tasks>
```

