

國立交通大學

電機學院與資訊學院 資訊學程

碩士論文

「軟體專案開發程序」(SPDP)之設計

The Design of Software Project Development Process (SPDP)

研究生：李志宏

指導教授：鍾乾癸 教授

中華民國九十五年六月

「軟體專案開發程序」(SPDP)之設計
The Design of Software Project Development Process(SPDP)

研究生：李志宏

Student：Chih-Hung Li

指導教授：鍾乾癸

Advisor：Chyan-Goei Chung

國立交通大學
電機學院與資訊學院專班 資訊學程
碩士論文

A Thesis

Submitted to Degree Program of Electrical Engineering and Computer Science

College of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

國立交通大學

博碩士論文全文電子檔著作權授權書

(提供授權人裝訂於紙本論文書名頁之次頁用)

本授權書所授權之學位論文，為本人於國立交通大學電機資訊學院資訊學程碩士班 資訊組，九十四學年度第二學期取得碩士學位之論文。

論文題目：「軟體專案開發程序」(SPDP)之設計

指導教授：鍾乾癸 教授

同意 不同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：

本校及台灣聯合大學系統區域網路	<input checked="" type="checkbox"/> 中華民國 年 月 日公開
校外網際網路	<input checked="" type="checkbox"/> 中華民國 年 月 日公開

授權人：李志宏

親筆簽名：_____

中華民國九十五年六月二十二日

國立交通大學

博碩士紙本論文著作權授權書

(提供授權人裝訂於全文電子檔授權書之次頁用)

本授權書所授權之學位論文，為本人於國立交通大學電機資訊學院資訊學程碩士班 資訊組，九十四學年度第二學期取得碩士學位之論文。

論文題目：「軟體專案開發程序」(SPDP)之設計

指導教授：鍾乾癸 教授

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：_____，請將論文延至____年____月____日再公開。

授權人：李志宏

親筆簽名：_____

中華民國 年 月 日

國立交通大學

論文口試委員會審定書

本校 電機學院與資訊學院 資訊 組 李志宏 君
碩士在職專班

所提論文:(中文)「軟體專案開發程序」(SPDP)之設計

(英文)The Design of Software Project Development Process

(SPDP)

合於碩士資格水準、業經本委員會評審認可。

口試委員：_____

指導教授：_____

班主任：_____

中華民國 年 月 日

「軟體專案開發程序」(SPDP)之設計

學生：李志宏

指導教授：鍾乾癸

國立交通大學 電機學院與資訊學院 資訊學程(研究所)碩士班

摘 要

軟體開發專案管理不善常造成專案延誤、預算超支，及推出功能不完整、品質不佳的系統，加班趕程式進度更是軟體工程師常有的夢魘，此種現象在國內軟體業界更為常見。

近年來資訊業界試圖在軟體開發專案導入先進的「統一軟體開發流程」(USDP) [1]或專案管理方法-「專案管理知識體系」(PMBOK) [2]，但許多單位常因準備不周而告失敗，甚至未謀其利、先受其害。

本論文提出的「軟體專案開發程序」是融合了「統一軟體開發流程」及「專案管理知識體系」的優點，依實務需求將「統一軟體開發流程」之步驟更為細緻化，並制定相關管理表單以管制各階段工作之確實完成，經實際應用於政府機構軟體開發專案，已確認其實用性，且本文所提出「軟體專案開發程序」具有學習時間短，方法簡單，進度及風險易於掌握等優點，進而提昇專案的成功率。

The Design of Software Project Development Process(SPDP)

student : Chih-Hung Li

Advisors : Dr. Chyan-Goei Chung

Degree Program of Electrical Engineering and Computer Science
National Chiao Tung University

ABSTRACT

Worse Project Management usually together with delay schedule, over budget, releasing broken and poor quality systems, that is nightmarish for the IT industry, especially in our country.

In recently years, many local enterprises were tried to apply Unified Software Development Process (USDP) or Project Management Body of Knowledge (PMBOK) in software development project, however, most of them failed due to insufficient preparation.

This research proposed a Software Project Development Process (SPDP), which combines the advantage of USDP and PMBOK. The proposed process is based on the procedures of the Unified Software Development Process; however some project management related procedures and artifacts are added. This process was used by a local software company in executing a government-sponsored software project in 2005. From the result of pilot run, we found the proposed process has the advantages of ease to learning, ease of use, and ease of progress and risk control.

誌 謝

在交大修業期間的上百個不覺天已大白的日子，或讀書，或寫作業，或作研究，或寫論文，或趕著完成明天要交付的工作進度，每每心有戚戚，總想著在畢業論文的誌謝要寫些什麼來紀念這一段不會忘記的日子。在終將得償所願時，卻充滿著近鄉情怯而無處下筆...

得之於人者太多，感謝各位體恤我一邊工作一邊讀書的各位長官與同事，尤其是我的指導教授- 鍾乾癸 老師，如果沒有您的耐心、體諒與撥冗於百忙公務之中的悉心指導，這篇誌謝終將永遠沒有撰寫的可能性...

最後感謝我的親人，尤其是我的媽媽，從小撫養不太聽話的我長大，並且在我工作及課業繁忙的三年間幫忙照顧小女，很慚愧在小女出生後的日子，我才能體察為人父母的用心良苦，還有一直扮演我的好榜樣的哥哥跟姊姊，另外感謝內子體諒我無數不歸的日子，因為有你們的成全，我才能沒有後顧之憂地完成學業，最後對不起的是小女，常常在妳要騎牛牛時老爸已經累垮了，相信以後能有多一點時間陪伴妳的成長....

目 錄

中文提要.....	i
英文提要.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	v
圖目錄.....	vi
一、 緒論.....	1
1.1 研究背景與動機.....	1
1.2 章節介紹.....	4
二、 軟體專案開發與管理相關技術探討.....	5
2.1 開發技術「統一軟體開發流程」簡介.....	5
2.1.1 「統一軟體開發流程」歷史沿革與特性簡介.....	5
2.1.2 導入「統一軟體開發流程」的優點及挑戰.....	11
2.2 管理技術「專案管理知識體系」簡介.....	14
2.2.1 「專案管理知識體系」歷史沿革.....	14
2.2.2 「專案管理知識體系」簡介.....	14
2.2.3 導入「專案管理知識體系」於軟體開發專案的優點.....	19
2.2.4 直接運用「專案管理知識體系」於軟體開發專案的挑戰.....	21
三、 「軟體專案開發程序」之制定.....	22
3.1 「統一軟體開發流程」及「專案管理知識體系」的整合.....	22
3.2 「軟體專案開發程序」之各階段程序工作.....	25
3.2.1 起動階段(Inception Phase)之程序工作.....	25
3.2.2 詳述階段(Elaboration Phase)之程序工作.....	43
3.2.3 建構階段(Construction Phase)之程序工作.....	57
3.2.4 移交階段(Transition Phase)之程序工作.....	69
四、 「軟體專案開發程序」之成效評估及優缺點.....	79
4.1 「軟體專案開發程序」之成效評估.....	79
4.1.1 「軟體專案開發程序」之成效評估目的.....	79
4.1.2 「軟體專案開發程序」之成效評估作法.....	79
4.2 「軟體專案開發程序」之優缺點檢討.....	83
4.2.1 「軟體專案開發程序」的優點.....	83
4.2.2 「軟體專案開發程序」的缺陷.....	83
五、 結論.....	84

表 目 錄

表 1 軟體專案管理經歷的三個主要階段.....	2
表 2 SPDP起動階段計劃檢核表.....	34
表 3 SPDP起動階段-範疇評估管理表單.....	35
表 4 SPDP起動階段-成本分析表.....	36
表 5 SPDP起動階段-第一階段可行性評估報告表.....	37
表 6 SPDP起動階段-第二階段可行性評估報告表.....	37
表 7 SPDP起動階段-第三階段可行性評估報告表.....	38
表 8 SPDP起動階段-風險管理表.....	40
表 9 SPDP起動階段-採購項目確認表.....	41
表 10 SPDP詳述階段-風險管理表.....	51
表 11 SPDP詳述階段-採購進度管制表.....	51
表 12 SPDP詳述階段-專案成員技能評估表.....	52
表 13 SPDP詳述階段-專案成員技能訓練管制表.....	53
表 14 SPDP詳述階段-客戶配合工作管制表.....	53
表 15 SPDP詳述階段-成本預估基準表.....	54
表 16 SPDP建構階段-反覆管制表.....	64
表 17 SPDP建構階段-建構管理稽核表.....	65
表 18 SPDP建構階段-需求變更管制表.....	65
表 19 SPDP建構階段-開發環境稽核表.....	66
表 20 SPDP建構階段-工作改進管制表.....	66
表 21 SPDP建構階段-溝通矩陣表.....	67
表 22 SPDP建構階段-專案預備人員表.....	68
表 23 SPDP移交階段-使用者提出問題管理表.....	75
表 24 SPDP移交階段-交付項目管制表.....	76
表 25 SPDP移交階段-常見問題與建議解決方式表.....	76
表 26 SPDP移交階段-專案獲得經驗記錄表.....	77

圖 目 錄

圖 1 USDP五大工作流程在起動階段的資源分佈比重	8
圖 2 USDP五大工作流程在詳述階段的資源分佈比重	8
圖 3 USDP五大工作流程在建構階段的資源分佈比重	9
圖 4 USDP五大工作流程在移交階段的資源分佈比重	10
圖 5 USDP四大階段及五大工作流程的關係圖	10
圖 6 PMBOK五大程序群組間關係	17
圖 7 PMBOK不同階段間五大程序群組的關係	18
圖 8 PMBOK九大領域與五大程序群組之間的關係	18
圖 9 SPDP起動階段之工作流程	30
圖 10 SPDP詳述階段之工作流程	47
圖 11 SPDP建構階段-甘特圖(工作分解結構)	55
圖 12 SPDP建構階段-要徑圖	56
圖 13 SPDP建構階段之工作流程	61
圖 14 SPDP移交階段之工作流程	72



一、 緒論

1.1 研究背景與動機

資訊/軟體產業的發展至今已數十年，與其它產業動輒數百年(如：建築、機械…等)而至上千年的發展史相比(如：農業、造紙…等)，實屬於新興產業，然而其迅速發展帶動了各種不同領域之應用，對此，進步之影響與其它俱有悠久歷史的產業不惶多讓。過去其數十年，軟體專案開發與管理歷經了如下三個主要階段：

1. 1950 至 1960 年代中期：軟體開發僅屬少數專家的特定領域，一般而言，軟體開發者也擔任維護者的角色，品質保證的觀念維繫在專業的信任、良好的顧客關係及服務上，沒有品質確保與品質設計的觀念。
2. 1960 年代晚期至 1970 年代：由於企業管理資訊系統與資料庫應用的興起，軟體由藝術創作逐漸普及為企業應用的工具，促成專業軟體公司及套裝軟體的興起，並引進其它工業產品設計與生產經驗，包含開發方法、標準、審查制度、檢驗制度、開發工具等等，以加速軟體發展速度及提升軟體品質。
3. 1980 至 2000 年代初期：由於個人電腦與網際網路的普及，各種應用軟體需求殷切，縮短軟體開發的生命週期，軟體開發的競爭愈見激烈，開發成本及時程需有良好控制，以免失去市場競爭力，軟體工程的方法與工具已不足以應付如此複雜的問題，開始導入管理的方法。

此三個階段的特性可歸納如下表：

表 1 軟體專案管理經歷的三個主要階段

	第一階段	第二階段	第三階段
軟體類別	訂製型軟體 特殊功能軟體	套裝軟體 應用軟體	消費性軟體 大型整合性軟體 網路應用軟體
時間	1950 至 1960 年 代中期	1960 年代晚期至 1970 年代	1980 至 2000 年代初期
應用領域	科學 國防	企業管理 工程應用	個人工作 網路應用 整合性系統
硬體	處理能力小 價格昂貴	迷你電腦與大型電腦 集中處理為主	個人電腦與工作店 網路連接 分散式系統
軟體開發的重點	效率 程式撰寫技巧	產品功能 降低錯誤	品質 生產力 競爭力 創新
支援工具	非常少	逐漸增加	普遍受重視
軟體開發的原則	藝術與技藝	工程原則	市場導向 管理與技術並重
挑戰	程式撰寫技巧	結構化方法 程序改進 支援工具的開發	縮短開發時程 確保品質 知識的累積 創新

軟體專案管理雖已受重視，但仍未趨理想，由以下事實可了解其現況：

- Standish Group 研究：52.7% 資訊系統專案會透支預算或無法如期完成(On-time Deliver)，31.1% 會中斷進行，只有 16.2% 在原訂的預算下如期完成(Hayes，1997)[12]。
- 美國政府機關的所有的資訊專案中，只有 18% 在原訂的預算下如期完成(Davis & Wilder，1998)[13]。
- 1999 年 Information Week 的一篇論文研究了 100 家公司，發現只有 37% 以軟體為主的專案如期完成；42% 在原訂的預算下完成(Gordon，1999)[14]。

- 尤其以大型的軟體開發專案，失敗的機率更高達 85%(Ambler，1999)[15]。

作者在國內半導體產業的資訊部門擔任軟體開發工作及在軟體公司從事政府軟體專案開發近九年，深深感受軟體開發專案無法如期完成之苦，尤其是執行政府部門的軟體開發專案因行政流程導致開發時間較為短暫、專案需求明確程度較低，民間業者深受煎熬。然作者擔任資訊部門主管，不得不主動解決此問題，實有必要探討更適當的軟體專案開發程序以降低公司的風險。

作者學習「統一軟體開發流程」技術，並在公司推廣，部門同仁均利用此方法來開發軟體，但此方式在軟體專案管理方面的準則卻較少敘述，因此作者研讀「專案管理知識體系」一書後，深深感受實有必要結合二者，使公司同仁能更精確地在預定時間內完成專案。本研究即依此目的而設計，並在作者服務的公司試用。



1.2 章節介紹

本論文之章節安排如下：

第二章 對軟體開發技術之「統一軟體開發流程」及管理技術-「專案管理知識體系」進行探討，並分析以上關鍵技術直接運用於軟體開發專案之實務困難。

第三章 提出「統一軟體開發流程」及「專案管理知識體系」整合為「軟體專案開發程序」之可行性、方向、作法步驟及範疇，並訂定「軟體專案開發程序」之各階段工作，包含：目標、主要工作、主要產出、及工作表單與管理表單樣本。

第四章 為確保「軟體專案開發程序」之實用性，將「軟體專案開發程序」運用於實務專案，分析「軟體專案開發程序」所帶來之成效，並檢討應用「軟體專案開發程序」之優缺點。

第五章 為本論文之結論。



二、 軟體專案開發與管理相關技術探討

2.1 開發技術「統一軟體開發流程」簡介

2.1.1 「統一軟體開發流程」歷史沿革與特性簡介

1987年，歸納易利信(Ericsson)開發軟體系統的方式(Approach)，提出「Objectory Process 1.0」。

1996-1997年間，Ration整合1995年提出的「Objectory Process 3.8」，並結合「Rational Approach」及「UML」[3]，提出「Rational Objectory Process 4.1」。

後來由逐步整合各項新技術，如：資料工程、使用者介面設計、企業工程…等，提出「統一軟體開發流程」。

「統一軟體開發流程」有別於簡略的軟體生命週期(Software Lifecycle)，務實的將達成使用者需求轉換為軟體系統的過程所需要的工作項目予以有系統的整理歸納為一組流程及框架(Framework)，並得以應用於不同領域的軟體系統開發。

「統一軟體開發流程」歸納軟體的開發流程為五大工作流程及四大階段，其設計具有以下三大特性：

1. 以使用案例驅動(Use-Case Driven)：

傳統的需求搜集多以訪談、觀察資訊化前的作業方式及閱讀標準作業程序(SOP)等文件方式進行，在系統分析師歸納、吸收後以條列方式以系統功能為導向記錄以文字記錄需求成為需求規格書，然而此一方式先天存在許多重大的缺點，舉例如下：

- 需求規格書是以系統分析師的認知所列出的功能，因此系統分析師想像的系統常跟使用者預期的系統有重大的差異。
- 使用者對以系統功能為導向的需求規格書吸收、了解程度不佳，因此往往匆匆瀏覽過了事，等實際看到、操作到系統時才發覺不符需求。

有鑑於以上問題，「統一軟體開發流程」納入了使用案例(Use-Case)、參與者(Actor, Worker)、情節(Scenario)等概念，引導使用者在需求搜集時能很容易想像、領會未來系統之性質，並藉由使用者參與程度的提昇，在需求搜集階段即大幅減少使用者與系統開發者對系統需求的差異，有系統歸納「使用案例」作為系統規格，並提供後續設計、實作與測試等工作「需求追溯」，並以「使用案例」作為軟體開發過程之量化管理，並確保使用者提出需求已確實

解決。

2. 以架構為中心(Architecture Centric)：

在「統一軟體開發流程」推出前，開發系統之架構文件由於沒有一定的表達方式，往往僅以一張架構圖及數張簡單的流程圖代表，且在系統上線後即不再被維護，以致於未來進行系統規模擴增時，往往遭遇無從下手的困境，「統一軟體開發流程」配合「統一塑模語言」(Unified Modeling Language, UML)對系統架構進行完整的描述，以架構為中心實務上將帶來以下重要的好處：

- 系統架構及早確定，有助於確定系統之技術可行性及計畫時程之掌握，對於專案風險也較能掌握。
- 在系統規模擴增或需求變更時，可以藉由這些圖形快速的找到切入點，並找出變更的方式。

3. 採反覆及增添方式進行(Iterative and Incremental)：

由於使用者及系統開發者對需求的認知確實存在相當的差異，且此差異會隨開發時間長度而增大，如以瀑布法(Waterfall Approach)方式進行系統開發，需經過長期開發過程後才發現需求的差異，必須付出重大的代價進行更改，或使用者被迫妥協使用一個不太符合需求的系統，此外，由於資訊技術的進展迅速，系統開發者隨之必須採取不甚熟悉的新技術進行系統開發，很多技術問題往往在程式撰寫(Implementation/Coding)或測試(Test)階段才發現，此時已沒有充足的應變時間，於是造成了時程的延宕。

有鑑於以上問題，「統一軟體開發流程」將整個系統切割成一個個小小的反覆，每次反覆產生一個可以執行的小版本，並透過每次的反覆增加系統的功能，此一作法讓使用者能在初期完成的小版本中及時發現與真實需求的差異，於是可以最快的速度進行修正，並預防尚未開發的部份產生相同的問題，此外，由於在初期的反覆(Iteration)即經歷整個系統的開發過程(需求搜求，分析，設計，程式撰寫，測試)，故能及早發現技術問題並從容進行排除。

● 「統一軟體開發流程」的五大工作流程(Workflow)

1. 需求搜集(Requirements)工作流程

需求搜集工作流程的目的為尋求使用者與開發者對系統的共識、精確捕捉使用者的實際需求、界定系統的範圍，作為評估開發成本與時間的基礎，其工作包含定義系統願景(Vision)-包含以雛型(Prototyping)方式設計使用者介面(User Interface)，將願景轉換為使

用案例模型(Use-case Model)，並制定輔助規格(Supplementary Specification)，以確定系統之非功能性需求。

2. 分析(Analysis)工作流程

分析工作流程目的是把需求搜集工作流程的使用案例轉成軟體設計師(Software Designer)所關心的應用子系統與類別，藉以確定應用系統所必需包含之資料模型(Data Model)及執行流程。

3. 設計(Design)工作流程

配合系統佈署之硬體架構，制定軟體系統架構，進而設計各子系統之詳細邏輯。

4. 實作(Implementation)工作流程

為將設計工作流程的產出進程式撰寫、單元測試(Unit Test)。

5. 測試(Test)工作流程

將單元測試完成的類別進行整合，將整合完成的軟體進行指標測試(Benchmark Test)、組態測試(Configuration Test)、功能測試(Function Test)、安裝測試(Installation Test)、整體性測試(Integrity Test)、負載測試(Load Test)、效能測試(Performance Test)、壓力測試(Stress Test)等以確保產品品質。

● 「統一軟體開發流程」的四大階段(Phase)：

1. 起動階段(Inception Phase)

此階段的重點在了解專案背景及範疇，決定系統初步架構，進行少部份之設計實作以評估專案之可行性，並進行專案建議書撰寫，上述五大工作流程在本階段之資源分佈比重如下：

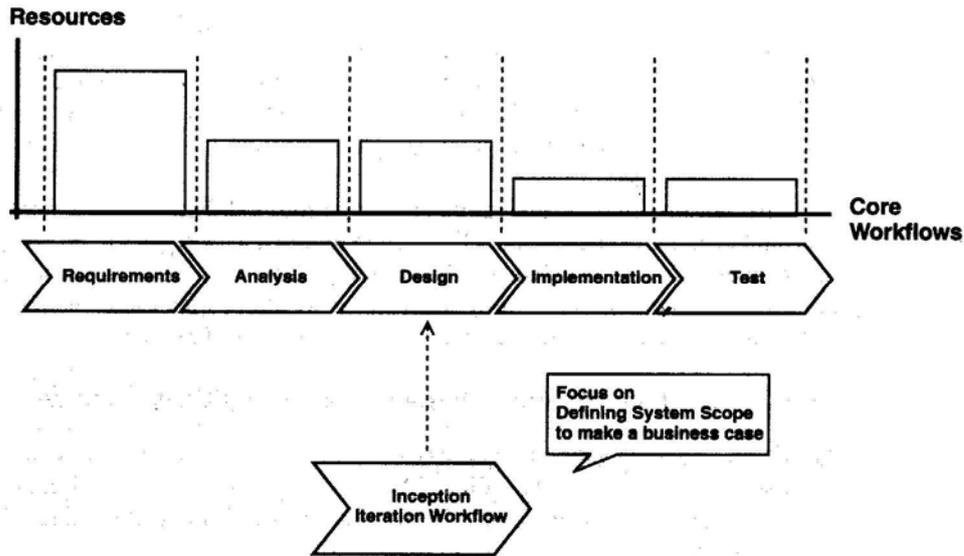


圖 1 USDP 五大工作流程在起動階段的資源分佈比重

資料來源：The Unified Software Development Process: Figure 13.1

2. 詳述階段(Elaboration Phase)

此階段的重點在制定系統之指標性架構，包含分析、設計及實作出架構的基準線，並嚐試對風險高的部份進行開發，以降低開發的風險，並學習使用特定工具與技術、據以安排後續工作時程，上述五大工作流程在本階段之資源分佈比重如下：

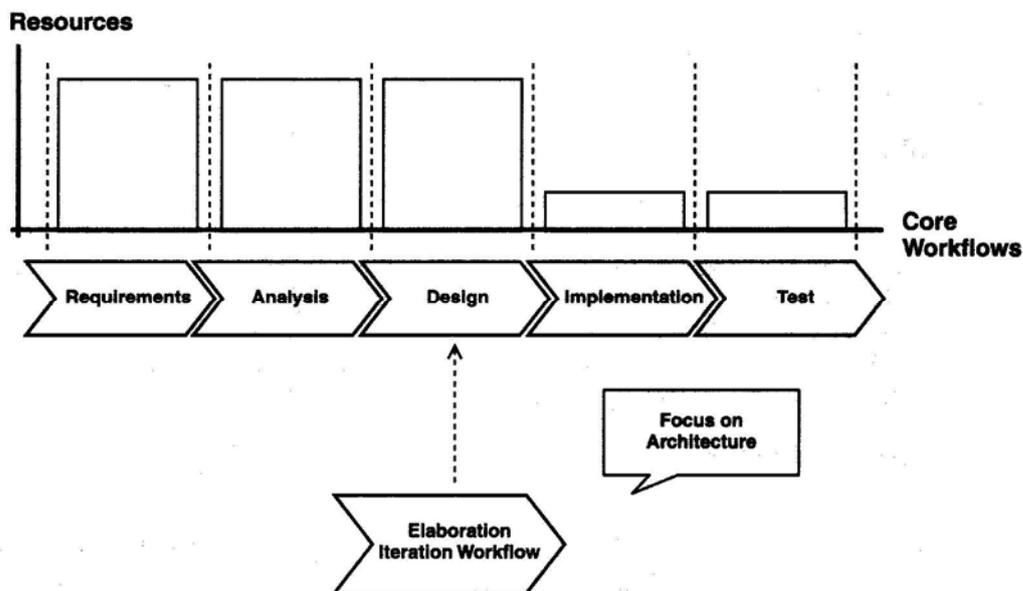


圖 2 USDP 五大工作流程在詳述階段的資源分佈比重

資料來源：The Unified Software Development Process: Figure 14.1

3. 建構階段(Construction Phase)

此階段的重點為實作出完整系統，包含所以系統功能之分析、設計及程式開發工作，延續上一階段的架構雛型(Prototype)，不斷附加、更新系統內容，使系統達成可以上線運作的目標，上述五大工作流程在本階段之資源分佈比重如下：

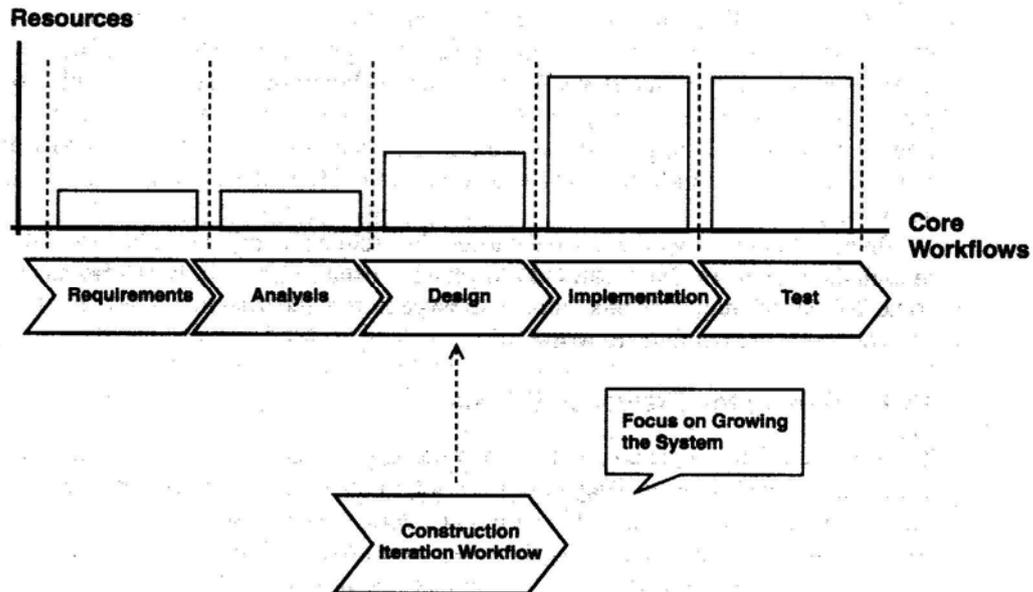


圖 3 USDP 五大工作流程在建構階段的資源分佈比重

資料來源：The Unified Software Development Process: Figure 15.1

4. 移交階段(Transition Phase)

此階段的重點在確保系統用戶可完全熟悉此系統之使用且軟體已達到足夠的品質水準，並符合軟體需求，工作內容包含修正錯誤、訓練終端使用者、調整系統加入缺少的必要功能以造出最終的軟體產品，及建立系統維運之機制以確保未來系統正常運作無虞。上述五大工作流程在本階段之資源分佈比重如下：

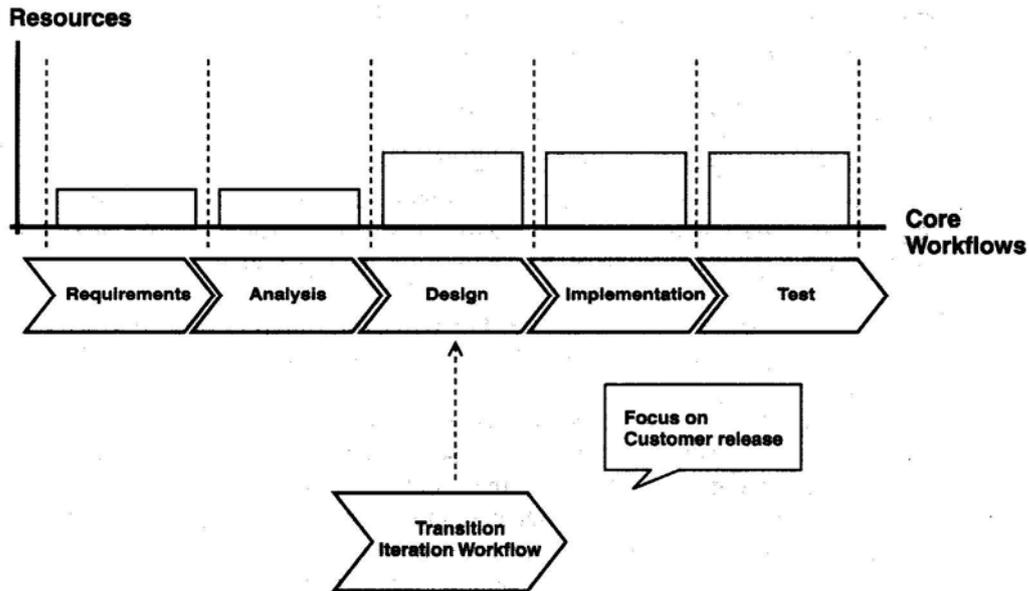


圖 4 USDP 五大工作流程在移交階段的資源分佈比重

資料來源：The Unified Software Development Process: Figure 16.1

- 上述四大階段及五大工作流程的關係可表示如下：

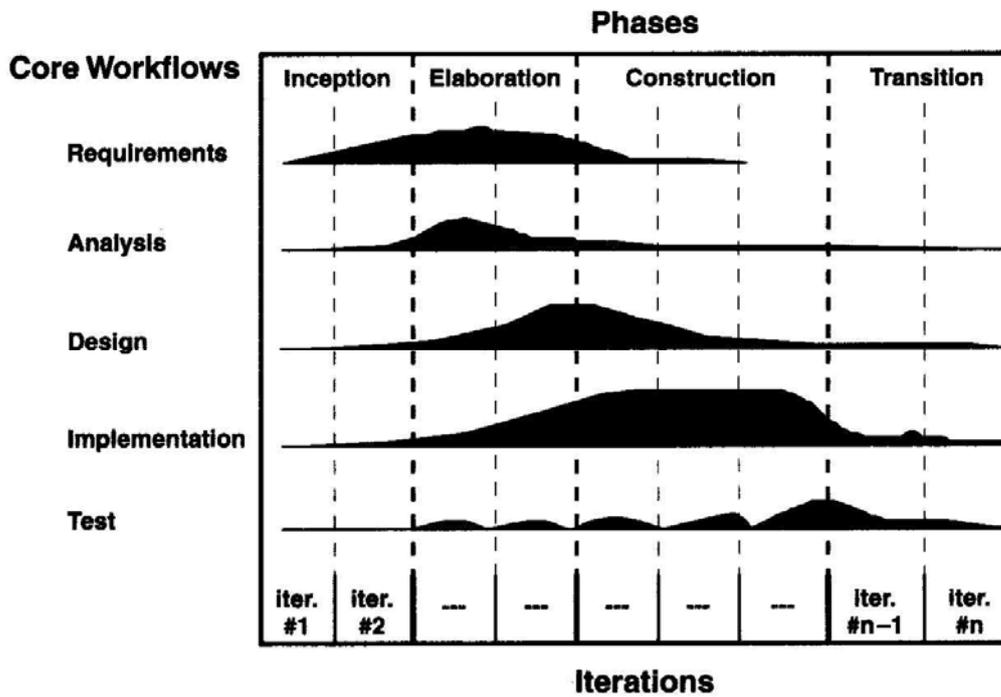


圖 5 USDP 四大階段及五大工作流程的關係圖

資料來源：The Unified Software Development Process: Figure 5.6

2.1.2 導入「統一軟體開發流程」的優點及挑戰

運用「統一軟體開發流程」於專案開發具有以下優點：

1. 「統一軟體開發流程」是結合理論與實務經驗的成果，且是在業界已被廣泛驗證的開發方式，可用以輔助軟體開發，快速增進軟體開發的品質。
2. 「統一軟體開發流程」以有組織、系統化的方式規劃軟體開發的製程，提高不同開發者進行軟體開發的品質一致性。
3. 「統一軟體開發流程」以使用者案驅動，有效建立使用者與開發者間的橋樑，並以視覺化的方式與使用者溝通需求。
4. 「統一軟體開發流程」以架構為中心，並搭配統一描述語言(UML)作為軟體開發過程的表達方式，與開發者間資訊交流的有效工具，並達成重要資訊保留的方式。
5. 「統一軟體開發流程」以反覆及增添方式進行軟體開發，改善傳統軟體開發生命週期(Life-cycle)的重大缺點。

「統一軟體開發流程」解決了許多重要的軟體專案問題，但是直接應用「統一軟體開發流程」於軟體開發專案仍存在許多挑戰如下：

1. 「統一軟體開發流程」著重於軟體開發問題的解決方案，然而許多軟體開發專案的問題存在於專案管理上，「統一軟體開發流程」對此部份問題相關著墨有限。
2. 「統一軟體開發流程」中提出許多先進的觀念與作法，改善軟體開發的品質(如：以反覆與增添方式進行開發)，然而，此一作法有別與傳統里程碑(Milestone)檢驗專案進度的方式，無疑增加了進行開發工作的負擔。
3. 「統一軟體開發流程」完整規範了軟體開發的過程，可有效增強專案的透明度，然而也需更巨細靡遺地進行專案規劃及控管，此舉對傳統的專案經理(Project Manager)產生了重大的挑戰，如果沒有更良好的管理技能訓練，在落實層面將遭遇困難。

4. 專案進度的透明度不佳：軟體開發專案不若其它產業的專案，如建築業的專案，可以用一眼看出挖地基、搭建模板、堆砌磚塊數量…等進度，軟體等進度，軟體開發專案常常必須等需求分析、系統設計、程式撰寫階段完成後，才能明確知道專案進度，且對於其階段品質，階段的過程中，對軟體開發專案進度往往只能用估計的方法自由心證。
5. 專案成員異動：成員異動，補充不易，且補充的人對專案開發流程及專案內容不熟悉，產生更多溝通問題。
6. 專案成員對專案成功要素(Critical Success Factor)的認知不一致：專案成員對專案成功要素往往存在相當大的認知差異，有的認為如期完成是最重要的，可能認為在預估成本內完成是最重要的，有的認為符合使用者需求及沒有軟體缺陷(Bug)是最重要的，共識不足，工作安排及執行均會產生溝通困難。
7. 脆弱的專案組織：專案成員(包含開發單位及需求單位)往往不是全職參與專案，有其平時必須進行的既定任務或同時參加多個專案，當多項工作同時進行時，常依個人或長官認知的優先度來安排工作，造成專案工作進度掌握不易，因而牽一髮而動全身，造成相關工作延遲。
8. 管理階層對專案要求不一致：由於缺乏一致的專案管理及文件標準，造成管理階層對專案的要求不一致，造成專案經理人或專案參與者的力量不能集中於專案的重要工作，必須花費很多力量作不同型式的報表或紙上作業(Paper Work)，因而分身乏術，這種情況尤其出現在規模較大的軟體開發專案或已經發生時程延遲的專案上。
9. 軟體開發專案時程估計過於樂觀：軟體開發專案往往依專案經理的經驗進行概念性估計，然而由於經驗所限或專案複雜度未被周詳考量，導致軟體開發專案時程延誤履見不鮮。
10. 使用者對軟體成本觀念認知不佳：使用者對修改軟體的觀念往往停留在”花一點時間改一下就好了”，加上軟體開發專案的透明程度不佳，頻頻提出修改需求，然而軟體的修改常牽涉到既定之後續工作及資源規劃、設計的概念及標準、規劃好的軟體架構…等，加上系統硬體的限制、要小心避免修改後衍生新問題、對其它功能產生影響…等，因而造成軟體開發專案預算頻頻超支及時程的延

誤。

上述問題經分析為專案管理的管理性問題，針對以上問題於以下管理技術「專案管理知識體系」進行探討，並提出解決方案。



2.2 管理技術「專案管理知識體系」簡介

2.2.1 「專案管理知識體系」歷史沿革

專案管理專家在 80 年代將專案管理技術區分為傳統專案管理與現代專案管理兩個不同階段，其最重要的里程碑是 1987 年由專案管理協會 (Project Management Institute，縮寫為 PMI) 集合眾多各領域專家窮十年之功制定的「專案管理知識體系」(Project Management Body of Knowledge，縮寫為 PMBOK)，「專案管理知識體系」經歷幾次大幅的修正，並於 2000 年的版本趨於完整。

2.2.2 「專案管理知識體系」簡介

「專案管理知識體系」將專案管理所需涵蓋的知識領域區分為九大知識領域，將專案進行過程為五大程序群組，茲將九大知識領域與五大知識群組介紹如下：

● 九大知識領域(Nine Knowledge Area)：

1. 專案整合管理 (Project Integration Management)：

目的在確保專案需執行的各項活動均能相互協調及有效整合的一系列程序與方法，其涉及在相互競爭的目標和備選方案間作一取捨，專案整合管理包含下列子程序：

- 專案計劃制定(Project Plan Development)。
- 專案計劃執行(Project Plan Execution)。
- 整合變更控制(Integration Change Control)。

2. 專案範疇管理(Project Scope Management)

目的在確保所有必要執行的工作事項均已被列入專案中，俾使專案能成功完成的一系列過程，其主要工作重點是定義及控制那些工作項目應該或不應該包含在專案之中，專案範疇管理包含下列子程序：

- 專案起始(Initiation)。
- 範疇計劃(Scope Planning)。
- 範疇定義(Scope Definition)。
- 範疇確認(Scope Verification)。
- 範疇變更控制(Scope Change Control)。

3. 專案時間管理(Project Time Management)：

目的在確保專案能如期完成的一系列過程，其包含將專案產出分解成一個個完成專案所必須包含的明確工作項目(活動)，安排它們間的相關工作順序，並對時間進行妥善估計，並在專案執行過程中進行管控，包含下列子程序：

- 活動定義(Activity Definition)。
- 活動排序(Activity Sequencing)。
- 活動時程估算(Activity Duration Estimation)。
- 時程制定(Schedule Development)。
- 時程控制(Schedule Control)。

4. 專案成本管理(Project Cost Management)：

目的在確保專案能夠在核定預算範圍內完成的一系列過程，包含規劃所需資源，進行成本預估，完成專案預算編列，並在專案過程中對成本進行控制，專案成本管理包含下列子程序：

- 資源規劃(Resource Planning)。
- 成本預估(Cost Estimating)。
- 預算編列(Cost Budgeting)。
- 成本控制(Cost Control)。

5. 專案品質管理(Project Quality Management)：

目的在確保專案能夠以規劃的品質滿足所有需求的一系列過程，其包括決定品質政策、目標和責任的所有活動項目，專案品質管理包含下列子程序：

- 品質規劃(Quality Planning)。
- 品質保證(Quality Assurance)。
- 品質管制(Quality Control)。

6. 專案人力資源管理(Project Human Resource Management)：

目的在確保以最有效的方式運用人員以完成專案工作與活動的一系列過程，其管理的對象為所有的專案利害關係人(Stakeholders)，包含出資人、顧客、合作夥伴及個別貢獻者，專案人力資源管理包含下列子程序：

- 組織規劃(Organizational Planning)。
- 人員獲得(Staff Acquisition)。
- 團隊制定(Team Development)。

7. 專案溝通管理(Project Communication Management)：

目的在確保能適時且妥當地將專案相關的資訊予以產生、蒐集、傳

送、儲存、彙整給所有專案關係人的一系列管理過程與方法，其目的在提供促使專案有關人員的知識及資訊可有效的連結，專案溝通管理包含下列子程序：

- 溝通規劃(Communication Planning)。
- 資訊散佈(Information Distribution)。
- 績效報告(Performance Reporting)。
- 結案管理(Administrative Closure)。

8. 專案風險管理(Project Risk Management)：

目的在以系統化方法辨識、分析、和解決專案各種風險問題的過程，包括將各種對專案目標達成有正面影響之所有事件的發生機率與其結果極大化，並使有負面影響之事件的發生機率與衝擊降低到最小，專案風險管理包含下列子程序：

- 風險管理規劃(Risk Management Planning)。
- 風險辨識(Risk Identification)。
- 風險定性分析(Qualitative Risk Analysis)。
- 風險定量分析(Quantitative Risk Analysis)。
- 風險回應規劃(Risk Response Planning)。
- 風險監控(Risk Management and Control)。

9. 專案採購管理(Project Procurement Management)：

目的在確保專案向外採購的物品或委外的服務能順利獲得，以「買」方的角度去制定必要的程序，包含下列子程序：

- 採購規劃(Procurement Planning)。
- 邀商規劃(Solicitation Planning)。
- 邀商作業(Solicitation)。
- 商源評選(Source Selection)。
- 履約管理(Contract Administration)。
- 合約終結(Contract Closeout)。

● 五大程序群組(Five Process Groups)：

所有的專案管理過程可被區分為五個程序群組，而每個程序群組都包含有一個以上的過程，五大程序群組為：

1. 起始程序(Initiating Processes)：

確認一專案或階段應開始進行並獲得執行之許可。

2. 計劃程序(Planning Processes)：

設計及維持一可實踐計劃以利具體執行足以滿足客戶需求的專案。

3. 執行程序(Executing Processes)：
運用及協調人及其資源以實現此專案目之所有活動。
4. 管制程序(Controlling Processes)：
藉品質監控及進度衡量以確保專案目標之有效達成，並視情況採必要的改善措施。
5. 結尾程序(Closing Processes)：
正式驗收專案的結果，並按計劃終結所有作業。

- 五大程序群組間關係可表示如下：

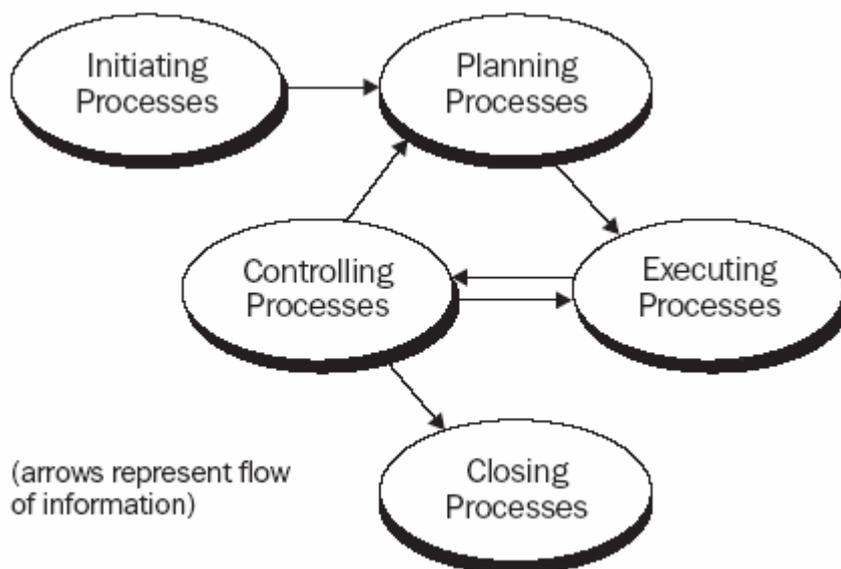


圖 6 PMBOK 五大程序群組間關係

資料來源：A Guide to the Project Management Body of Knowledge (PMBOK® Guide) 2000 Edition: Figure 3-1

- 一個專案間可能存在不同階段(Phase)，在不同階段間五大程序群組的關係可表示如下：

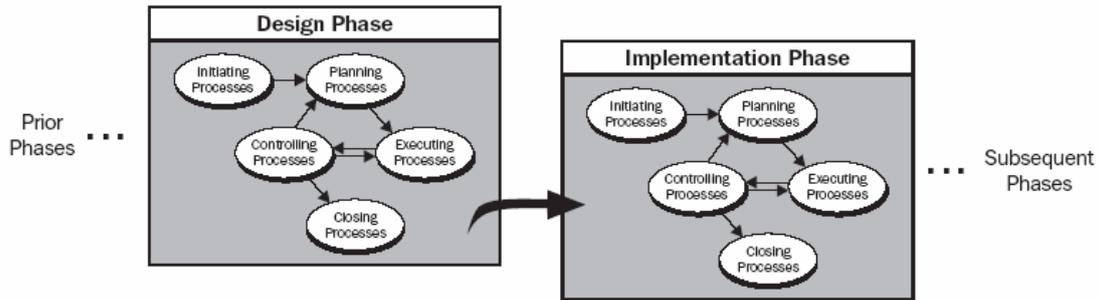


圖 7 PMBOK 不同階段間五大程序群組的關係

資料來源：A Guide to the Project Management Body of Knowledge (PMBOK® Guide) 2000 Edition: Figure 3-3

- 上述九大領域與五大程序群組之間的關係可以表示如下表：

Process Groups Knowledge Area	Initiating	Planning	Executing	Controlling	Closing
4. Project Integration Management		4.1 Project Plan Development	4.2 Project Plan Execution	4.3 Integrated Change Control	
5. Project Scope Management	5.1 Initiation	5.2 Scope Planning 5.3 Scope Definition		5.4 Scope Verification 5.5 Scope Change Control	
6. Project Time Management		6.1 Activity Definition 6.2 Activity Sequencing 6.3 Activity Duration Estimating 6.4 Schedule Development		6.5 Schedule Control	
7. Project Cost Management		7.1 Resource Planning 7.2 Cost Estimating 7.3 Cost Budgeting		7.4 Cost Control	
8. Project Quality Management		8.1 Quality Planning	8.2 Quality Assurance	8.3 Quality Control	
9. Project Human Resource Management		9.1 Organizational Planning 9.2 Staff Acquisition	9.3 Team Development		
10. Project Communications Management		10.1 Communications Planning	10.2 Information Distribution	10.3 Performance Reporting	10.4 Administrative Closure
11. Project Risk Management		11.1 Risk Management Planning 11.2 Risk Identification 11.3 Qualitative Risk Analysis 11.4 Quantitative Risk Analysis 11.5 Risk Response Planning		11.6 Risk Monitoring and Control	
12. Project Procurement Management		12.1 Procurement Planning 12.2 Solicitation Planning	12.3 Solicitation 12.4 Source Selection 12.5 Contract Administration		12.6 Contract Closeout

圖 8 PMBOK 九大領域與五大程序群組之間的關係

資料來源：A Guide to the Project Management Body of Knowledge (PMBOK® Guide) 2000 Edition: Figure 3-9

2.2.3 導入「專案管理知識體系」於軟體開發專案的優點

「專案管理知識體系」所揭櫫的重點為：明確設定專案目標、完善規劃完成專案所需的各項計劃、落實專案計劃中各項工作、以制度化的方式管控專案的變異及工作品質，並配合系統化方法累積專案經驗，以提高每一個專案的成功機率，其優點可歸納如下：

1. 增進專案任務的明確性：
藉由在專案計劃中制定明確的專案目標、範疇、時程、所需資源、各項專案管理計劃，使專案成員明確了解專案任務。
2. 增進專案進度的透明度：
透過客觀標準的專案評價手法(如：Earned Value Management)，定期檢視專案最重要的時程及成本。
3. 增進專案過程的秩序性：
經由排定的工作順序，避免因多項工作同時進行或多項工作需要相同資源所造成衝突。
4. 增進專案工作的完整性：
透過工作分解結構(Work Breakdown Structure)的手法分解專案的工作，並於各項工作訂定明確時程與可檢視的工作成果，避免因考慮不周詳而遺漏重要工作。
5. 突顯專案的重點工作：
藉由網路圖(Network Diagram)的手法找出專案要徑(Critical Path)上的工作，並投注心力於這些少數的重要工作，可以最小的成本大幅降低專案延遲的可能性。
6. 提高專案的品質：
藉由明確定義工作計劃及預期產生工作成果(Deliverable)、檢視(Review)專案中每一個里程碑(Milestone)的各項工作成果，確實因應專案的問題及風險，確保專案中各項工作均被確實執行。
7. 降低專案的風險：
主動發掘、量化、制定風險的回應計劃、採用解決、規避、移轉或接受等方式因應風險、並定期監控風險的狀態，以減少專案的非預

期因素，減少專案失敗的機率，進而提高專案成功率。

8. 促進專案經驗的累積：

以有系統的方式累積專案經驗，並經由回饋機制提昇專案管理的能力，藉由已累積專案經驗，避免因參與人員經驗不足而影響專案品質，及降低專案失敗的風險。



2.2.4 直接運用「專案管理知識體系」於軟體開發專案的挑戰

導入適當的管理方法確可降低軟體開發專案的風險問題，有效提昇專案的成功率，但在導入管理方法時，將面臨下列的挑戰：

1. 學習成本：

專案管理是一門相當專門的學問，國外有專門的碩士學程，要在組織內廣泛提昇專案管理的專業知識，必須耗費大量的人力及物力，業界希望避開艱深的理論研究、縮短或免除摸索或嚐試錯誤的時間，可以快速學習、馬上應用，加上導入初期，因技能不熟悉可能影響正在進行專案的品質，此為導入管理方法必須面對的實質挑戰。

2. 建立及推行組織軟體開發專案管理標準

由於「專案管理知識體系」係針對各行各業的專案廣泛適用的方向進行，並非針對軟體開發專案制定，導入此管理方法，於軟體開發專案管理必須投入大量資源以建立專案管理標準及程序，此牽涉高度的溝通、爭論及妥協，因此在建立及推行組織專案管理標準及程序的過程中，專案戰力的折損將是難以避免。

3. 增加額外的專案管理工作：

軟體專案管理的目的是透過有系統的方法妥善安排專案計劃，嚴密監控、追蹤專案工作進度，防範並提早排除可能導致專案失敗的因子，並累積專案管理的經驗，雖然這些方法雖然有效增進了專案成功的可能性，但無可避免的增加了專案管理工作的負擔及複雜性，如何適切引用管理法，盡可能降低額外負擔的情形，獲得專案最大的效益，亦為必須因應的重要挑戰。

三、「軟體專案開發程序」之制定

3.1 「統一軟體開發流程」及「專案管理知識體系」的整合

軟體開發專案中如果只擁有軟體開發技術，在進行小型軟體開發專案時還勉能負荷，隨著在進行的軟體開發專案規模漸漸增大、參與的人員與所需的資源增加時就會顯得力不從心或分身乏術，這樣的專案往往無法有通盤可行的計劃，只能見招拆招，專案落入就會必須草草收場或沒有結案日的牢籠；相反的，如果只擁有專案管理技術，往往因為紮實地以滿足使用者的方式完成軟體開發的各個工作流程：需求搜集- 分析- 設計- 實作- 測試，軟體開發的技術問題總是這類型專案延遲與超支的關鍵因素。

「統一軟體開發流程」聚焦於「軟體開發技術」，提出許多前瞻的務實作法，「專案管理知識體系」則以有系統的方式組織專案的流程、必須俱備的知識，這些技術在進行軟體開發專案時確可有互補的效果。

分析「統一軟體開發流程」及「專案管理知識體系」兩大技術後，訂定出整合方向如下：

1. 由於軟體開發專案在實務上通常具有分階段交付及付款之特性，整合後之「軟體專案開發程序」階段以「統一軟體開發流程」定義之「四大階段」為主。
2. 因「專案管理知識體系」之定義流程對象為任意行業之專案，「統一軟體開發流程」之定義流程對象為軟體開發，故整合後之程序以「統一軟體開發流程」規範之工作為主體，增加軟體公司承接業主之軟體標案進行系統開發所需額外具備工作，本整合需對上述工作進行整理、串連為可行之工作流程。
3. 因「專案管理知識體系」將任意行業專案之各項工作特性歸類為「五大程序」，以利使用分類之工作特性進行分別管理，本整合需將上述工作進行「五大程序」歸納，以利管理工作之進行。
4. 使用「專案管理知識體系」之「九大知識領域」進行本階段主要問題檢視，並分析提出解決方案。
5. 因應工作流程中以「專案管理知識體系」精神制定各階段之重要工

作表單與管理表單樣本，以符合業界對軟體開發專案的快速應用需求。

6. 充份考量軟體開發專案之「軟體開發技術」-「統一軟體開發流程」及「專案管理技術」-「專案管理知識體系」，制定「軟體專案開發程序」(Software Development Project Process)，簡寫為「SPDP」，以符合整合目的。

依上整合方向，本程序之各階段採以下六大作法步驟，進行程序制訂，茲分述如下：

1. 制訂階段之目標

為確保階段目標得以如質完成，本作法步驟之主要目標為明確定義各階段之目標。

2. 制訂階段之主要工作

為使專案對各階段之工作內容有所依循，本作法步驟之主要目標為以「統一軟體開發流程」之工作為主軸，增加軟體公司承接業主之軟體標案進行系統開發所需額外具備工作，進行明確定義各階段之各項主要工作。

3. 制訂階段之工作流程

為使業界能按部就班使用本程序進行專案，本作法步驟之主要目標為將上述之各項工作之優先順序確認後，以「專案管理知識體系」之「五大程序」之工作類型進行歸納、排列為本「軟體專案開發程序」之工作流程。

4. 制訂階段之主要工作產出

為確保各階段之主要工作得以如質完成，本作法步驟之主要目標為明確定義各階段工作之主要產出。

5. 制訂階段之管理問題分析

為避免專案執行過程因管理性問題而造成執行困難之可能性，本作法步驟之主要目標為利用「專案管理知識體系」之「九大知識領域」進行本階段主要問題檢視，並分析提出解決方案。

6. 制訂階段管理問題之解決方案及管理表單樣本

為符合業界對軟體開發專案的快速應用需求，本作法步驟之主要目

標為因應工作流程中以「專案管理知識體系」之精神制定各階段之重要管理表單樣本，包含常用之管理表單樣本、使用方法及使用時機說明。

「軟體專案開發程序」包含以下兩大部份：

- 流程：

以「統一軟體開發流程」之各項工作流程為主軸，增加軟體公司承接業主之軟體標案進行系統開發所需額外具備工作，將上述之各項工作之優先順序確認後，「專案管理知識體系」之五大程序之工作類型進行歸納、排列為本「軟體專案開發程序」之工作流程，本文第四章，我們將逐一介紹程序中四大階段之各項工作。

- 工作與管理表單：

除了在「軟體專案開發程序」中制定各項工作，為了使業界得以快速運用本程序，本程序亦包含常用的樣版，包含常用之管理表單樣本、使用方法及使用時機說明。



3.2 「軟體專案開發程序」之各階段程序工作

「軟體專案開發程序」之各階段皆包含四大階段，其目標、主要工作、工作流程、主要工作產出、可能問題及因應對策及管理表單如下：

3.2.1. 起動階段(Inception Phase)之程序工作

1. 起動階段之目標：

在招標性軟體開發專案的起動階段，由業主進行招標公告開始，至專案進行開標，確認是否得標為止，軟體公司在本階段必須完成以下重要工作項目：

- 確認專案的可行性。
- 提出專案架構的適用性。
- 辨識專案的風險。
- 預估專案之工作時程。
- 評估專案之工作成本。
- 提出專案建議書。

最後開標時若雀屏中選，則贏得軟體開發專案之合約。

2. 起動階段之主要工作：

依「統一軟體開發流程」(USDP)所述之主要工作，配合專案管理知識體系的精神，定義本階段工作如下：

● USDP-起動階段早期工作

(1)起動階段前置工作(Before the Inception Phase Begins; USDP-

13.2.1)：在業主公告招標後，專案業務人員於獲得招標資訊後進行領標工作，以取得各項招標文件，如：「招標須知」、「徵求需求說明書」、「招標投標及契約文件」...等，經業務及工程評估負責人詳閱後粗略評估公司參與該標案之可行性後，並考量公司策略，完成「第一階段可行性評估報告表」，以決定是否放棄投標或繼續進行後續工作，考量本程序定位，本工作名稱應調整為「招標文件取得與評估」。

(2)規劃起動階段(Planning the Inception Phase; USDP- 13.2.2)：若決定進行投標，則進行專案工作團隊成員工作分派及時程安排，通常由業務及研發主管共同協商，而後以開會方式向團隊成員說

明，並分發相關資料，即刻進行相關工作。

(3)延伸系統願景(Expanding the System Vision; USDP- 13.2.3)：本工作之主要作法通常包含以下步驟：

詳閱招標文件中「徵求需求說明書」(RFP)之「專案概述」、「交付產品項目」、「階段與時程」、「功能性需求」、「非功能性需求」、「公告預算金額」...等。在詳閱招標文件過程中，除對專案之概略性了解，另須彙整對本專案之相關疑問，並將疑問為兩大類-「不需訪談業主或專案相關人員疑問」及「需訪談業主或專案相關人員疑問」，並分別用以下兩種方式進行處理：

- A.「不需訪談業主或專案相關人員疑問」：此類問題之處理方式為搜集領域知識進行自修，請教領域專家...等。
- B.「需訪談業主或專案相關人員疑問」：此類問題之處理方式邀約並訪談業主或專案相關人員，進行此類疑問之釋疑，並進行專案背景與預期效益之了解，包含相關人員對專案之期望、範疇、需求重點及可能競爭者等資訊，與此專案有關之現有系統使用情況及問題搜集。

(4)可行性評估(Setting the Evaluation Criteria; USDP- 13.2.4)：當專案經理取得足夠之專案資訊後，會設定階段評估條件來進行可行性評估，一般常見之評估條件如下：

- A.確認系統範疇之了解程度(Resolve System Scope)。
- B.階段中模糊的需求之澄清程度(Resolve Ambiguities in the Requirements Needed in this Phase)。
- C.候選架構資訊是否明確(Establish a Candidate Architecture)。
- D.關鍵風險是否有明確的因應方案(Mitigate the Critical Risks)。
- E.初期的商業效益判斷是否已經完成(Judge the Worth of the Initial Business Case)。

在對上述條件進行初步評估後，進行「第二階段可行性評估報告表」撰寫，決定是否放棄投標或繼續進行後續工作。

● USDP-執行需求訪談至設計之核心工作

(5)獲得大部份系統需求(Capture the Requirements; USDP- 13.4.1)：

本工作為本階段之重點項目，主要目的為獲知系統需求，其工作主要包含四項行動(Activities)如下：

- A.表列候選需求(List Candidate Requirements)。
- B.了解系統內涵(Understanding the System Context)。
- C.獲得功能性需求(Capture Functional Requirements)。

D.獲得非功能性需求(Capture Nonfunctional Requirements)。

(6)候選架構分析(Analysis; USDP- 13.4.2)：本工作為將需求用物件模型(Object Model)作精確表達，產出粗略的分析模型(Analysis)作為規範候選架構之用，在業主未規範需進行雛型展示的案子中，只有少數把握度低及風險性高(約 10%)的需求將在本階段優先進行分析、精煉詳述完成，其它把握度高之需求則可跳過「分析」、「設計」、「實作」與「測試工作」，直接進行成本分析，本工作主要包含三項行動：

A.架構分析 (Architectural Analysis)。

B.分析使用個案(Analyze a Use Case)。

C.類別與包裝設計(Analyze a Class and Analyze a Package)。

(7)候選架構設計(Design; USDP- 13.4.3)：本工作為將分析工作產出轉換為設計模型(Design Model)的候選架構(Candidate Architecture)，在業主未規範需進行雛型展示的案子中，只對極少數把握度不足的使用者介面或演算法(約 5%)可使用快速開發技術來進行雛型設計，利用雛型展示使專案關係人(Stakeholders)相信專案團隊可達成專案的需求，並證實專案過程中不致因極少數把握度不足的使用者介面或演算法而失敗，本工作主要包含三項行動：

A.架構設計 (Architectural Design)，包括硬體架構及軟體架構之初步設計。

B.設計使用個案(Design a Use Case)。

C.設計類別與子系統(Design a Class and Design a Subsystem)。

(8)概念驗證實作(Implementation; USDP- 13.4.4)：專案經理可決對是否對以上極少數把握度不足的使用者介面或演算法進行雛型實作，以證實候選架構的可行性，如果考量資源有限性及把握度夠高，也可只產出候選架構描述即結束本階段，不需進行實作。

(9)概念驗證測試(Test; USDP- 13.4.5)：由於雛型展示之雛型有別於正在運作之系統，故有請測試工程師對實作之雛型進行測試之需求，並嚐試撰寫部份探索性之測試計劃。

● USDP-樹立初期的商業效益

(10) 描述評估成本(Outline Business Bid; USDP- 13.5.1)：對本專案進行成本估價，包含硬體、套裝軟體授權費用...等等外購項目之

詢價，人力成本則依軟體候選架構進行每一個子系統之人力預估後，加總各子系統之需求人日作為開發此系統所需人力，再乘以每人日之成本作為開發該子系統所需成本，此外也可利用本專案與過往之基準專案進行比較來決定商業價格，考量本程序定位，本工作名稱應調整為「進行成本分析」。

(11) 預估投資報酬率(Estimate Return on Investment; USDP-

13.5.2)：預估本專案開發軟體完成後，業主可從導入本專案開發軟體獲得之效益，促進業主委託本團隊進行軟體開發之意願，並就供、需因素決定投標價格，並完成「第三階段可行性評估報告表」，並考量專案投資報酬率，以決定是否放棄投標或繼續進行後續工作，考量本程序定位，本工作名稱應調整為「分析專案預期效益」。

● USDP-評價起始階段工作

(12) 評價起始階段(Assess the Iterations in the Inception Phase;

USDP- 13.6)：利用評估條件，對起始階段之成果進行評價，以確保階段目標已經完成，常見之評價條件如下：

- A. 確認系統範疇之了解程度。
- B. 階段中模糊的需求之澄清程度。
- C. 候選架構資訊是否明確。
- D. 關鍵風險是否有明確的因應方案。
- E. 初期的商業效益判斷是否已經完成。

● SPDP-因應本程序為軟體公司承接業主之軟體標案進行系統開發而須完成之工作

(13) 撰寫投標建議書：投標建議書一般由候選之專案經理進行撰寫，並由以下兩個階段完成：

- A. 於業務人員取得業主提供之徵求建議書後，依專案團隊之現有專案執行方式進行專案共通性章節之撰寫，如：「公司簡介」、「專案管理方法」、「建構管理方法」...等。
- B. 歸納上述各項技術評估工作後，依各項工作之產出進行專案特定性章節之撰寫，如：「解決方案建議」、「專案組織」、「成本分析」...等。

本工作除產出「投標建議書」外，另外會將「投標建議書」之重點製作為「建議書簡報」。

(14) 進行投標：在撰寫建議書後，於截標日前完成投標動作，並依

業主規定完成投標建議書介紹，以進行投標建議書評審、議價等動作。

(15) 建議書簡報：在完成投標及通過資格審查後，則會由業主召開建議書簡報會議，專案團隊將對評審進行建議書內容之簡報說明及必要之雛型展示。

(16) 接獲開標結果：由業主正式公佈評審結果及得標廠商，如果結果為得標，則進行議價、簽約，否則本專案即召開投標檢討會，檢討本次開標結果失敗原因供後續投標參考，並宣佈專案團隊本專案宣告終止。

(17) 議價及議約：在接獲開標結果後即展開議價之價格策略制定及專案合約審訂，並於議價及議約後進入後續詳述階段(Elaboration Phase)。

3. 起動階段之工作流程：

將上述之各項工作之優先順序確認後，「專案管理知識體系」之五大程序之工作類型進行歸納、排列為本起動階段之工作流程如下：



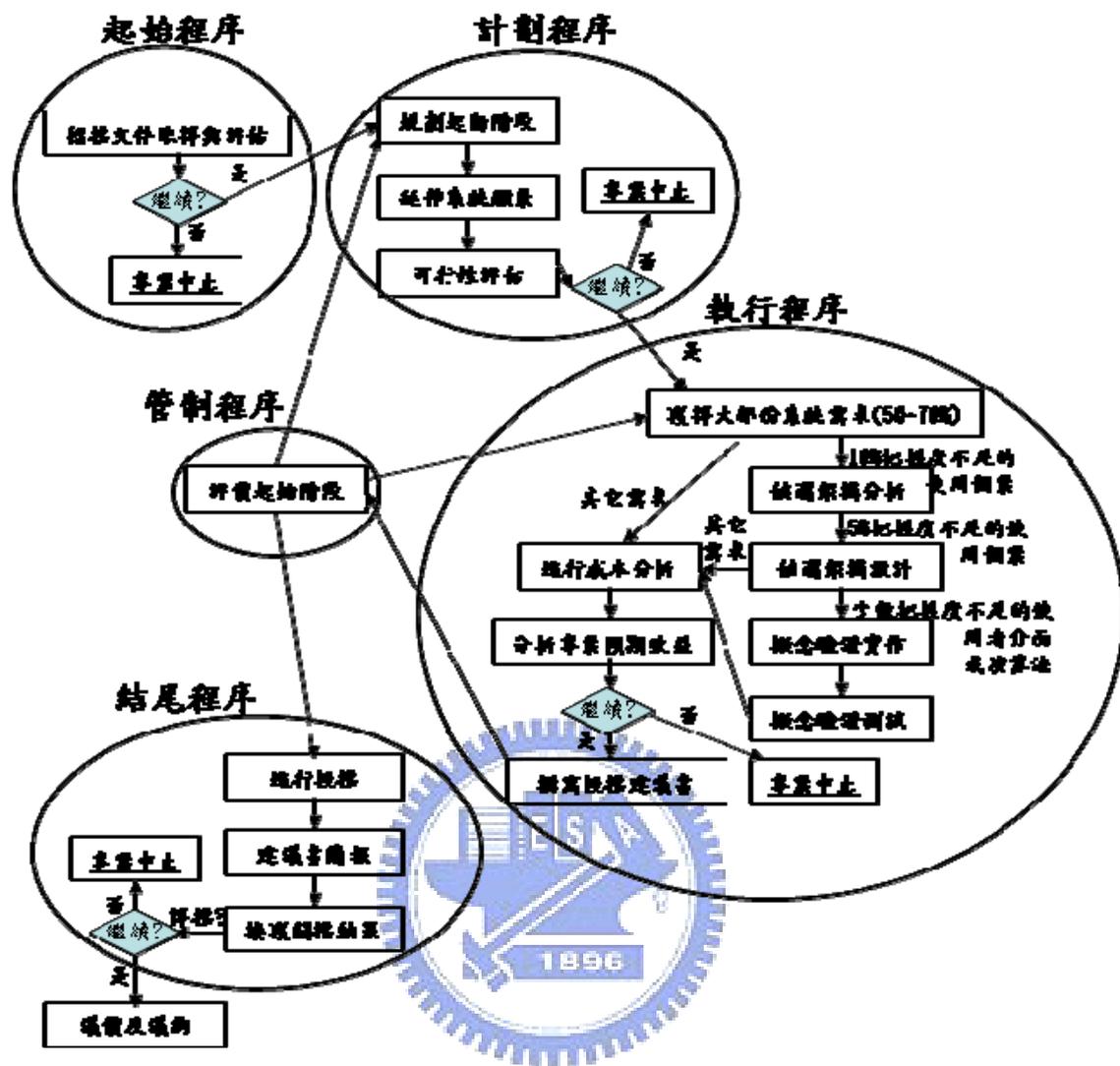


圖 9 SPDP 起動階段之工作流程

4. 起動階段之主要工作產出：

起動階段之主要交付項目(Artifact)為「投標建議書」及「建議書簡報」，除專案共通性章節，如：「公司簡介」、「專案管理方法」、「建構管理方法」...等可依專案團隊之現有專案執行方式進行撰寫，其它專案特定性章節將於前述起動階段之各項工作中陸續產出，起動階段各項工作之產出如下：

- (1) 招標文件取得與評估：業主提供之各項招標文件，如：「招標須知」、「徵求需求說明書」、「招標投標及契約文件」...等，團隊內部產出則為「第一階段可行性評估報告表」。
- (2) 規劃起動階段：起動階段之工作計劃。

- (3) 延伸系統願景：專案之「願景」、「範疇」、「需求重點」及「可能競爭者」...等資訊及「業務模型」，後續據以分析提出「投標建議書」中之「解決方案建議」。
- (4) 可行性評估：「階段評估條件列表」，用以確保達成階段之主要目標，團隊內部產出則為「第二階段可行性評估報告表」。
- (5) 獲得大部份系統需求：使用案例排序後之列表，此產出為「投標建議書」中「候選需求列表」、「系統內涵描述」、「功能性需求描述」、「非功能性需求描述」之主要內容。
- (6) 候選架構分析：定義之「子系統」及「分析模型」，後續據以分析提出「投標建議書」中之「解決方案建議」。
- (7) 候選架構設計(架構設計)：候選架構之「架構圖」，此產出為「投標建議書」中「系統架構建議」之重要項目。
- (8) 概念驗證實作：「使用者介面雛型」及「演算法雛型」，此產出為「建議書簡報」中「雛型展示」之主要內容。
- (9) 概念驗證測試：「雛型測試結果」與「測試計劃之樣本」，「測試計劃之樣本」為「投標建議書」中「測試方法說明」之主要內容。
- (10) 進行成本分析：「專案成本分析表」及「專案成本分析圖」，此產出為「投標建議書」中「專案成本分析」之內容。
- (11) 分析專案預期效益：「專案預期效益說明」，此產出為「投標建議書」中「投標價格」及「專案預期效益」之內容，團隊內部產出則為「第三階段可行性評估報告表」。
- (12) 評價起始階段：起始階段之工作內容及成果是否完成，且符合專案目標。
- (13) 撰寫投標建議書：「投標建議書」及「建議書簡報檔」，供「進行投標」及「建議書簡報」之用。
- (14) 進行投標：收文憑據/押標金收據。

(15) 建議書簡報：進行建議書簡報及回答評審詢問之問題。

(16) 接獲開標結果：是否得標之正式結果通知，如為得標，則通知議價及簽約時間。

(17) 議價及議約：專案之正式合約內容。

5. 起動階段之管理問題分析：

考量本階段限制及特性，彙整實務常見之管理問題如下：

(1)階段參與人員整合問題：各項工作此階段之時程緊迫，不確定因素多，且經由不同人負責，工作順序及成果又必須緊密整合，如無有效管理方式，可能因牽一髮而動全身，影響整體階段成果。

(2)階段範疇認知不當問題：階段評估之專案範疇廣泛且時間有限，如無有效管理方法，可能無法依範疇之掌握度而進行適切之深度評估，而導致部份重要專案範疇認知不當，小則影響開標結果，大則得標後因範疇認不當，而造成無法結案。

(3)階段時間短暫問題：本階段時程相當短暫，各項工作如無有效工作時間分配安排及管理方法，可能到本階段末期，才發現仍有許多工作尚未完成。

(4)階段評估專案成本客觀性問題：因成本評估人員之經驗或疏失而造成專案成本低估或部份成本項目漏估，導致實際專案執行成本大幅超出預期，甚至不符成本。

(5)對把握度不足專案投入之階段費用損耗問題：每個專案的投標成本相當高(投入各項工作之人力、建議書與簡報檔大量印製...等)，如果無論是否把握度如何皆進行投標將產生大量成本耗費。

(6)建議書提出架構無法滿足專案效能需求問題：由於本階段時間短暫，無論就時間或以成本考量都難以對建議書提出架構進行完整建置及效能驗證，因此可能提出之架構將無法滿足專案效能需求，造成專案驗收之困難。

- (7)階段參與人員之工作調配問題：因本階段部份參與之技術人員可能非專職進行本階段，因此可能因投入程度不足而無法產出完整且周全的建議書。
- (8)階段參與人員之間相互溝通問題：階段參與成員可能因工作默契不足，又因本階段時間相當有限，無法妥善進行相互溝通之磨合，導致因參與成員間之溝通方式不同，而造成資訊傳遞不良。
- (9)專案風險無法有效掌握及回應問題：本階段對專案存在之重大風險尚未完全搜集、辨識及客觀評價、尚未提出適當之因應計劃並有效管理，導致無法得標或得標後難以結案。
- (10) 建議提出之採購項目無法如期完成問題：建議提出之採購項目，包含：套裝軟體版權及硬體...等，因上游廠商之庫存或交期等因素無法如期完成採購，造成專案無法如期順利結案。

6. 起動階段管理問題之解決方案及管理表單樣本：

上述問題可以「專案管理知識體系」之「九大知識領域」進行歸納如下：

- 
- (1)階段參與人員整合問題：整合管理(Integration Management)問題。
- (2)階段範疇認知不當問題：範疇管理(Scope Management)問題。
- (3)階段時間短暫問題：時間管理(Time Management)問題。
- (4)階段評估專案成本客觀性問題：成本管理(Cost Management)問題。
- (5)對把握度不足專案投入之階段費用損耗問題：成本管理(Cost Management)問題。
- (6)建議書提出架構無法滿足專案效能需求問題：品質管理(Quality Management)問題。
- (7)階段參與人員之工作調配問題：人力資源管理(Human Resource

Management)問題。

(8)階段參與人員之間相互溝通問題：溝通管理(Communication Management)問題。

(9)專案風險無法有效掌握及回應問題：風險管理(Risk Management)問題。

(10)建議提出之採購項目無法如期完成問題：採購管理(Procurement Management)問題。

上述各項問題可以參考「專案管理知識體系」之「九大知識領域」，在階段各項工作提出解決方案及管理表單如下：

(1)階段參與人員整合問題：藉由制定「規劃起動階段」工作運用之「起動階段計劃檢核表」(請參閱表 2)，輔助階段中各項工作安排，並進行各項工作進度情況管理，其應用方法、時機及表單樣本如下：

A.應用方法：於「規劃起動階段」工作輔助階段中各項工作安排，並進行各項工作進度及情況管理。

B.應用時機：

a. 填表：於「規劃起動階段」工作開始時進行填寫。

b. 檢視及修改：每天定時進行實際工作情況檢討，並依實際工作情況進行內容調整。

C.表單樣本：

表 2 SPDP 起動階段計劃檢核表

專案名稱：

階段工作負責人：

項次	工作項目	工作負責人	其它參與人員	工作時程	交付成果項目	確認者
1	招標文件取得與評估			/ / ~ / /		
2	規劃起動階段			/ / ~ / /		
3	延伸系統願景			/ / ~ / /		
4	可行性評估			/ / ~ / /		
5	獲得大部份系統需求			/ / ~ / /		
6	候選架構分析			/ / ~ / /		
7	候選架構設計			/ / ~ / /		
8	概念驗證實作			/ / ~ / /		
9	概念驗證測試			/ / ~ / /		
10	進行成本分析			/ / ~ / /		
11	分析專案預期效益			/ / ~ / /		
12	撰寫投標建議書			/ / ~ / /		
13	進行投標			/ / ~ / /		
14	建議書簡報			/ / ~ / /		

(2)階段範疇認知不當問題：藉由制定「獲得系統需求」工作運用之「範疇評估管理表單」(請參閱表 3)，管理各範疇之掌握度，以利決定適切之評估深度，其應用方法、時機及表單樣本如下：

A.應用方法：於「獲得系統需求」工作管理各範疇之掌握度，以利決定適切之評估深度。

B.應用時機：

- a. 填表：於「獲得系統需求」工作開始時進行填寫。
- b. 檢視及修改：每天定時進行檢討，新增剛獲得之系統範疇評估項目。

C.表單樣本：

表 3 SPDP 起動階段-範疇評估管理表單

專案名稱：

階段工作負責人：

項次	候選需求名稱	評估者	相關經驗評估						
			相關程度		相關經驗專案名稱	確認者			
			<input type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input type="checkbox"/> <20%		
			<input type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input type="checkbox"/> <20%		
			<input type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input type="checkbox"/> <20%		
			<input type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input checked="" type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input type="checkbox"/> <20%		
			<input type="checkbox"/> >80%	<input checked="" type="checkbox"/> >60%	<input type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input type="checkbox"/> <20%		
			<input type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input checked="" type="checkbox"/> >40%	<input checked="" type="checkbox"/> >20%	<input type="checkbox"/> <20%		
			<input type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input checked="" type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input checked="" type="checkbox"/> <20%		
			<input checked="" type="checkbox"/> >80%	<input checked="" type="checkbox"/> >60%	<input checked="" type="checkbox"/> >40%	<input checked="" type="checkbox"/> >20%	<input checked="" type="checkbox"/> <20%		
			<input checked="" type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input checked="" type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input checked="" type="checkbox"/> <20%		
			<input type="checkbox"/> >80%	<input type="checkbox"/> >60%	<input type="checkbox"/> >40%	<input type="checkbox"/> >20%	<input type="checkbox"/> <20%		

(3)階段時間短暫問題：藉由制定「規劃起動階段」工作運用之「起動階段計劃檢核表」(請參閱表 2)，輔助階段中各項工作安排，並進行各項工作進度情況管理，其應用方法、時機及表單樣本如下：

A.應用方法：於「規劃起動階段」工作輔助階段中各項工作安排，並進行各項工作進度情況管理。

B.應用時機：

- a. 填表：於「規劃起動階段」工作開始時進行填寫。
- b. 檢視及修改：每天定時進行實際工作情況檢討，並依實際工作情況進行內容調整。

C.表單樣本：(請參閱表 2)

(4)階段評估專案成本客觀性問題：藉由制定「進行成本分析」工作運用之「成本分析表」(請參閱表 4)，包含軟體開發專案之常見成本項目，以避免因成本評估人員之經驗或疏失而造成專案成本低估或部份成本項目漏估，導致實際專案執行成本大幅超出預

期，甚至不符成本，其應用方法、時機及表單樣本如下：

A.應用方法：於「進行成本分析」，以進行客觀分析及掌握之專案成本。

B.應用時機：

a. 填表：於「進行成本分析」工作開始時進行填寫。

b. 檢視及修改：每天定時進行檢討，進行修正調整。

C.表單樣本：

表 4 SPDP 起動階段-成本分析表

專案名稱：		階段工作負責人：				
項目	成本類型	說明	數量	單位	單價	小計
1	管理行政	計畫主持人		人月		
		專案經理		人月		
		計畫助理		人月		
		行政雜支-印刷、交通		月		
2	系統開發	子系統1:		人月		
		子系統2:		人月		
		子系統3:		人月		
		子系統4:		人月		
3	保固維護	提供7x24保固維護		月		
4	教育訓練	教材撰寫				
		教材印刷				
		教室租借		場		
		師資人力				
		交通差勤				
5	採購	硬體採購項目及規格		套		
6		軟體採購項目及規格		套		
7	技術顧問			式		
總計					新台幣	元整(含稅)

(5)對把握度不足專案投入之階段費用損耗問題：藉由制定「招標文件取得與評估」、「可行性評估」及「分析專案預期效益」工作運用之「第一階段可行性評估報告表」(請參閱表 5)、「第二階段可行性評估報告表」(請參閱表 6)及「第三階段可行性評估報告表」(請參閱表 7)，在工作完成後即客觀進行評估並決定是否要繼續進行後續工作，以避免無謂的成本損失，其應用方法、時機及表單樣本如下：

A.應用方法：於「招標文件取得與評估」、「可行性評估」及「分析專案預期效益」，在工作完成後即分別填寫「第一階段可行性評估報告表」、「第二階段可行性評估報告表」及「第三階段可行性評估報告表」並決定是否要繼續進行後續工作，以避免無謂

的成本損失。

B.應用時機：

a. 填表：於「招標文件取得與評估」、「可行性評估」及「分析專案預期效益」工作中進行填寫「第一階段可行性評估報告表」、「第二階段可行性評估報告表」及「第三階段可行性評估報告表」。

C.表單樣本：

表 5 SPDP 起動階段-第一階段可行性評估報告表

專案名稱：_____ 階段工作負責人：_____

業務部門評估-評估者簽名：_____

項次	評估項目	評估結果	備註
1	對客戶熟悉程度 說明：_____	<input type="checkbox"/> >80%, <input type="checkbox"/> >60%, <input type="checkbox"/> >40%, <input type="checkbox"/> >20%, <input type="checkbox"/> <20%	
2	對公司策略相關性 說明：_____	<input type="checkbox"/> >80%, <input type="checkbox"/> >60%, <input type="checkbox"/> >40%, <input type="checkbox"/> >20%, <input type="checkbox"/> <20%	
3	後續商機 說明：_____	預估未來_____年，共_____萬元	
4	其它_____ 說明：_____		
是否建議繼續進行後續工作： <input type="checkbox"/> 強烈建議 <input type="checkbox"/> 建議 <input type="checkbox"/> 普通 <input type="checkbox"/> 不建議 <input type="checkbox"/> 強烈不建議			
其它說明：_____			
工程部門評估-評估者簽名：_____			
項次	評估項目	評估結果	備註
1	對專案技術把握度 說明：_____	<input type="checkbox"/> >80%, <input type="checkbox"/> >60%, <input type="checkbox"/> >40%, <input type="checkbox"/> >20%, <input type="checkbox"/> <20%	
2	對專案人力「量」的充沛程度 說明：_____	<input type="checkbox"/> >80%, <input type="checkbox"/> >60%, <input type="checkbox"/> >40%, <input type="checkbox"/> >20%, <input type="checkbox"/> <20%	
3	其它_____ 說明：_____		
是否建議繼續進行後續工作： <input type="checkbox"/> 強烈建議 <input type="checkbox"/> 建議 <input type="checkbox"/> 普通 <input type="checkbox"/> 不建議 <input type="checkbox"/> 強烈不建議			
其它說明：_____			
會議結論：決議： <input type="checkbox"/> 繼續進行, <input type="checkbox"/> 放棄 主席簽名_____			
參與人員：業務部_____ 工程部_____			
原因說明：_____			

表 6 SPDP 起動階段-第二階段可行性評估報告表

專案名稱：

階段工作負責人：

項次	評估項目	評估者簽名_____
1	確認系統範疇之了解程度	
	需求1：_____	了解程度： <input type="checkbox"/> 非常了解, <input type="checkbox"/> 很了解, <input type="checkbox"/> 普通, <input type="checkbox"/> 很模糊, <input type="checkbox"/> 非常模糊
	需求2：_____	了解程度： <input type="checkbox"/> 非常了解, <input type="checkbox"/> 很了解, <input type="checkbox"/> 普通, <input type="checkbox"/> 很模糊, <input type="checkbox"/> 非常模糊
	需求3：_____	了解程度： <input type="checkbox"/> 非常了解, <input type="checkbox"/> 很了解, <input type="checkbox"/> 普通, <input type="checkbox"/> 很模糊, <input type="checkbox"/> 非常模糊
2	階段中模糊的需求之澄清程度 (至少列出「招標文件取得」工作之模糊需求)	
	模糊需求1：_____	
	模糊需求2：_____	
	模糊需求3：_____	
3	關鍵風險是否有明確的因應方案	
	關鍵風險1說明：_____	
	可能影響說明：_____	
	因應方案說明：_____	
	關鍵風險2說明：_____	
	可能影響說明：_____	
	因應方案說明：_____	
	關鍵風險3說明：_____	
4	初期的商業效益判斷是否已經完成	
	預期商業效益說明：_____	
5	其它補充：_____	
會議結論：決議： <input type="checkbox"/> 繼續進行, <input type="checkbox"/> 放棄 主席簽名_____		
參與人員：業務部_____ 工程部_____		
原因說明：_____		

表 7 SPDP 起動階段-第三階段可行性評估報告表

專案名稱：

階段工作負責人：

業務部門預估-預估者簽名：					
項次	預估分類	預估項目	預估值	單位	備註
1	非系統開發成本	採購硬體成本		元	
2		採購軟體成本		元	
3		其它非軟體開發成本		元	
是否建議繼續進行後續工作： <input type="checkbox"/> 強烈建議 <input type="checkbox"/> 建議 <input type="checkbox"/> 普通 <input type="checkbox"/> 不建議 <input type="checkbox"/> 強烈不建議					
其它說明：					
工程部門預估-預估者簽名：					
項次	預估分類	預估項目	預估值	單位	備註
1	系統開發	合計系統開發成本		人日	
1.1		子系統1：		人日	
1.2		子系統2：		人日	
1.3		子系統3：		人日	
2	差旅成本	合計系統差旅成本		元	
2.1		需求搜集		元	
2.2		系統分析		元	
2.3		系統設計		元	
2.4		其它差旅		元	
是否建議繼續進行後續工作： <input type="checkbox"/> 強烈建議 <input type="checkbox"/> 建議 <input type="checkbox"/> 普通 <input type="checkbox"/> 不建議 <input type="checkbox"/> 強烈不建議					
其它說明：					
會議結論：決議： <input type="checkbox"/> 繼續進行, <input type="checkbox"/> 放棄 主席簽名_____ 預估利潤_____元					
參與人員：業務部_____ 工程部門_____					
原因說明：					

(6)建議書提出架構無法滿足專案效能需求問題：藉由規範於「評價起始階段」工作撰寫「效能分析報告書」，並提出相關佐證，對建議書提出架構之效能提出預估，：

A.應用方法：於「評價起始階段」中撰寫，以確保建議書提出之架構滿足專案之效能需求。

B.應用時機：

a. 撰寫：於「評價起始階段」工作開始中進行填寫「效能分析報告書」。

b. 審查：於「效能分析報告書」完成後，由工程部門主管進行審查確認。

C.表單樣本：無

(7)階段參與人員之工作調配問題：藉由制定「規劃起動階段」工作運用之「起動階段計劃檢核表」(請參閱表 2)，明定參與人員負責之工作及工作區間，以確保該人員於工作進行期間之責任歸屬，其應用方法、時機及表單樣本如下：

A.應用方法：於「規劃起動階段」工作輔助階段中各項工作安排，

並進行各項工作進度情況管理。

B.應用時機：

- a. 填表：於「規劃起動階段」工作開始時進行填寫。
- b. 檢視及修改：每天定時進行實際工作情況檢討，並依實際工作情況進行內容調整。

C.表單樣本：(請參閱表 2)

(8)階段參與人員之間相互溝通問題：於階段期間，每日定期召開會議(如每天下午四點)，由階段參與人員對各項工作之情況及問題進行討論，以減少參與人員之間相互溝通問題，其應用方法、時機如下：

A.應用方法：於階段「規劃起動階段」至「評價起始階段」期間，階段參與人員對各項工作之情況及問題進行討論。

B.應用時機：每日定期召開

C.表單樣本：無。

(9)專案風險無法有效掌握及回應問題：藉由制定於階段「規劃起動階段」至「評價起始階段」期間工作運用之「風險管理表」(請參閱表 8)，以確保主要風險皆已獲得掌握及妥善因應，其應用方法、時機及表單樣本如下：

A.應用方法：於階段「規劃起動階段」至「評價起始階段」期間工作時進行運用，用以記錄風險及其因應方式，以確保主要風險皆已獲得掌握及妥善因應。

B.應用時機：

- a. 新增：於階段「規劃起動階段」至「評價起始階段」期間工作，於發現風險時隨時予以填寫。
- b. 檢視及修改：每天依需求進行檢討及調整。

C.表單樣本：

表 8 SPDP 起動階段-風險管理表

專案名稱：

階段工作負責人：

項次	反應者	主要的風險描述	造成影響	因應策略	可能性	衝擊性	風險評分 (可能×衝擊)
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	

(10) 建議提出之採購項目無法如期完成問題：藉由制定「進行成本分析」工作運用之「採購項目確認表」(請參閱表 9)，以確保建議書提出之套裝軟體版權及硬體可依計劃完成採購，其應用方法、時機及表單樣本如下：

A. 應用方法：於「進行成本分析」工作時填寫，以確保建議書提出之套裝軟體版權及硬體可依計劃完成採購。

B. 應用時機：

a. 填表：於「進行成本分析」工作開始時進行填寫。

b. 檢視及修改：每天定時進行檢討，進行修正調整。

C. 表單樣本：

表 9 SPDP 起動階段-採購項目確認表

專案名稱：

階段工作負責人：

項次	採購分類	採購項目	供應商	聯絡人	報價資訊			付款方式	交貨天數	報價單編號
					單價	數量	總價			
	<input type="checkbox"/> 硬體採購 <input type="checkbox"/> 軟體採購							<input type="checkbox"/> 現金 <input type="checkbox"/> 月結__天		
	<input type="checkbox"/> 硬體採購 <input type="checkbox"/> 軟體採購							<input type="checkbox"/> 現金 <input type="checkbox"/> 月結__天		
	<input type="checkbox"/> 硬體採購 <input type="checkbox"/> 軟體採購							<input type="checkbox"/> 現金 <input type="checkbox"/> 月結__天		
	<input type="checkbox"/> 硬體採購 <input type="checkbox"/> 軟體採購							<input type="checkbox"/> 現金 <input type="checkbox"/> 月結__天		
	<input type="checkbox"/> 硬體採購 <input type="checkbox"/> 軟體採購							<input type="checkbox"/> 現金 <input type="checkbox"/> 月結__天		



3.2.2. 詳述階段(Elaboration Phase)之程序工作

1. 詳述階段之目標：

此階段的主要工作包含以下四項：

- (1)獲得大多數的剩餘需求，並將功能性需求以使用個案(Use Cases)進行表示。
- (2)建立基準架構，用以規範「建構階段」及「移交階段」之工作產出。
- (3)持續監控殘餘關鍵風險，並對重大風險進行辨視、評估對商業效益的衝擊。
- (4)「專案執行計劃書」之撰寫、通過審查並正式完成交付。

2. 詳述階段之主要工作：

本階段之重點著重於獲得及精煉多數的需求、發展專案之基準架構，將「統一軟體開發流程」(USDP)所述之主要工作，配合專案管理知識體系的精神，將本階段工作調整如下：

● USDP-詳述階段早期工作

- (1)規劃詳述階段(Planning the Elaboration Phase; USDP- 14.2.1)：由於專案建議書審查過程中可能承諾一些專案附加條款，故簽約後應即刻召集工程與業務部門進行工作內容確認，並規劃本專案之各項工作內容及完成日期，以確認專案之複雜度變化，由於起動階段之評估時間短暫且因附加條款之追加，所以專案時間及預算必須在詳述階段重新檢視、調整或重新規劃，以因應實際專案情況。
- (2)組建專案團隊(Build the Team; USDP- 14.2.2)：起動階段之專案團隊一般為臨時依需求組成之暫時編組，詳述階段之專案團隊一般將持續至後續建構階段，且建構階段需要大量人力，故必須一併考量；專案編組為依專案需求調配人力時間的「功能性組織」(Functional Organization)，尤需注重編制於其它部門之專案參與人員，需要將其時間進行妥善預訂。此外，對高技術風險的議題(如：建議書提出專案團隊不熟悉之外購套裝軟體之適用性)，應即刻安排人員著手進行解決。
- (3)制定開發環境(Modify the Development Environment; USDP-

14.2.3)：階段之工作重點為確認採購專案所需軟硬體並建立專案之開發環境，並對專案成員進行技能確認，必要時應施以教育訓練，以確保完成專案之人員能力及環境之充份準備。

(4)設定評估條件(Setting the Evaluation Criteria; USDP- 14.2.4)：因應階段目標，詳細列出下列項目之詳細內容：

- A.需求是否已經展開(Extend the Requirements)。
- B.基準架構是否已完成制定(Establish Baseline Architecture)。
- C.重大風險是否已有明確的因應方案(Mitigate the Significant Risks)。
- D.專案效益評估是否已經完成(Judge the Worth of the Initial Business Case)。

- USDP-執行需求訪談至測試之核心工作：詳述階段之核心工作為獲得完整之系統需求，並對剩餘技術風險進行分析、設計、實作與測試，以完成風險之澄清，此外，必須對主要功能(核心功能)進行分析及設計，以確認系統架構並建立架構基準線。

(5)獲得完整系統需求(Capture the Requirements; USDP- 14.4.1)：本工作之重點為找尋、排列、詳述及建構使用個案，其主要有以下五項行動：

- A.構建 100%之業務模型(Business Model)。
- B.依使用個案之重要性及優先性找尋及辨視 80%以上之使用個案及 100%主動者(Find Use Cases and Actors)。
- C.以技術風險高者及功能優先性進行使用個案之後續工作順序排列(Prioritize Use Cases)。
- D.詳述 40%~80%重要或把握程度較低的使用個案(Detail a Use Case)。
- E.視專案需求製作使用者介面雛型(Prototype User-Interface)。

(6)架構基準線分析(Analysis; USDP- 14.4.2)：在起動階段已經建立了一個初略的分析模型，在本階段的分析工作除將進行子系統架構分析及確認，並將技術風險高或與架構有密切關係之子系統進行細部分析，以進行分析模型之精煉，本工作包含以下四項行動：

- A.分析架構-確認子系統架構(Architectural Analysis)。
- B.分析 20%~40%技術風險高或與架構有密切關係的包裝(Analyze a Package)。
- C.分析 20%~40%技術風險高或與架構有密切關係的使用個案(Analyze a Use Case)。

D.分析 20%~40% 技術風險高或與架構有密切關係的類別 (Analyze a Class)。

(7)架構基準線設計(Design; USDP- 14.4.3)：本階段通常僅設計及實作少於 10%的重要使用個案，這些被設計的使用個案、類別與子系統僅限於技術風險高或與架構有重大相關者，以完成系統架構基準線，本工作包含以下四項行動：

A.設計架構-確認架構設計(Architectural Design)，包含定義系統軟體架構、中繼軟體(Middleware)及資料庫類型。

B.設計小於 10%技術風險高或與架構有密切關係的使用個案 (Design a Use Case)。

C.設計小於 10%技術風險高或與架構有密切關係的類別 (Design a Class)。

D.設計小於 10%技術風險高或與架構有密切關係的子系統(Design a Subsystem)。

(8)少數功能實作(Implementation; USDP- 14.4.4)：本階段僅實作極少數技術風險高之功能，通常實作的數量將少於 10%的重要使用個案，本工作包含以下四項行動：

A.架構性建置(Architectural Implementation)。

B.實作小於 10%技術風險最高的類別與子系統(Implement a Class and Implement a Subsystem)。

C.不斷將各反覆之產出整合為系統(Integrate System)。

(9)少數功能測試(Test; USDP- 14.4.5)：本工作聚焦於確認基準架構各子系統與其相互介面間作業之可行性，包含以下四項行動：

A.進行測試計劃之規劃(Plan Test)。

B.設計專案測試之作法(Design Test)。

C.對上述實作之系統執行整合測試(Perform Integration Test)。

D.執行系統效能測試(Perform System Performance Test)。

● USDP-樹立商業效益

(10) 調整成本預估基準(Prepare the Business Bid; USDP- 14.5.1)：起動階段預估之採購軟硬體項目及軟體開發成本是根據軟體規模及過往開發經驗作為成本預估基準進行估計，在前述工作進行技術風險澄清後將可確定專案採購之軟硬體，本工作將實際進行軟硬體採購之議價，以取得精確之採購成本，此外由於後續「建構階段」將大幅進行軟體開發，在本階段之主要目標為建構架構基

準線，為了使專案之成本以最節約方式進行，故必須依專案團隊架構基準線之實際執行情況進行成本預估基準調整，以建立最節約之成本規劃及預估。

- (11) 更新投資報酬率(Update Return on Investment; USDP-14.5.2)：根據上述調整後之成本預估基準，進行專案預期獲利計算，以建立接近真實情況之獲利預估，考量本程序定位，本工作名稱應調整為「更新專案預期效益」。

● USDP-評價詳述階段工作

- (12) 評價詳述階段(Assess the Iterations in the Elaboration Phase; USDP- 14.6)：本工作之重點為評價此階段提出架構基準線之適切性，一般以召開會議對內部提出說明，並在專案團隊內部確認專案之主要技術風險皆已獲得排除。

● USDP-規劃建構階段工作

- (13) 規劃建構階段(Planning the Construction Phase; USDP- 14.7)：依據制定之基準架構進行系統之分析、設計、建置及測試等開發工作之安排，因此本工作之重點置於上述開發工作之安排。

● SPDP-因應本程序為軟體公司承接業主之軟體標案進行系統開發而須完成之工作

- (14) 專案簽約：於專案議價及議約工作完成後，專案之價格及內容即完成確認，在業主與公司完成用印後，專案即正式展開。

- (15) 撰寫專案執行計劃書：一般在專案正式展開後的第一個交付項目為「專案執行計劃書」，「專案執行計劃書」將進行整個專案執行過程的規劃，通常包含「專案組織」、「專案工作分解結構」、「專案風險及回應說明」...等。

- (16) 專案啟動會議：一般於「專案執行計劃書」完成後會邀集主要之專案利害關係人進行「專案啟動會議」，進行「專案執行計劃」摘要、架構與需求確認、專案風險及回應，及需請業主配合之工作項目說明(如：不包含網路線路佈建之專案，須請業主配合完成網路線路佈建之時程)，並正式授權專案委由專案團隊進行專案執行任務。

- (17) 交付項目修訂：將「專案執行計劃書」交付業主後，一般業主

將於一定期間回覆審查意見，並由專案團隊進行交付項目之修訂，以符合業主需求而獲接受。

(18) 進行階段驗收：在業主正式接受專案團隊交付之「專案執行計劃書」後，一般會以驗收會議方式進行正式檢收，如為付款階段則於正式驗收後依合約所訂條件進行階段請款程序。

3. 詳述階段之工作流程：

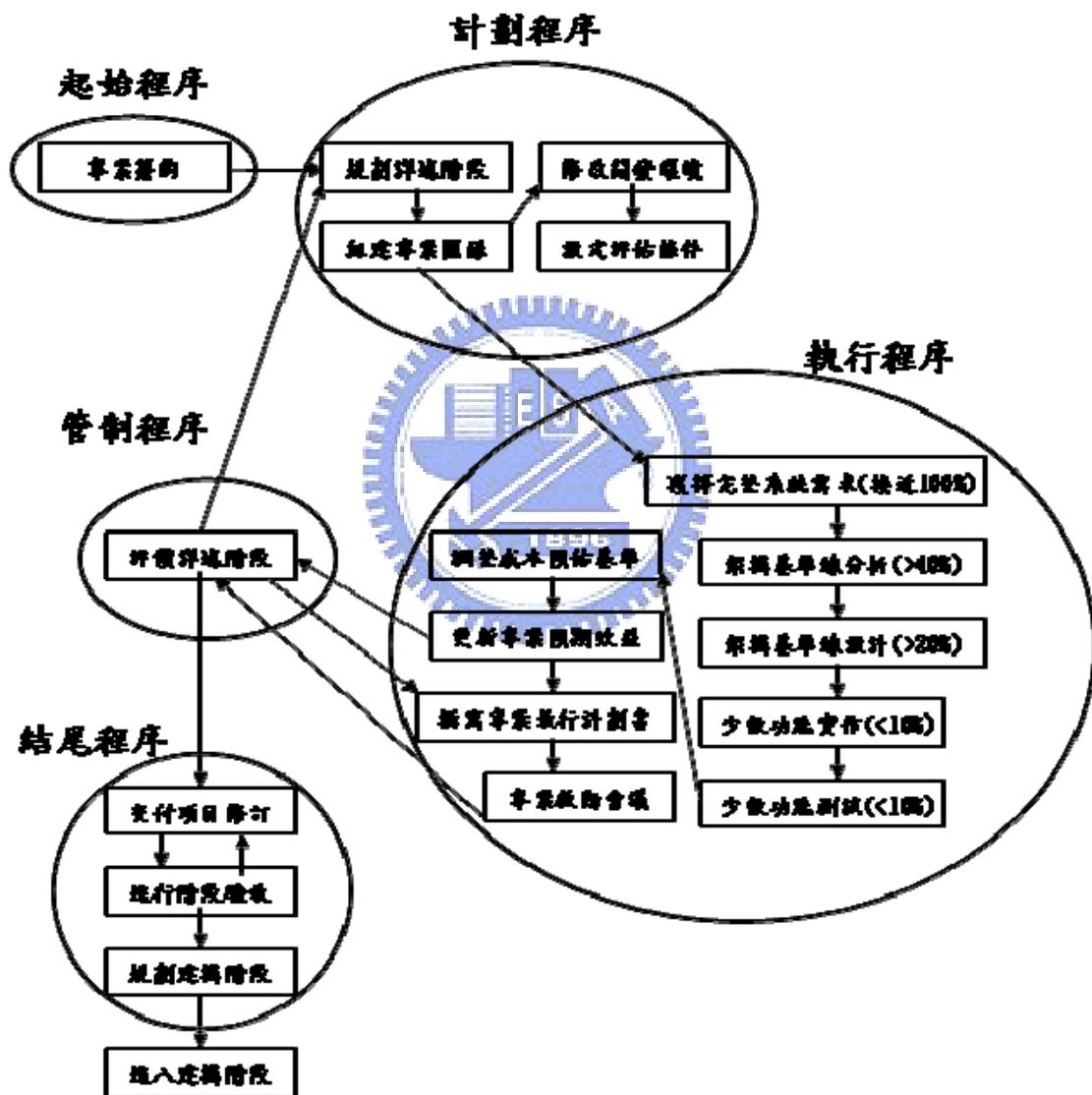


圖 10 SPDP 詳述階段之工作流程

4. 詳述階段之主要工作產出：

詳述階段之主要交付項目(Artifact)包含「專案執行計劃書」將於詳述

階段之各項工作中陸續產出，各項工作之產出如下：

- (1) 規劃詳述階段：詳述階段之工作計劃。
- (2) 組建專案團隊：專案後續階段(包含詳述、建構及移交階段)參與人員名單，包含參與時機及所需技能。
- (3) 制定開發環境：專案開發環境之說明及應舉辦或接受技能訓練課程及人員。
- (4) 設定評估條件：階段評估條件。
- (5) 獲得完整系統需求：完整的「業務模型」及大部「使用個案模型」文件內容。
- (6) 架構基準線分析：系統的「分析模型」包裝圖(Package Diagram)及架構關鍵性「類別」(Class)。
- (7) 架構基準線設計：系統的軟、硬體架構及系統的「設計模型」包裝圖(Package Diagram)。
- (8) 少數功能實作：高技術風險功能的實作程式。
- (9) 少數功能測試：高技術風險功能的測試文件。
- (10) 調整成本預估基準：「成本預算表」。
- (11) 更新專案預期效益：「專案預估獲利表」。
- (12) 評價詳述階段：「詳述階段評價會議」，高技術風險項目因應情況及專案團隊內部確認之架構基準線。
- (13) 規劃建構階段：建構階段人力及時程規劃。
- (14) 專案簽約：簽訂之「專案合約」。
- (15) 撰寫專案執行計劃書：「專案執行計劃書」。
- (16) 專案啟動會議：啟動會議簡報及會議記錄。

(17) 交付項目修訂：交付項目的正式版本。

(18) 進行階段驗收：「詳述階段驗收合約」。

5. 詳述階段之管理問題分析：

考量本階段限制及特性，彙整實務常見之管理問題如下：

(1)高技術風險掌握及回應問題:部份技術風險在起動階段尚無法完全澄清，如本階段仍未完成辨識及客觀評價、尚未提出適當之因應計劃並有效管理，可能導致難以結案。

(2)專案規劃採購項目風險：專案規劃之採購項目，包含：套裝軟體版權及硬體...等，可能因採購不及或交貨期間因素，造成擔誤後續工作之進行。

(3)專案成員之技能訓練問題:因專案成員之組成人員可能不完全具備擔任安排專案角色所需技能，因而造成專案之產品品質不佳或開發進度不如預期。

(4)客戶負責工作延遲影響專案團隊工作問題:專案部份工作須由客戶與專案團隊合作進行(如：由客戶準備網路佈線)，此工作如未及時完成將影響專案工作時程之如期執行(如：將系統主機部署至客戶端)。

(5)評估後續成本精準度問題：起動階段因評估時間短暫，故開發人力成本之預估客觀性不佳，如果仍依起動階段之採購成本及人力成本預估，導致實際專案執行無法以最節約之成本進行。

(6)人員工作先後順序安排問題:本後續階段牽涉客戶或跨單位之配合事項與大量開發工作安排，如：在規模較大之專案，通常需要交由多組人同時進行設計及實作，再由另一組人進行測試，如果工作無法妥善安排，通常將會發生人員問題，此外，開發過程中有延遲之情形在所難免，如無及時有效因應，牽一髮而動全身，影響後續工作時程。

(7)工作重要程度不清晰：本階段牽涉大量工作之執行，然而各項工

作之重要性有別，如：特定子系統之開發工作延遲可能導致系統架構之不確定，而影響整體專案時程延遲。

6. 詳述階段管理問題之解決方案及管理表單樣本：

上述問題可以「專案管理知識體系」之「九大知識領域」進行歸納如下：

- (1)高技術風險掌握及回應問題：風險管理(Risk Management)問題。
- (2)專案規劃採購項目風險：採購管理(Procurement Management)問題。
- (3)專案成員之技能訓練問題：人力資源管理(Human Resource Management)問題。
- (4)客戶負責工作延遲影響專案團隊工作問題：時間管理(Time Management)問題。
- (5)評估後續成本精準度問題：成本管理(Cost Management)問題。
- (6)人員工作先後順序安排問題：整合管理(Integration Management)問題。
- (7)工作重要程度不清晰：成本管理(Cost Management)問題。

上述各項問題可以參考「專案管理知識體系」之「九大知識領域」，在階段各項工作提出解決方案及管理表單如下：

- (1)高技術風險掌握及回應問題：藉由制定本階段工作運用之「風險管理表」(請參閱表 10)，以確保主要風險皆已獲得掌握及妥善因應，其應用方法、時機及表單樣本如下：
 - A.應用方法：於本階段期間工作時進行運用，用以記錄風險及其因應方式，以確保主要風險皆已獲得掌握及妥善因應。
 - B.應用時機：
 - a. 新增：於本階段期間工作，於發現風險時隨時予以填寫。

b. 檢視及修改：每天依需求進行檢討及調整。

C. 表單樣本：

表 10 SPDP 詳述階段-風險管理表

專案名稱： _____ 階段工作負責人： _____

項次	反應者	主要的風險描述	造成影響	因應策略	可能性	衝擊性	風險評分 (可能X衝擊)
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	
					<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	<input type="checkbox"/> 非常高(0.9) <input type="checkbox"/> 高(0.7) <input type="checkbox"/> 中等(0.5) <input type="checkbox"/> 低(0.3) <input type="checkbox"/> 非常低(0.1)	

(2) 專案規劃採購項目風險：藉由制定「規劃詳述階段」工作運用之「採購進度管制表」(請參閱表 11)，以確保建議書提出之套裝軟體版權及硬體可以如期完成採購，其應用方法、時機及表單樣本如下：

A. 應用方法：於「規劃詳述階段」進行規劃，以確定建議書提出之套裝軟體版權及硬體可以如期完成採購。

B. 應用時機：

a. 填表：於「規劃詳述階段」時進行規劃，並於議價後追蹤預計訂單日期及預計交貨日期。

b. 檢視及更新：於每次「定期工作會議」進行確認，並更新最新情況。

C. 表單樣本：

表 11 SPDP 詳述階段-採購進度管制表

專案名稱：

階段工作負責人：

項次	採購項目	需求使用日期	供應商	聯絡人	訂貨至交貨時間預估	預計請購日期	預計訂單日期	預計交貨日期

(3)專案成員之技能訓練問題：藉由制定「組建專案團隊」工作運用之「專案成員技能評估表」(請參閱表 12)，以進行專案成員之技能評估，並了解專案成員擔任專案角色之技能差異，需要安排教育訓練之情況，則以「專案成員技能訓練管制表」(請參閱表 13)進行管理，以確保專案成員技能訓練確實完成：

A.應用方法：於「組建專案團隊」工作確認專案成員後進行技能評估，了解專案成員擔任專案角色之技能差異，另需安排教育訓練，則進行技能訓練課程安排及管制。

B.應用時機：

- a. 填表：於「組建專案團隊」工作確認專案成員後即進行填寫「專案成員技能評估表」及「專案成員技能訓練管制表」。
- b. 檢核及更新：於每週專案會議進行「專案成員技能訓練管制表」檢核，直到技能訓練皆已完成。

C.表單樣本：

表 12 SPDP 詳述階段-專案成員技能評估表

專案名稱： 階段工作負責人：

項次	專案成員	擔任專案角色	技能評估				加強方案		
			領域知識	技能1	技能2	相關經驗	領域知識	技能1	技能2
			<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足		<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它
			<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足		<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它
			<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足		<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它
			<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足	<input type="checkbox"/> 非常充足 <input type="checkbox"/> 充足 <input type="checkbox"/> 尚可 <input type="checkbox"/> 不足 <input type="checkbox"/> 嚴重不足		<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它	<input type="checkbox"/> 參加外部課程 <input type="checkbox"/> 舉辦內部訓練 <input type="checkbox"/> 自行學習 <input type="checkbox"/> 工作中學習 <input type="checkbox"/> 其它

表 13 SPDP 詳述階段-專案成員技能訓練管制表

專案名稱： 階段工作負責人：

項次	專案成員姓名	加強技能項目	加強技能作法	技能訓練舉辦單位	技能訓練課程名稱	講師	開課時間	需求經費	實際訓練完成日期	完成訓練確認人員
			<input type="checkbox"/> 內訓 <input type="checkbox"/> 外訓							
			<input type="checkbox"/> 內訓 <input type="checkbox"/> 外訓							
			<input type="checkbox"/> 內訓 <input type="checkbox"/> 外訓							
			<input type="checkbox"/> 內訓 <input type="checkbox"/> 外訓							
			<input type="checkbox"/> 內訓 <input type="checkbox"/> 外訓							

(4) 客戶負責工作延遲影響專案團隊工作問題：藉由制定「撰寫專案執行計劃書」及「專案啟動會議」工作運用之「客戶配合工作管制表」(請參閱表 14)，以正式告知客戶配合工作及時程，並進行後續管制：

A. 應用方法：於展開工作分解結構後，將須請客戶配合工作列入管制，並於專案啟動會議提出說明。

B. 應用時機：

a. 填表：於「撰寫專案執行計劃書」工作展開工作分解結構後，將請客戶配合工作填入本表。

b. 說明及檢視：於「專案啟動會議」中提出說明，並由專案經理列入管制。

C. 表單樣本：

表 14 SPDP 詳述階段-客戶配合工作管制表

專案名稱： _____ 階段工作負責人： _____

項次	工作分解結構編號	工作項目	預訂工作開始時間	預訂工作結束時間	需請客戶配合事項說明	需請客戶配合人員	備註

(5)評估後續成本精準度問題：藉由制定「調整成本預估基準」工作運用之「成本預估基準表」(請參閱表 15)，於建置架構基準線時，一併調整專案之成本預估基準，並記錄專案採購項目之議價後成本：

A.應用方法：於「調整成本預估基準」工作時，依建置架構基準線之實際情況，調整成本預估基準表，以進行後續更精準預估開發成本。

B.應用時機：

a. 填表：於「調整成本預估基準」工作時進行本表填寫。

b. 檢視及更新：於下個反覆完成後進行「成本預估基準」檢視、填寫或調整。

C.表單樣本：

表 15 SPDP 詳述階段-成本預估基準表

專案名稱： _____ 階段工作負責人： _____

成本預估基準線								
項次	基準線使用個案規模	基準線使用個案	分析人日/使用個案	設計人日/使用個案	實作人日/使用個案	測試人日/使用個案	基準線確認者簽名	備註
	大							
	中							
	小							
採購成本								
項次	採購項目名稱	採購項目規格	數量	詢價價格	議價價格	議價者	備註	

(6)人員工作先後順序安排問題：在「撰寫專案執行計劃書」工作運用甘特圖(Gantt Chart)進行製作「工作分解結構」(請參閱圖 11)，並於安排時標注各工作間之前後關係，於工作時程異動時，即可明確知道被影響之後續工作：

A.應用方法：甘特圖(Gantt Chart)及工作間之前後關係可使用工具- Microsoft Project 進行管理，便於管理階段工作。

B.應用時機：

- 安排：於「撰寫專案執行計劃書」運用甘特圖(Gantt Chart)進行工作規劃。
- 檢視及更新：於每次「專案會議」工作檢討時進行確認，進行更新。

C.表單樣本：(以一組系統分析者、三組系統設計、實作者、一組測試者為例)

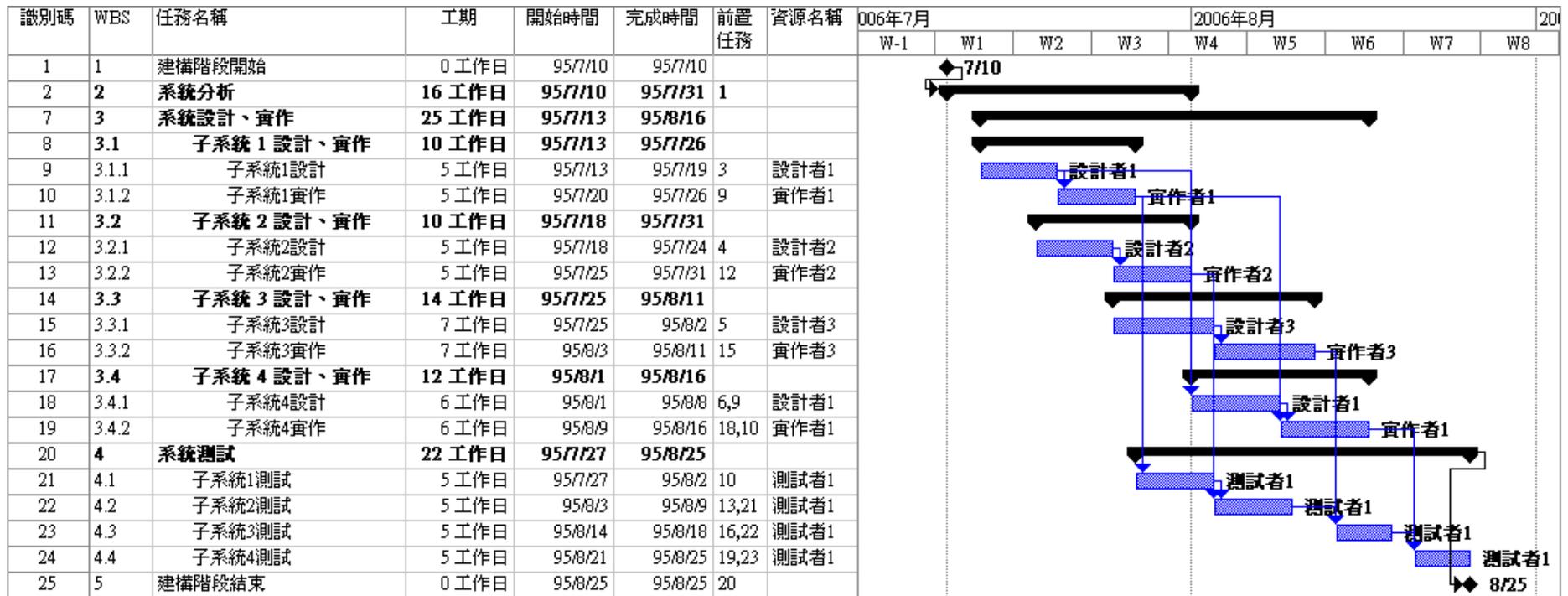


圖 11 SPDP 建構階段-甘特圖(工作分解結構)

(7)工作重要程度不清晰：在「撰寫專案執行計劃書」工作運用甘特圖(Gantt Chart)進行「工作分解結構」(WBS)安排，並轉換成「要徑圖」(Critical Path;請參閱圖 12)，要徑上之工作如果延遲，則可以能影響整體專案時程：

A.應用方法：將甘特圖轉換為「要徑圖」，可使用工具- Microsoft Project 轉換，簡少管理負擔。

B.應用時機：

- a. 安排：於「工作分解結構」完成後將甘特圖轉換為「要徑圖」，顯示專案最重要之工作。
- b. 檢視及更新：於每次「專案會議」工作檢討時進行確認，進行更新。

C.表單樣本：

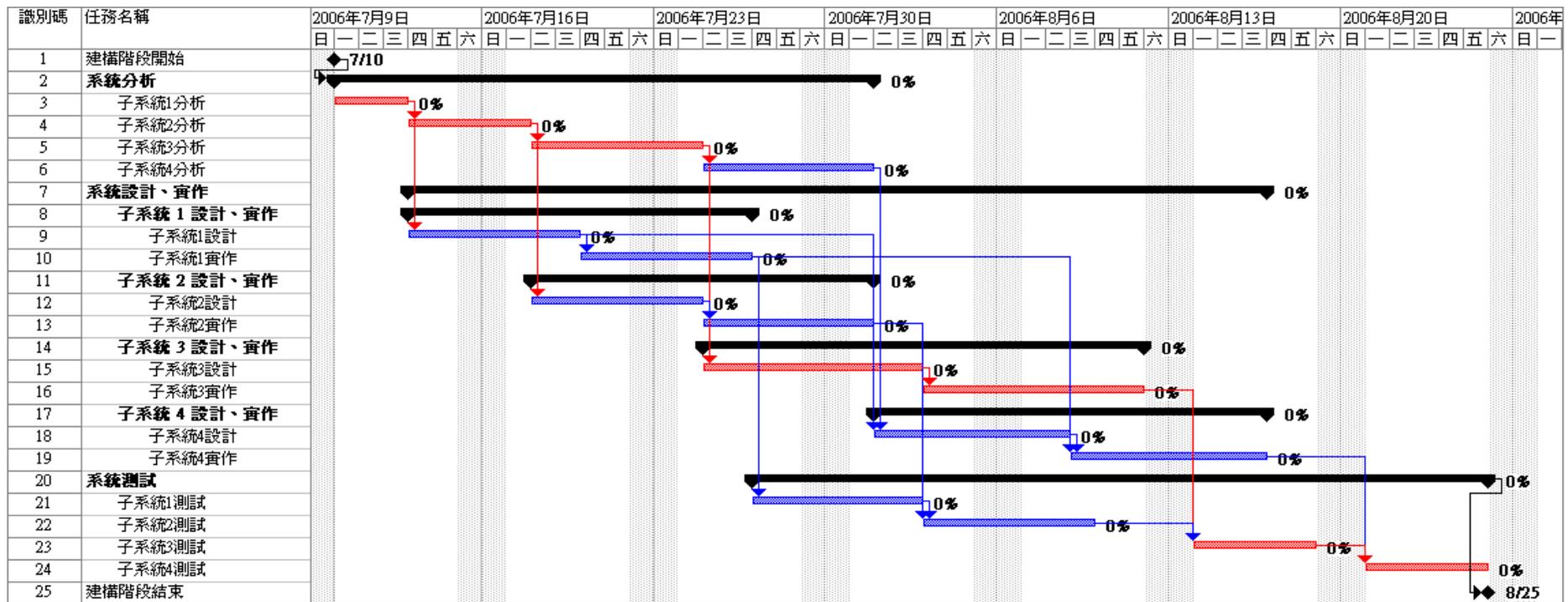


圖 12 SPDP 建構階段-要徑圖

3.2.3. 建構階段(Construction Phase)之程序工作

1. 建構階段之目標：

此階段的重點放在分析、設計及程式開發的實作部份，並繼續發展上一階段的基準架構，不斷附加、更新系統功能，使系統達成可以上線運作的目標。

2. 建構階段之主要工作：

依「統一軟體開發流程」(USDP)所述之主要工作，配合專案管理知識體系的精神，定義本階段工作如下：

● USDP-建構階段早期工作

(1)分派人員工作(Staffing the Phase; USDP- 15.2.1)：詳述階段已完成架構基準線，並將子系統及介面進行切割，本工作之重點在於將子系統分配給明確之開發成員- 「使用個案工程師」(Use-case Engineer)、「元件工程師」(Component Engineer)、「測試工程師」(Test Engineer)、「系統整合工程師」(System Integrator)、「整合測試工程師」(Integration Tester)及「系統測試工程師」(System Tester)。

(2)設定評估條件(Setting the Evaluation Criteria; USDP- 15.2.2)：因應階段目標，我們通常對以下項目進行評估：

A.專案產出系統之完整性及成熟度是否足以移交給客戶。

B.階段之里程碑，一般包含各反覆(Iteration)規劃完成時間，及合約規範交付文件(User Material)，本階段交付文件包含：

- 系統分析規格書。
- 系統設計規格書。
- 系統測試計劃書。
- 系統測試報告書。
- 原始程式及程式執行檔
- 系統使用手冊。
- 保固維護計劃
- 系統安裝手冊

C.專案產能。

● USDP-需求了解至測試之核心工作：建構階段之核心工作為不

斷反覆進行需求了解至測試之工作，並整合各次反覆之成果進行建構可執行之系統。

(3)了解完整系統需求(Requirements; USDP- 15.4.1)：詳述階段已對近 80%的重點使用個案進行了解，本工作之重點為對詳述階段尚未完全了解之使用個案進行了解、詳述及建構，並完成使用者介面雛型，本工作包含以下五項行動：

- A.完成找尋及辨視所有剩餘的使用個案及主動者(Find Use Cases and Actors)。
- B.進行未完成使用個案之開發順序排列(Prioritize Use Cases)。
- C.詳述所有剩餘的使用個案(Detail a Use Case)。
- D.建構完整的使用個案模型(Structure the Use-Case Model)。

(4)完整功能分析(Analysis; USDP- 15.4.2)：在詳述階段已經分析了約 40%的使用個案，本工作之重點為對完成詳述階段尚未完成之使用個案分析，並更新架構圖，本工作包含以下四項行動：

- A.分析架構-建立完整之包裝架構圖(Architectural Analysis)。
- B.分析所有剩餘的包裝(Analyze a Package)。
- C.分析所有剩餘的使用個案(Analyze a Use Case)。
- D.分析所有剩餘的類別 (Analyze a Class)。

(5)完整功能設計(Design; USDP- 15.4.3)：在詳述階段已經設計了約 10%的重點使用個案與架構基準線，本工作之重點為對完成詳述階段尚未完成之使用個案設計，並以不斷增添子系統之內容完成系統設計，本工作包含以下四項行動：

- A.設計架構-建構完整之架構(Architectural Design)。
- B.設計所有剩餘的使用個案(Design a Use Case)。
- C.設計所有剩餘的類別 (Design a Class)。
- D.設計所有剩餘的子系統(Design a Subsystem)。

(6)完整功能實作(Implementation; USDP- 15.4.4)：在詳述階段已經實作了約 10%的重點使用個案，本工作之重點為對完成詳述階段尚未完成之使用個案實作，並以不斷反覆(Iterative)增添子系統之方式完成系統實作、單元測試及系統整合，完成實作完整系統，本工作包含以下四項行動：

- A.架構性建置(Architectural Implementation)。
- B.實作所有剩餘的類別與子系統(Implement a Class and Implement a Subsystem)。
- C.進行單元測試(Perform Unit Testing)。

D.不斷將各反覆之產出整合為系統(Integrate System)。

(7)完整功能測試(Test; USDP- 15.4.5)：在詳述階段已經開始規劃測試方法及進行少量測試，本工作之重點為以不斷反覆(Iterative)增添子系統之進度完成系統之測試工作，完成完整系統之測試，本工作包含以下五項行動：

A.進行測試計劃之規劃(Plan Test)。

B.設計系統測試之作法(Design Test)。

C.對上述實作之系統執行整合測試(Perform Integration Test)。

D.執行系統測試(Perform System Test)。

E.評估測試結果(Evaluation Test)

● USDP-管理商業效益

(8)管理商業效益(Controlling the Business Case; USDP- 15.5)：在詳述階段已經藉由建構架構基準線的過程建立了成本基準線，本工作之重點為由專案經理藉由監控規劃之時程、人力、成本與實際情況之差異，盡力達成或超越原有規劃，以達成管理商業效益之目標。

● USDP-評價建構階段工作

(9)評價建構階段之反覆(Assess the Iterations in the Construction Phase; USDP- 15.6)：本工作之重點為由專案經理及其組成之評價小組，確認里程碑(反覆及合約交付項目)之達成性及品質程度，並對測試結果進行檢討，以確保專案產出系統之完整性及成熟度足以移交給客戶。

● SPDP-因應本程序為軟體公司承接業主之軟體標案進行系統開發而須完成之工作

(10) 建構階段起動會議：由於本階段將大量進行系統之分析、設計、建置及測試等開發工作，因此大部份之開發人員將在此一階段才開始參與專案進行，本會議之目的在於對專案團隊階段參與成員進行專案相關背景情況、目前進度、後續規劃及專案團隊統一規範說明。

(11) 撰寫系統分析規格書：一般在專案正式展開後的第二個交付項目為「系統分析規格書」，通常可由系統分析人員於分析後產出，「系統分析規格書」之重點章節包含「使用個案列表及說明」、「子系統及介面說明」、「系統架構圖」...等。

- (12) 撰寫系統設計規格書：一般在專案正式展開後的第三個交付項目為「系統設計規格書」，通常可由系統設計人員於設計後產出，「系統設計規格書」之重點章節包含「使用個案圖及描述」、「系統流程循序圖及說明」、「系統行動圖」、「類別圖」、「元件圖」...等。
- (13) 撰寫系統測試計劃書：一般在專案正式展開後的第四個交付項目為「系統測試計劃書」，通常可由測試人員於測試前預先進行規劃及撰寫，「系統測試計劃書」之重點章節包含「測試範圍及執行時間」、「測試工具及方法說明」、「測試項目的通過/失敗準則」、「測試個案」...等。
- (14) 撰寫系統測試報告書：一般在專案正式展開後的第五個交付項目為「系統測試報告書」，通常可由測試人員於測試後進行撰寫，「系統測試報告書」之重點章節包含「測試個案」、「測試結果」、「測試問題與處理報告單」、「效能測試結果」...等。
- (15) 撰寫系統使用手冊：一般在專案正式展開後的第六個交付項目為「系統使用手冊」，通常可指定專人進行撰寫，「系統使用手冊」通常包含「系統簡介」、「系統流程」、「系統角色與對應功能」、「功能操作畫面及說明」...等。
- (16) 撰寫保固維護計劃：一般在專案正式展開後的第七個交付項目為「保固維護計劃」，通常內容將評估系統之實際開發、測試情況，擬定保固維護期間之維護方案，「保固維護計劃」通常包含「保固維護小組之編組」、「保固維護期間之溝通方式」...等。
- (17) 交付項目修訂：將階段交付項目交付業主後，一般業主將於一定期間回覆審查意見，並由專案團隊進行交付項目之修訂，以符合業主需求而獲接受。
- (18) 進行階段驗收：在業主正式接受專案團隊交付之「專案執行計劃書」後，一般會以驗收會議方式進行正式檢收，如為付款階段則於正式驗收後依合約所訂條件進行階段請款程序。

3. 建構階段之工作流程：

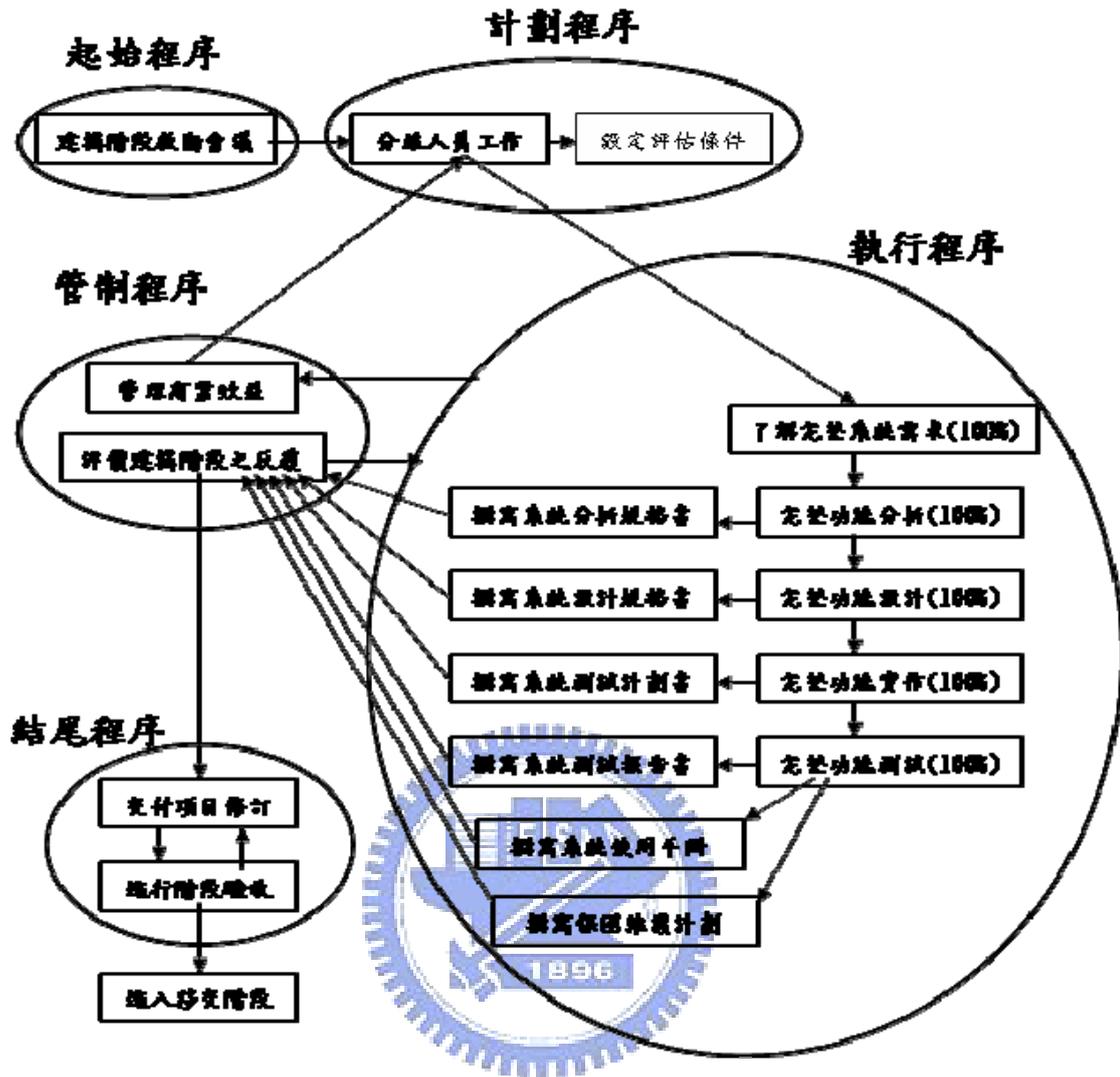


圖 13 SPDP 建構階段之工作流程

4. 建構階段之主要工作產出：

- (1)分派人員工作：專案分工表。
- (2)設定評估條件：里程碑(含反覆及合約交付項目)之評估條件及足以移交給客戶之專案產出系統的完整性及成熟度定義。
- (3)了解完整系統需求：專案團隊內部作為需求追溯使用之需求文件。
- (4)完整功能分析：專案團隊內部維護使用之分析文件。
- (5)完整功能設計：專案團隊內部維護使用之設計文件。

- (6)完整功能實作：「程式碼」、「執行檔」及「完整系統」。
- (7)完整功能測試：「測試範圍及執行時間」、「測試工具及方法說明」、「測試項目的通過/失敗準則」、「測試個案」、「測試結果」、「測試問題與處理報告單」、「效能測試結果」。
- (8)管理商業效益：工作計劃與實際進度與工時落差，程式行數。
- (9)評價建構階段之反覆：會議記錄，包含檢討會議之問題列表。
- (10) 建構階段起動會議：建構階段起動會議簡報及會議記錄。
- (11) 撰寫系統分析規格書：正式交付之「系統分析文件」。
- (12) 撰寫系統設計規格書：正式交付之「系統設計文件」。
- (13) 撰寫系統測試計劃書：正式交付之「系統測試文件」。
- (14) 撰寫系統測試報告書：正式交付之「系統測試文件」。
- (15) 撰寫系統使用手冊：「系統使用手冊」。
- (16) 撰寫保固維護計劃：「保固維護計劃」。
- (17) 交付項目修訂：交付項目的正式版本。
- (18) 進行階段驗收：「階段驗收合約」。

5. 建構階段之管理問題分析：

考量本階段工作特性，彙整實務常見之管理問題如下：

- (1)開發進度管理問題：本階段之主要工作是由大量開發人員密集地從事系統開發工作，然而因為軟體開發的不可視特性、人員之不可預期突發情況及除錯(Debug)過程可能因思考盲點讓開發進度長久延遲...等，造成開發進度落後。

(2)建構管理問題:本階段之主要工作是由大量開發人員密集地從事系統開發工作,在較大型之專案通常由多組開發人員進行,因此各組人員開發成果之合併及各反覆之產出版本常常造成相互干擾,而造成專案時程之延遲,此外,實務上常發生不可預期之問題,如:開發者電腦中毒、硬碟毀損...等情況,或因開發者擁有相當多開發版本而產生人為疏失(如:改錯版本),造成必須重新重覆開發已完成程式,造成成本浪費,此外,在此一階段,使用者通常在進行反覆(Iteration)開發的功能確認時會提出功能變更之需求,這些需求變更如未妥善進行管理時會造成使用者對專案的滿意度不佳。

(3)開發環境與正式環境差異問題:本階段之最重要產物為提供一版可移交予客戶之系統,然而可能因開發環境與正式環境之細微差異,造成交付客戶之系統存在部份瑕疵或少數功能無法執行。

(4)工作品質不佳的問題:因本階段開發工作或撰寫文件有瑕疵而未能及時予以改善、階段進行測試完整性不足,導致後續相關開發工作品質遭受影響或工作產出之品質不佳。

(5)階段參與人員之間相互溝通問題:階段參與成員因工作默契不足,又因本階段可能必須與其它成員開發產品頻頻進行整合,如無法妥善進行相互溝通之磨合,導致因參與成員間之溝通方式不同,而造成資訊傳遞不良。

(6)人員因突發事件而無法繼續從事專案開發問題:專案人員可能因突發之個人事件或公司之緊急支援需求而長期無法從事專案開發,後續人員接手後往往難以接續順利進行開發。

6. 建構階段管理問題之解決方案及管理表單樣本:

(1)開發進度管理問題:時間管理(Time Management)問題。

(2)建構管理問題:品質管理(Cost Management)問題。

(3)開發環境與正式環境差異問題:品質管理(Quality Management)問題。

(4)工作品質不佳的問題:品質管理(Quality Management)問題。

(5)階段參與人員之間相互溝通問題：溝通管理(Communication Management)問題。

(6)人員因突發事件而無法繼續從事專案開發問題：人力資源管理(Human Resource Management)問題。

上述各項問題可以參考「專案管理知識體系」之「九大知識領域」，在階段各項工作提出解決方案及管理表單如下：

(1)開發進度管理問題：藉由制定「分派人員工作」運用之「反覆管制表」(請參閱表 16)，於工作分派後明確記載各反覆之反覆編號、分析、設計、實作與測試的負責人及里程碑日期，並於里程碑日期到達提出進行確認，以確保開發進度掌握並及時協助排除問題：

A.應用方法：於「分派人員工作」工作，完成分派人員工作後進行里程碑登記，並進行後續管理。

B.應用時機：

a. 填表：於「分派人員工作」工作分派後進表單填寫。

b. 檢視及更新：於里程碑日期到達提出進行確認，並填寫實際完成日期及計算花費工時，以確保開發進度掌握並及時協助排除問題。

C.表單樣本：

表 16 SPDP 建構階段-反覆管制表

專案名稱： _____ 階段工作負責人： _____

項次	子系統名稱	反覆編號	分析				設計				實作				測試			
			負責人	里程碑日期	實際完成日期	花費工時												

(2)建構管理問題：藉由於「建構階段啟動會議」工作中統一規定建構管理使用之工具(如：CVS、SVN、TeamSource、Visual SourceSafe...等)及方式，並於「分派人員工作」時指定稽核人員進行抽查，記錄「建構管理稽核表」(請參閱表 17)並定時回報稽核結果，此外，於使用者提出需求變更時，運用「需求變更管制表」(請參閱表 18)進行需求管制：

A.應用方法：於「建構階段啟動會議」統一規範由建構管理主機下

載最新版本程式及上傳程式之時機，並於「分派人員工作」時指定稽核人員進行抽查，記錄並定時回報稽核結果，此外，於使用者提出需求變更時，進行「需求變更表管制表」(請參閱表 18)之「需求變更資訊」填寫，並由專案團隊進行變更評估後進行「評估變更資訊」填寫，作成變更決議後進行「需求變更結果資訊」填寫。

B. 應用時機：

- a. 文件上傳：如：每天下班前必須進程式上傳，以確保「建構管理主機」隔天取得之程式為最新版，不定時進行稽核。
- b. 文件下載：如：每天上班前必須由「建構管理主機」取得最新版程式進行修改，不定時進行稽核。
- c. 需求變更：於使用者提出需求變更時進行。

C. 表單樣本：

表 17 SPDP 建構階段-建構管理稽核表

專案名稱： _____ 階段工作負責人： _____

項次	開發者 姓名	程式上傳稽核			程式下載稽核			備註
		稽核日期	稽核項目	稽核結果	稽核日期	稽核項目	稽核結果	

表 18 SPDP 建構階段-需求變更管制表

專案名稱： _____ 階段工作負責人： _____

項次	需求提出資訊			評估變更資訊			需求變更結果資訊		
	提出時間	需求變更項目	提出者	評估影響項目	評估需求變更人日	評估者	需求變更結果	原因說明	確認者
							<input type="checkbox"/> 接受 <input type="checkbox"/> 不接受 <input type="checkbox"/> _____		
							<input type="checkbox"/> 接受 <input type="checkbox"/> 不接受 <input type="checkbox"/> _____		
							<input type="checkbox"/> 接受 <input type="checkbox"/> 不接受 <input type="checkbox"/> _____		
							<input type="checkbox"/> 接受 <input type="checkbox"/> 不接受 <input type="checkbox"/> _____		
							<input type="checkbox"/> 接受 <input type="checkbox"/> 不接受 <input type="checkbox"/> _____		

(3)開發環境與正式環境差異問題：藉由於「建構階段啟動會議」開發環境之軟體版本，如：開發工具、編譯器、應用伺服器...等，

儘量與正式環境採用相同版本，並於「分派人員工作」時指定稽核人員進行抽查，記錄「開發環境稽核表」(請參閱表 18)並定時回報稽核結果：

- A. 應用方法：於「建構階段啟動會議」統一規範開發環境之軟體版本，並於「分派人員工作」時指定稽核人員進行抽查，記錄並定時回報稽核結果。
- B. 應用時機：於開發時進行，不定時進行稽核。
- C. 表單樣本：

表 19 SPDP 建構階段-開發環境稽核表

專案名稱：

階段工作負責人：

項次	開發者姓名	規範版本			開發環境版本			備註
		稽核日期	稽核項目	稽核結果	稽核日期	稽核項目	稽核結果	

(4)工作品質不佳的問題：藉由制定專案會議工作運用之「工作改進管制表」(請參閱表 20)，於工作檢討後明確記載工作負責人員姓名、工作名稱、問題分類、問題描述、解決方案、改進期限及負責確認人員，並於下次專案會議提出進行確認，以確保發現問題已被確實改進：

- A. 應用方法：於專案會議工作檢討時由會議記錄者進行填寫，隨同會議記錄寄發，並於下次會議提出確認，並填寫改進情況並請負責確認人員簽名。
- B. 應用時機：
 - a. 填表：於專案會議工作檢討後發現問題即進行填寫。
 - b. 檢視及更新：於下次專案會議工作檢討時進行確認，並填寫完成日期及改進情況，並更新問題解決情況。
- C. 表單樣本：

表 20 SPDP 建構階段-工作改進管制表

專案名稱：

階段工作負責人：

項次	人員姓名	工作名稱	問題分類	問題描述	解決方案	改進期限	完成日期	改進情況	負責確認人員簽名

(5)階段參與人員之間相互溝通問題：藉由於「建構階段啟動會議」宣佈「溝通矩陣表」(請參閱表 21)，明定資訊散佈方式。：

A.應用方法：於「建構階段啟動會議」前進行規劃，並公佈給相關專案成員，以明定資訊散佈方式。

B.應用時機：

a. 填表：於「建構階段啟動會議」前工作開始時進行填寫。

b. 檢視及修改：依實務需求進行必要之調整。

C.表單樣本：

表 21 SPDP 建構階段-溝通矩陣表

專案名稱：

階段工作負責人：

項次	溝通者	溝通對象	溝通頻率	溝通方式	結果通知方式	結果通知人員
			<input type="checkbox"/> 定期 <input type="checkbox"/> 非定期	<input type="checkbox"/> 會議 <input type="checkbox"/> E-mail <input type="checkbox"/>	<input type="checkbox"/> E-mail <input type="checkbox"/> 口頭報告 <input type="checkbox"/>	<input type="checkbox"/> 專案經理 <input type="checkbox"/> 所有專案關係人 <input type="checkbox"/>
			<input type="checkbox"/> 定期 <input type="checkbox"/> 非定期	<input type="checkbox"/> 會議 <input type="checkbox"/> E-mail <input type="checkbox"/>	<input type="checkbox"/> E-mail <input type="checkbox"/> 口頭報告 <input type="checkbox"/>	<input type="checkbox"/> 專案經理 <input type="checkbox"/> 所有專案關係人 <input type="checkbox"/>
			<input type="checkbox"/> 定期 <input type="checkbox"/> 非定期	<input type="checkbox"/> 會議 <input type="checkbox"/> E-mail <input type="checkbox"/>	<input type="checkbox"/> E-mail <input type="checkbox"/> 口頭報告 <input type="checkbox"/>	<input type="checkbox"/> 專案經理 <input type="checkbox"/> 所有專案關係人 <input type="checkbox"/>
			<input type="checkbox"/> 定期 <input type="checkbox"/> 非定期	<input type="checkbox"/> 會議 <input type="checkbox"/> E-mail <input type="checkbox"/>	<input type="checkbox"/> E-mail <input type="checkbox"/> 口頭報告 <input type="checkbox"/>	<input type="checkbox"/> 專案經理 <input type="checkbox"/> 所有專案關係人 <input type="checkbox"/>
			<input type="checkbox"/> 定期 <input type="checkbox"/> 非定期	<input type="checkbox"/> 會議 <input type="checkbox"/> E-mail <input type="checkbox"/>	<input type="checkbox"/> E-mail <input type="checkbox"/> 口頭報告 <input type="checkbox"/>	<input type="checkbox"/> 專案經理 <input type="checkbox"/> 所有專案關係人 <input type="checkbox"/>

(6)人員因突發事件而無法繼續從事專案開發問題：藉由制定於「分派人員工作」工作運用之「專案預備人員表」(請參閱表 22)，安排專案人員之預備/替代人員：

A.應用方法：於「分派人員工作」時運用「專案預備人員表」，預

先排定專案之預備/替代人員，為達成可預備/替代之目的，該人員必須參與定期舉行之專案工作會議，以進行工作進度及情況之了解。

B.應用時機：

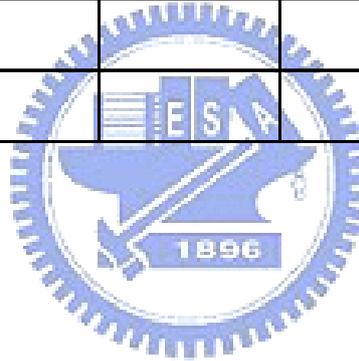
- a. 填表：於「分派人員工作」工作時進行安排。
- b. 檢視及修改：於專案工作會議依實際人力及時間情況進行預備人員調整。

C.表單樣本：

表 22 SPDP 建構階段-專案預備人員表

專案名稱： _____ 階段工作負責人： _____

項次	開發人員類型	人員姓名	負責工作期間	預備人員姓名	備註
	<input type="checkbox"/> 分析 <input type="checkbox"/> 設計 <input type="checkbox"/> 實作 <input type="checkbox"/> 測試				
	<input type="checkbox"/> 分析 <input type="checkbox"/> 設計 <input type="checkbox"/> 實作 <input type="checkbox"/> 測試				
	<input type="checkbox"/> 分析 <input type="checkbox"/> 設計 <input type="checkbox"/> 實作 <input type="checkbox"/> 測試				
	<input type="checkbox"/> 分析 <input type="checkbox"/> 設計 <input type="checkbox"/> 實作 <input type="checkbox"/> 測試				
	<input type="checkbox"/> 分析 <input type="checkbox"/> 設計 <input type="checkbox"/> 實作 <input type="checkbox"/> 測試				



3.2.4. 移交階段(Transition Phase)之程序工作

1. 移交階段之目標：

在詳述階段完成後，專案已經產出符合專案目標的初始版本系統，此階段的重點放在確保系統已達到足夠的品質水準，並符合軟體需求為目標、修正錯誤、訓練終端使用者、調整系統並加入缺少的必要功能，造出最終的軟體產品。

2. 移交階段之主要工作

依「統一軟體開發流程」(USDP)所述之主要工作，配合專案管理知識體系的精神，定義本階段工作如下：

● USDP-移交階段早期工作

(1)規劃移交階段(Planning the Transition Phase; USDP- 16.2.1)：規劃階段之重要任務之一為將建構階段產出之產品進行微調，以產出符合專案目標的初始版本系統，因此階段之工作規劃包括客戶訓練、驗收測試、維護階段等工作之安排。

(2)分派移交階段任務(Staffing the Transition Phase; USDP- 16.2.2)：移交階段重點之一為調整建構階段產出之產品，因此部份的人力分派與建構階段有很大的相關及吻合性，然而為確保專案最後階段之達成性，此一階段任務分派之考量將以開發導向(Development Oriented)轉換為服務導向(Service Oriented)，此外，必須安排進行使用者教育訓練...等工作，確保系統移交之過程順暢性。

(3)設定評估條件(Setting the Evaluation Criteria; USDP- 16.2.3)：移

交階段一般必須評估以下五項評估條件：

- A.產出之系統是否涵蓋使用者需求之關鍵功能。
- B.產出之系統是否通過合約規範之驗收測試。
- C.交付文件的品質是否可被使用者接受。
- D.教育訓練、教材及相關文件是否已備妥，隨時可加以運用。
- E.客戶及使用者是否滿意專案之產品。

● USDP-移交階段工作

- (4) 確認客戶環境(Getting Beta Release Out; USDP- 16.4.1 之一)：為使「客戶驗收測試版本」於「客戶驗收測試環境」運作無誤，故於完成客戶驗收測試版本前，必須進行客戶使用環境之確認。
- (5) 完成客戶驗收測試版本(Getting Beta Release Out; USDP- 16.4.1 之二)：開發團隊依確認後之客戶環境，完成「客戶驗收測試版本」系統並進行伺服器端(Server)系統安裝，包含本版本之「安裝步驟」。
- (6) 新舊系統資料轉換(Installing Beta Release; USDP-16.4.2 之一)：使用者進行驗收測試時可能必須比對新舊系統的結果以確認系統之正確性，故必須先進行新舊系統之資料轉換。
- (7) 客戶系統安裝及操作訓練(Installing Beta Release; USDP-16.4.2 之二)：進行教育訓練教導客戶安裝及操作「客戶驗收測試版本」系統，以確保使用者可順利安裝及操作新系統進行驗收測試，在訓練後，即可開始進行驗收測試；由於此時客戶尚未完成確認新系統的適用性，故此一階段可能採新舊系統並行方式進行作業。
- (8) 協助客戶使用系統(Installing Beta Release; USDP-16.4.2 之三)：在客戶進行驗收測試時，可能有操作或系統相關問題，此時除輔導客戶進行系統操作外，另外進行客戶反應問題之記錄，以確保客戶反應問題皆已確實掌握。
- (9) 回應客戶驗收測試結果(Responding Test Results; USDP-16.4.3)：收集並分析客戶驗收測試結果，並針對其中測試失敗項目制定回應計劃。
- (10) 使用者環境差異之因應(Adapting Product to Varied User Environment; USDP-16.4.4)：專案產出系統可能於同類業務性質之客戶皆有潛在需求，故此階段可將系統發展為多種版本(如：發展免費資料庫方案之版本)，以利專案產出系統之再利用。
- (11) 完成專案交付項目(Completing Artifacts; USDP-16.4.5)：彙總專案各階段產出之專案交付項目，完成專案合約之交付項目。
- USDP-完成商業效益評估
- (12) 管理進度(Controlling Progress; USDP- 16.5.1)：本工作之重點

為監控本階段工作排定之時程、成本與實際執行之落差，並定期進行檢討，以確保專案得以如預期進行驗收。

(13) 檢討商業效益(Review of the Business Plan; USDP- 16.5.1)：本作之重點為總結專案成本，以計算本專案之實際商業效益，並作成記錄以供後續專案執行參考之用。

● USDP-評價移交階段工作

(14) 評價移交階段之反覆(Assess the Iterations in the Transition Phase; USDP- 16.6.1)：本工作之重點為控制階段反覆在專案時程及預算內完成，並進行本階段「設定評估條件」工作所設定之工作評估。

(15) 專案回顧檢討(Postmortem of the Project; USDP- 16.6.2)：為使後續專案之執行可以更有效率及更成功，本工作之重點為檢討及記錄專案之重要資訊如下：

- A. 專案採用之設計及專案中進行評估但不獲採用設計之原因。
- B. 專案過程中可以進行檢討改進的工作，以有助於後續專案進行改進。
- C. 可再利用項目之後續發展。

● USDP-評價移交階段工作

(16) 準備下一個世代專案(Planning the Next Release or Generation; USDP- 16.7)：本階段之主要是進行專案範疇內之系統調整，使用者在本階段通常會提出一些專案範疇外之系統需求，這些需求雖然不會在本階段被滿足，但是這些需求將是下一個世代專案參考之重要資訊。

● SPDP-因應本程序為軟體公司承接業主之軟體標案進行系統開發而須完成之工作

(17) 撰寫系統安裝手冊：一般在專案正式展開後的第八個交付項目為「系統安裝手冊」，通常於系統於「完成客戶驗收測試版本」工作後，依安裝步驟進行編寫，「系統安裝手冊」通常包含「安裝步驟」、「常見安裝問題與解決建議說明」...等。

(18) 撰寫驗收報告：因應專案之驗收，本工作之重點為彙整專案之執行狀況，以證明符合專案合約之規範，並準備專案之正式驗收合約，以供後續「專案正式驗收」工作之用。

(19) 專案正式驗收：專案之正式驗收通常會由業主邀集專案利益關係人(Stakeholders)進行驗收會議，進行驗收報告之審查，並簽訂專案驗收合約。

3. 移交階段之工作流程：

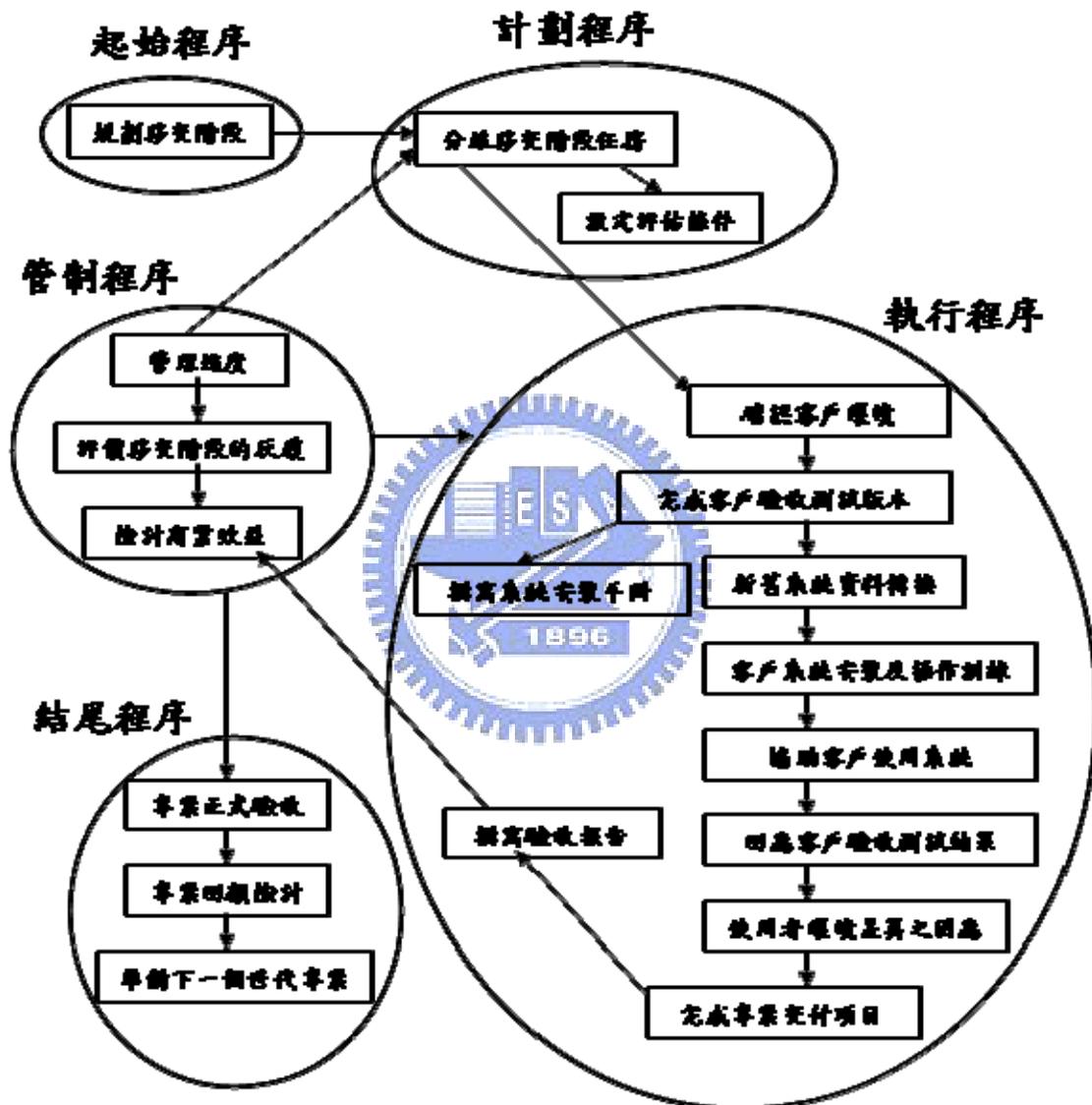


圖 14 SPDP 移交階段之工作流程

4. 移交階段之主要工作產出

- (1) 規劃移交階段：移交階段人力及時程規劃。
- (2) 分派移交階段任務：移交階段人員分工表。

- (3)設定評估條件：移交階段評估條件。
- (4)確認客戶環境：客戶環境清單。
- (5)完成客戶驗收測試版本：「客戶驗收測試版本」及伺服器系統。
- (6)新舊系統資料轉換：「資料轉換說明書」及包含資料之伺服器系統。
- (7)客戶系統安裝及操作訓練：「安裝及操作訓練」。
- (8)協助客戶使用系統：客戶反應問題記錄。
- (9)回應客戶驗收測試結果：客戶反應問題處理情況。
- (10) 使用者環境差異之因應：適用其它環境之系統版本。
- (11) 完成專案交付項目：專案交付項目。
- (12) 管理進度：工作計劃與實際進度與工時落差，程式行數。
- (13) 檢討商業效益：專案商業效益總結報告。
- (14) 評價移交階段之反覆：會議記錄，包含檢討會議之問題列表。
- (15) 專案回顧檢討：「專案回顧報告」。
- (16) 準備下一個世代專案：專案潛在需求。
- (17) 撰寫系統安裝手冊：「系統安裝手冊」。
- (18) 撰寫驗收報告：「專案驗收報告」。
- (19) 專案正式驗收：「專案驗收合約」。

5. 移交階段之管理問題分析

考量本階段工作特性，彙整實務常見之管理問題如下：

- (1)使用者驗收的問題不斷導致遲遲無法驗收問題：使用者通常在進行驗收測試時會提出一些系統之實務問題，如果這些問題未被以有效率之方式進行管理並解決，將導致使用者不斷發掘並提出新問題，使專案團隊頻頻面對新問題而疲於奔命，導致驗收遙遙無期。
- (2)最後階段才發覺交付項目缺漏而延誤驗收問題：實務上常見在最後階段才發現交付項目有缺漏或品質不佳，然而要完成這些交付項目往往需要花費一段時間，因而延誤原本預定之驗收時程。
- (3)不同使用者重覆詢問相同問題：部份顯見之系統問題往往在使用者進行驗收測試時，不停地被不同的使用者重覆提出，這些常見問題往往耗費專案團隊大量人力進行說明，而造成專案成本增加。
- (4)專案經驗無法有效累積至下一個專案問題：專案經驗如無法以有效方式進行收集，並提供公司之相關專案人員進行參考，將造成專案經驗無法有效被利用，而導致專案成本無法有效降低。

6. 移交階段管理問題之解決方案及管理表單樣本：

- (1)使用者驗收的問題不斷導致遲遲無法驗收問題：品質管理(Quality Management)問題。
- (2)最後階段才發覺交付項目缺漏而延誤驗收問題：風險管理(Risk Management)問題。
- (3)不同使用者重覆詢問相同問題：成本管理(Cost Management)問題。
- (4)專案經驗無法有效累積至下一個專案問題：成本管理(Cost Management)問題。

上述各項問題可以參考「專案管理知識體系」之「九大知識領域」，在階段各項工作提出解決方案及管理表單如下：

(1)使用者驗收的問題不斷導致遲遲無法驗收問題：藉由制定「協助客戶使用系統」與「回應客戶驗收測試結果」工作運用之「使用者提出問題管理表」(請參閱表 23)，以有效收集、因應使用者需求管理：

A.應用方法：於「協助客戶使用系統」工作進行驗收測試時進行問題收集，並於「回應客戶驗收測試結果」進行因應方案管理。

B.應用時機：

a. 填表：於「協助客戶使用系統」工作進行驗收測試時進行問題資訊收集。

b. 檢視及更新：於「回應客戶驗收測試結果」進行完成排除日期填寫。

C.表單樣本：

表 23 SPDP 移交階段-使用者提出問題管理表

專案名稱：

階段工作負責人：

問題編號	系統分類	問題描述	反應方式	接獲問題者	提出問題人員資訊				處理情形描述	預計完成日期	完成排除日期
					提出日期	提出單位	提出者	提出者聯絡電話			
	<input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 操作問題 <input type="checkbox"/>				//					//	//
	<input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 操作問題 <input type="checkbox"/>				//					//	//
	<input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 操作問題 <input type="checkbox"/>				//					//	//
	<input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 操作問題 <input type="checkbox"/>				//					//	//
	<input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 操作問題 <input type="checkbox"/>				//					//	//

(2)最後階段才發覺交付項目缺漏而延誤驗收問題：藉由制定「規劃移交階段」及「管理進度」運用之「交付項目管制表」(請參閱表 24)，以確保交付項目在驗收前已被確實完成：

A.應用方法：於「規劃移交階段」工作明確規劃交付項目之名稱、負責人及預計完成日期，並於「管理進度」工作進行檢討，並於完成後填寫實際完成日期。

B.應用時機：

a. 填表：於「規劃移交階段」工作進行交付項目之名稱、交付方式、負責人及預計完成日期填寫。

b. 檢視及更新：於「管理進度」工作進行檢討，並於完成後

填寫實際完成日期。

C.表單樣本：

表 24 SPDP 移交階段-交付項目管制表

專案名稱：

階段工作負責人：

項次	交付項目名稱	交付方式	負責人	預計完成日期	實際完成日期	品保人員簽名	備註
		<input type="checkbox"/> 紙本 <input type="checkbox"/> 電子檔 <input type="checkbox"/>		//	//		
		<input type="checkbox"/> 紙本 <input type="checkbox"/> 電子檔 <input type="checkbox"/>		//	//		
		<input type="checkbox"/> 紙本 <input type="checkbox"/> 電子檔 <input type="checkbox"/>		//	//		
		<input type="checkbox"/> 紙本 <input type="checkbox"/> 電子檔 <input type="checkbox"/>		//	//		
		<input type="checkbox"/> 紙本 <input type="checkbox"/> 電子檔 <input type="checkbox"/>		//	//		

(3)不同使用者重覆詢問相同問題：藉由制定「協助客戶使用系統」及「回應客戶驗收測試結果」工作運用之「常見問題與建議解決方式表」(請參閱表 25)，以記錄並公告常見問題與解決方案，減少重覆解釋相同問題所耗費之成本：

A.應用方法：於「協助客戶使用系統」工作進行常見問題記錄，並於「常見問題與建議解決方式表」工作登記建議解決方案，並定期公佈給使用者。

B.應用時機：

a. 填表：於「協助客戶使用系統」工作進行本表子系統名稱、問題類型、常見問題描述填寫。

b. 檢視及更新：於「管理進度」工作進行建議解決方式、負責客服人員姓名、負責客服人員聯絡方式填寫。

C.表單樣本：

表 25 SPDP 移交階段-常見問題與建議解決方式表

專案名稱：

階段工作負責人：

項次	子系統名稱	問題類型	常見問題描述	建議解決方式	負責客服人員姓名	負責客服人員聯絡方式
		<input type="checkbox"/> 操作問題 <input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 其它問題				
		<input type="checkbox"/> 操作問題 <input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 其它問題				
		<input type="checkbox"/> 操作問題 <input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 其它問題				
		<input type="checkbox"/> 操作問題 <input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 其它問題				
		<input type="checkbox"/> 操作問題 <input type="checkbox"/> 系統瑕疵 <input type="checkbox"/> 擴增需求 <input type="checkbox"/> 其它問題				

(4)專案經驗無法有效累積至下一個專案問題：藉由制定「專案回顧檢討」工作運用之「專案獲得經驗記錄表」(請參閱表 26)，以有效記錄並管理專案獲得經驗：

A.應用方法：於「專案回顧檢討」工作進行檢討時由專案成員發表專案獲得經驗，並作成記錄，公告公司之專案成員，以確保公司專案執行成本得以有效降低。

B.應用時機：填表：於「專案回顧檢討」工作進行經驗提供者、專案角色、問題類型、發生問題描述、建議改進方式填寫。

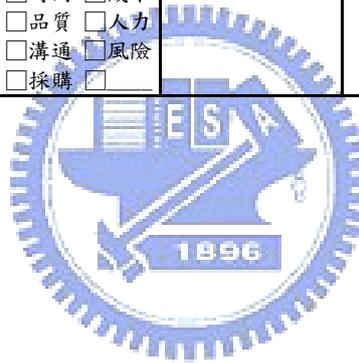
C.表單樣本：

表 26 SPDP 移交階段-專案獲得經驗記錄表

專案名稱：

階段工作負責人：

項次	經驗提供者	專案角色	問題類型	發生問題描述	建議改進方式	備註
		<input type="checkbox"/> 分析人員 <input type="checkbox"/> 設計人員 <input type="checkbox"/> 建置人員 <input type="checkbox"/> 測試人員 <input type="checkbox"/> _____	<input type="checkbox"/> 整合 <input type="checkbox"/> 範疇 <input type="checkbox"/> 時間 <input type="checkbox"/> 成本 <input type="checkbox"/> 品質 <input type="checkbox"/> 人力 <input type="checkbox"/> 溝通 <input type="checkbox"/> 風險 <input type="checkbox"/> 採購 <input type="checkbox"/> _____			
		<input type="checkbox"/> 分析人員 <input type="checkbox"/> 設計人員 <input type="checkbox"/> 建置人員 <input type="checkbox"/> 測試人員 <input type="checkbox"/> _____	<input type="checkbox"/> 整合 <input type="checkbox"/> 範疇 <input type="checkbox"/> 時間 <input type="checkbox"/> 成本 <input type="checkbox"/> 品質 <input type="checkbox"/> 人力 <input type="checkbox"/> 溝通 <input type="checkbox"/> 風險 <input type="checkbox"/> 採購 <input type="checkbox"/> _____			
		<input type="checkbox"/> 分析人員 <input type="checkbox"/> 設計人員 <input type="checkbox"/> 建置人員 <input type="checkbox"/> 測試人員 <input type="checkbox"/> _____	<input type="checkbox"/> 整合 <input type="checkbox"/> 範疇 <input type="checkbox"/> 時間 <input type="checkbox"/> 成本 <input type="checkbox"/> 品質 <input type="checkbox"/> 人力 <input type="checkbox"/> 溝通 <input type="checkbox"/> 風險 <input type="checkbox"/> 採購 <input type="checkbox"/> _____			
		<input type="checkbox"/> 分析人員 <input type="checkbox"/> 設計人員 <input type="checkbox"/> 建置人員 <input type="checkbox"/> 測試人員 <input type="checkbox"/> _____	<input type="checkbox"/> 整合 <input type="checkbox"/> 範疇 <input type="checkbox"/> 時間 <input type="checkbox"/> 成本 <input type="checkbox"/> 品質 <input type="checkbox"/> 人力 <input type="checkbox"/> 溝通 <input type="checkbox"/> 風險 <input type="checkbox"/> 採購 <input type="checkbox"/> _____			
		<input type="checkbox"/> 分析人員 <input type="checkbox"/> 設計人員 <input type="checkbox"/> 建置人員 <input type="checkbox"/> 測試人員 <input type="checkbox"/> _____	<input type="checkbox"/> 整合 <input type="checkbox"/> 範疇 <input type="checkbox"/> 時間 <input type="checkbox"/> 成本 <input type="checkbox"/> 品質 <input type="checkbox"/> 人力 <input type="checkbox"/> 溝通 <input type="checkbox"/> 風險 <input type="checkbox"/> 採購 <input type="checkbox"/> _____			



四、「軟體專案開發程序」之成效評估及優缺點

4.1 「軟體專案開發程序」之成效評估

4.1.1 「軟體專案開發程序」之成效評估目的

為確認本論文制訂之「軟體專案開發程序」於實務專案運用之可行性，故於本公司承接之專案前進行導入，並於專案進行過程予以運用，以評估實際執行成效。

4.1.2 「軟體專案開發程序」之成效評估作法

1. 成效評估作法說明

為量化評價「軟體專案開發程序」之執行成效，區分為以下五個步驟進行評估：

(1)專案遴選：

因「軟體專案開發程序」著重於軟體開發專案，故遴選之專案以資訊軟體勞務委外服務類為主。

(2)成本預估：

於本專案由三位專案經理分別精確預估專案所需人月數後及預估之專案毛利率，並取最低值作為本專案之預估基準。

(3)專案執行：

由成本預估最低之專案經理，經一週之密集訓練後，依「軟體專案開發程序」執行專案，直到完成專案及驗收。

(4)效果確認：

於專案開發完成驗收、系統上線後繼續觀察半年之維護情況，以確保軟體品質不會因採用新程序而降低。

(5)成效評估：

於專案完成保固六個月後對專案經理及專案成員進行優缺點調查及進行成本計算，包含保固期間所花費之成本後，計算實際執行之專案毛利率。

2. 專案背景資訊說明

- 專案領域：公共衛生領域軟體開發專案。
- 專案金額：議價後結標金額 181 萬。
- 專案起始日期：94 年 7 月 1 日。
- 專案結案日期：94 年 12 月 31 日前。
- 預估專案毛利率(含：保固六個月)：10%(與同業利潤標準相近)。
- 預估專案成本：162.9 萬
- 預估專案細項成本：

項目	成本類型	說明	數量	單位
1	管理行政	專案經理(兼系統分析)	3	人月
		行政雜支-印刷、交通	6	月
2	系統開發	系統分析	0	人月
		系統設計	2	人月
		系統實作	7	人月
		系統測試	2	人月
3	保固維護	提供7x24保固維護	6	月
4	教育訓練	教材印刷	40	份
		師資人力	2	次
		交通差勤	2	
總計	\$1,629,000			

3. 「軟體專案開發程序」之成效評估結果

(1) 優缺點調查結果

經彙總成果調查之結果如下：

A. 專案經理：

a. 優點：

- 避免導入前常有因工作繁忙而漏掉一些重要的工作。
- 專案過程的透明度增加，在專案進行中可以明確預估專案的後續進度。
- 專案過程有明確記錄，可以明確知道每位專案成員的產能與績效。
- 專案工作可以預先進行安排，故執行過程可以照表操課，執行過程與因專案成員工作爭議必須進行溝通的時間減少。

b. 缺點：比導入程序前，每一週平均多花了一小時進行管理表單填寫。

B.專案成員：

a. 優點：

- 專案過程中的工作負載較為平均，不會在專案末期才密集加班趕進度。
- 在專案早期即可得知後續要進行的時程及工作，可以及早準備。
- 在工作遭遇困難時，會即時被發現並得到協助。

b. 缺點：在專案工作過程中的壓力較大。

(2)成本之成效評估

本專案執行結果為提早半個月進行驗收，經實際計算後，專案細項成本如下：

項目	成本類型	說明	數量	單位
1	管理行政	專案經理(兼系統分析)	3	人月
		行政雜支-印刷、交通	6	月
2	系統開發	系統分析	0	人月
		系統設計	1.75	人月
		系統實作	6.25	人月
		系統測試	1.5	人月
3	保固維護	提供7x24保固維護	6	月
4	教育訓練	教材印刷	40	份
		師資人力	2	次
		交通差勤	2	
總計		\$1,475,020		

比較預估與實際成本如下：

- 將預估與實際成本依各「成本類型」製成比較表如下：

項目	成本類型	節省成本	節省比率
1	管理行政	\$12,180	3.0%
2	系統開發	\$140,000	13.6%
3	保固維護	\$0	0.0%
4	教育訓練	\$1,800	10.6%

說明：「節省比率」之計算方式=節省成本 / 預估成本。

- 預估與實際成本最主要差異為：系統開發節省約 1.5 人月，包含系統設計節省約一週、系統實作節省約三週及系統測試節省約兩週，合計節省成本 14 萬。
- 專案之實際成本為 147.5 萬，專案毛利率為 18.5%，比原先預估(10%)節省約 8.5%。

由於本程序是首次嚐試於專案中進行試用，專案經理及專案成

員對程序之熟悉程度尚未純熟已可達成如此成效，相信如在後續專案繼續運用必可達成更顯著之成效。



4.2 「軟體專案開發程序」之優缺點檢討

4.2.1 「軟體專案開發程序」的優點

1. 經驗較少的專案經理人得以省去冗長的學習及摸索過程，迅速套用本程序立即大幅提昇軟體開發專案的品質。
2. 俱豐富經驗的專案經理人可採用本程序減省專案規劃的時間及與專案新成員反覆溝通的過程，有效降低專案成本。
3. 所有專案關係人(Stakeholders)得以清晰、明確地了解專案的進行方式及在專案中主要任務，可以預先進行工作安排，提昇工作效率。

4.2.2 「軟體專案開發程序」的缺陷

1. 由於不同大小及性質的軟體開發專案可能因專案時間的急迫性、複雜度等因素，而無法完全適用本程序。
2. 雖然本程序已盡可能簡化軟體開發專案的必備工作，但是沒有系統輔助，執行者還是可能對既定工作產生遺漏或疏失。



五、 結論

軟體開發專案是資訊業界永遠的痛，規模大的軟體開發專案，往往就是時程延誤與超支預算的代名詞，加班趕程式進度是軟體工程師常有的夢魘，伴隨而來的是推出功能不完整、品質不佳的系統。

本論文以收集軟體開發專案實務面臨的挑戰為出發點，研究軟體工程技術「統一軟體開發流程」，進而發現部份導致軟體開發專案失敗的重要原因並非資訊技術可以完全解決，這些因素隨專案規模的增加、重要性的提昇而更形重要，這些原因可以隨著引進管理技術「專案管理知識體系」而獲得排除，然而於講求時效的業界，充份融合上述兩大理論並擬定適合的流程遭遇緩不濟急的務實困難，本論文遂結合實務經驗，制定「軟體專案開發程序」，提供業界可立即改進專案品質的有效方法。

本研究之貢獻有：

1. 串連「統一軟體開發流程」之各項工作，制定可行之「軟體專案開發程序」，業界可快速運用此程序進行軟體專案開發，並發揮「統一軟體開發流程」之優點。
2. 以規模 150~500 萬之軟體開發專案實務之參與人力，簡併「統一軟體開發流程」之各項工作，制定可行之「軟體專案開發程序」，業界得以立即適用，發揮「統一軟體開發流程」之優點。
3. 於「軟體專案開發程序」的各項工作中，逐步產出實務軟體專案需要交付之文件，有效節約專案時間並提昇軟體品質。
4. 以「專案管理知識體系」之管理精神輔助「統一軟體開發流程」制定「軟體專案開發程序」，有效提昇軟體開發專案之專案管理品質。
5. 以務實之角度，在「軟體專案開發程序」中提供關鍵之管理表單，得以立即套用，進行專案管理的標準化。

由於軟體開發專案之種類/規模迥異，限於時間及研究人力，仍有許多加強的空間：

1. 軟體專案之招標程序可分為「公開評選標」、「公開價格標」、「限制性招標」...等諸多類型，其「起動階段」之工作項目略有差異，本「軟體專案開發程序」係以「公開評選標」進行設計，其它招標方

式及非標案類型(如：民間之專案)之軟體開發專案，需確認本論文各階段工作類型為：「SPDP-因應本程序為軟體公司承接業主之軟體標案進行系統開發而須完成之工作」之適用性，並依實際情況予以調整，方可適用。

2. 本「軟體專案開發程序」係以規模 150~500 萬之軟體開發專案進行設計及進行實務驗證，更大型的專案可能需要予以客制化或採用其它模型(如：CMMI)，以符合更複雜的軟體專案開發需求，更小型的專案可能僅需採用部份「軟體專案開發程序」的工作項目。
3. 本「軟體專案開發程序」若能輔以適合之專案管理資訊系統進行，將可使執行更嚴謹，減少人治的因素，因研究時間/資源有限，並未進行專案管理資訊系統開發。



參考文獻

● 參考書籍

- [1] Jacobson, Booch, Rumbaugh, The Unified Software Development Process。
- [2] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) 2000 Edition。
- [3] Booch, Jacobson, Rumbaugh, The Unified Modeling Language User Guide。
- [4] IAN SOMMERVILLE, Software Engineering 6th Edition。
- [5] 趙光正, RATIONAL統一流程入門 第二版。
- [6] 趙光正, UML與樣式徹底研究-物件導向分析與設計及統一流程入門。
- [7] 張裕益, UML理論與實作-個案討論與經驗分享。
- [8] 鄭雯妮, 物件導向程式設計與分析。
- [9] 林信惠, 黃明祥, 王文良, 軟體開發專案管理。
- [10] 黃明祥, 軟體工程。
- [11] Phillippe Kruchten, The Rational Unified Process An Introduction 2nd Edition。

● 參考論文

- [12] Hayes, F., “Managing User Expectation.”, Computerworld, 31, pp. 8-9, 1997.
- [13] Davis, B., & Wilder, C., “False Start, Strong Finishes” , Information Week, 711, pp.41-53, 1998.
- [14] Gordon, P., “To Err Is Human, to Estimate, Divine.” , Information Week, 711, pp.65-72, 1999.
- [15] James J. Jiang, Gary Klein, T. Selwyn Ellis, “A Measure of Software

Development Risk” , Project Management Journal , September 2002, pp. 30-35, 2002.

- [16] Francies Hardman , Rafi A. Ashrafi, “Project Management in the Information Systems and Information Technologies Industries” , Project Management Journal, September 2002, pp.5-15, 2002.
- [17] James J. Jiang, Gary Klein, “Software Project Risks and Development Focus” , Project Management Journal, March 2000, pp.4-9, 2000.
- [18] Jan Terje Karlsen, Petter Gottschalk, “Management Roles for Successful IT Projects” , Project Management Vol.8 No.1, 2002.
- [19] Bin Jiang, “Key Elements of A Successful Project Manager” , Project Management Vol. 8 , No.1, 2002.
- [20] Shailesh Sood, “Critical-chain buffer management helps minimize risk in the project equation” , PM Network, March 2003.
- [21] James J. Jiang, Gary Klein, Thomas L. Means, “Project Risk Impact on Software Development Team Performance” , Project Management Journal, December 2000.

