

## 摘要

隨著現今環境快速變遷許多非例行性的工作也日漸增加，專案的進行方式較具有彈性與可變性，因此對於目前日益複雜的環境能有較好的適應性，將使得軟體業者在掌握專案上受到許多因素影響，造成專案大小的預估不易，致使在專案執行上可能會遭遇許多的風險。過去的軟體規模估算大多依賴經驗判斷法，如果專案管理者將開發成本估算的太低，可能會導致開發後期資源的不足，使工作無法如期完成，進而影響專案的品質。如果專案管理者將開發成本估算得太高，太多的資源投入專案，形成資源沒有妥善分配，造成資源的不當使用，對其他工作排擠。所以專案規模的估算，對於專案成功與否，佔有很重要的因素。而如何在專案初期進行可行性評估時，依據有限且粗略的需求，進行專案成本與時程的估算，更是極為困難。有鑑於此，本研究以專案風險分析的角度切入探討，經由內部標準化程序將事先預估或執行時曾經出現的風險因子，透過風險識別、評估等過程，找出影響專案執行的風險因子。再利用專案管理之工作分工結構結合影響專案執行的潛在因素，藉由數學預估模式進行專案規模預估並運用組織內之歷史資料庫來驗證導入風險評估機制是否使得軟體專案規模估算準確性更能有效提昇。最後，本研究探討之風險評估與專案預估模式，可提供既有軟體資訊服務業者於其公司內部導入風險管理及進行專案規模估算時的重要參考依據。

關鍵詞：風險評估、風險因子、專案規模預估、預估模型

## **Abstract**

Risk is uncertainty. Uncertainty resulting from diversity and complexity of the society is the key factor which makes the software project management difficult and uncontrollable. Risk management is, consequently, of great interest in implementing projects. Software companies used to base their project evaluation on the rules generalized from experiences, which always resulted in inaccurate evaluation of projects and uneven distribution of resources, and then influenced adversely the schedule and quality of the implementation of projects. In this regard, our objective in the paper is to explore the effect the risk evaluation mechanism can achieve to project management. A two-phase study was designed to select the influential risk factors and to test the accuracy of the model concerned. First, the internal SOP was employed to evaluate, induce, and analyze risk factors predicted by the project manager or met previously and to screen out the influential risk factors. To test the improvement of the accuracy of the model concerned, Work Breakdown Structure together with potential risk factors were estimated arithmetically to size the projects. Next, the contribution the risk evaluation mechanism made to the accuracy of the software project estimation was confirmed by historical data base possessed by the organization concerned. The study concludes with important implications for software companies when integrating risk evaluation into project sizing and estimating.

**Keywords :** Risk Evaluation 、 Risk Factor 、 Estimating Model 、 Estimation of the Size of Software Projects

# 目錄

第一章、緒論.....	1
1.1 研究動機與背景.....	1
1.2 研究目的.....	2
1.3 研究範圍與限制.....	3
1.3.1 研究範圍.....	3
1.3.2 研究限制.....	3
1.4 研究流程.....	4
1.5 研究架構.....	7
第二章、文獻探討.....	10
2.1 軟體規模預估方式探討.....	10
2.1.1 IFPUG 功能點分析法.....	12
2.1.2 Mark II 功能點分析法.....	16
2.1.3 建構成本模型 (COCOMO).....	18
2.1.4 使用案例功能點分析法.....	20
2.1.5 專家判斷法.....	26
2.1.6 類比法.....	27
2.2 專案風險.....	27
2.3 工作分解結構.....	30
2.4 變異數分析(ANOVA).....	32
2.5 實證軟體工程.....	33
第三章、專案風險分析方法.....	35
3.1 軟體風險因子.....	35
3.2 風險管理.....	38
3.3 專案風險識別.....	38
3.4 專案風險評估.....	41

3.5 專案風險分級 .....	42
第四章、專案規模預估模型設計 .....	44
4.1 線性規劃 .....	45
4.2 專案規模預估模型 .....	49
4.3 估算流程與方法 .....	53
4.4 預估模型與企業資訊入口系統整合 .....	56
第五章、個案研究：某學術研究中心 .....	64
5.1 個案描述 .....	64
5.2 個案組織內部流程探討 .....	65
5.3 資料來源 .....	70
5.4 資料分析 .....	72
第六章、研究結果 .....	75
6.1 分析結果 .....	75
6.2 估算模型適合性測試 .....	77
6.3 估算品質測試 .....	78
第七章、結論與建議 .....	80
7.1 研究貢獻 .....	80
7.2 後續研究建議 .....	81
參考文獻 .....	83
中文文獻 .....	83
英文文獻 .....	84
附錄 .....	87
附錄一 專案主要工作項訓練資料 .....	87
附錄二 本文預估模型檢測資料估算品質測量數據 .....	89
附錄三 Randy K. Smith 估算模型檢測資料估算品質測量數據 .....	91

## 表目錄

表 2-1：ISO 國際標準的功能點分析法 .....	11
表 2-2：14 項通用系統特徵說明 .....	13
表 2-3：功能點分析計算表 .....	15
表 2-4：Mark II 技術複雜度因子說明 .....	17
表 2-5：COCOMO 公式係數表 .....	19
表 2-6：COCOMO 資料集構面與影響因子說明 .....	19
表 2-7：角色複雜度權重對應表 .....	21
表 2-8：角色複雜度權重計算表 .....	21
表 2-9：使用案例分類 .....	22
表 2-10：未調整使用案例權重計算表 .....	22
表 2-11：未調整功能點 .....	23
表 2-12：複雜度影響性與權重對應表 .....	23
表 2-13：技術複雜度因子說明與加權值對應表 .....	23
表 2-14：技術複雜度因子計算表 .....	24
表 2-15：環境複雜度因子說明與加權值對應表 .....	25
表 2-16：環境複雜度因子計算表 .....	25
表 2-17：專案定義 .....	28
表 2-18：專案風險的定義 .....	30
表 2-19：ANOVA 表 .....	32
表 3-1：軟體風險因子相關文獻 .....	36
表 3-2：Linda Wallace 提出的風險構面與風險因子 .....	36
表 3-3：專案工作細項分類表 .....	40
表 3-4：軟體專案風險影響權重表 .....	41
表 3-5：專案工作細項風險識別表 .....	42
表 3-6：專案風險評估表 .....	43
表 4-1：迴歸分析訓練資料表 .....	47
表 4-2：工作項目的時間離散程度計算範例 .....	52
表 5-1：迴歸分析表 .....	73
表 5-2：迴歸係數的變異數分析(ANOVA)摘要表 .....	73
表 5-3：模式摘要表 .....	73
表 6-1：各項因子於本研究模型中的人天數影響性 .....	75
表 6-2：線性迴歸模式的適合性測量結果 .....	78
表 6-3：估算品質測試結果 .....	79

## 圖目錄

圖 1-1：研究流程.....	6
圖 1-2：研究架構.....	9
圖 2-1：功能點分析法發展歷程.....	11
圖 2-2：風險與不確定性區分圖.....	29
圖 2-3：WBS 範例.....	31
圖 3-1：Boehm 的風險管理架構.....	38
圖 3-2：依據專案工作細項分類之 WBS 內容說明.....	40
圖 4-1：軟體規模估算影響構面.....	45
圖 4-2：WBS 結構內容.....	53
圖 4-3：軟體專案規模與成本估算流程.....	53
圖 4-4：專案研擬單.....	57
圖 4-5：專案研擬成立審核流程圖.....	58
圖 4-6：專案工作項目細項分類設定.....	59
圖 4-7：專案項目管理服務.....	59
圖 4-8：由專案 WBS 內容可取得相關專案規模估算因子.....	60
圖 4-9：專案成員工時回報介面.....	60
圖 4-10：專案成員工時回報記錄.....	61
圖 4-11：專案工作細項分類工時投入統計圖.....	62
圖 4-12：專案工作項目細項分類工時比率與工時統計圖.....	62
圖 4-13：專案工作項目與團隊成員工時投入明細統計圖.....	63
圖 4-14：專案整體工作項目投入工時分佈比率圖.....	63
圖 5-1：個案專案生命週期與專案流程.....	66
圖 5-2：工作日誌系統-專案 WBS 工作項規劃.....	71
圖 5-3：工作日誌系統-工時填寫回報.....	71

# 第一章、緒論

## 1.1 研究動機與背景

近年來，隨著網際網路的蓬勃發展，資訊快速散佈流通，帶動了知識經濟社會的來臨，間接也促使電腦與軟體相關產業的成長迅速，軟體亦隨著電子化與資訊化的趨勢而獲得高度重視，也為軟體產業帶來無限的商機。為了滿足政府、民間企業與個人多樣化的需求，軟體規模變得愈來愈龐大，軟體需求複雜度也日漸提高。然而在軟體開發的專案中，顯然面臨著高風險的環境，如果無法在短時間內開發完成或者無法滿足顧客的需求，將會導致開發失敗而造成企業嚴重的損失。再加上軟體開發需要高度依賴人力智慧與創新，面臨著適度不確定性，因此，如何準確評估風險對專案的影響，並制定有效的風險評估機制以確保軟體品質，成為軟體業者最迫切與困擾的議題之一。

軟體開發商想要提供穩定且高品質的軟體產品，就必須找出可能影響專案執行的影響因素，讓專案管理者能夠精確地預估軟體開發工時、人力與時程，進一步得知專案規模來進行成本預估、資源分配、時程安排與人員投入等專案規劃活動。軟體開發的工作量、成本與時間具有高度不確定性，如果專案管理者將開發成本估算的太低，可能會導致開發後期資源的不足，造成時程延誤，使得工作無法如期完成，進而影響專案的品質。如果專案管理者將開發成本估算得太高，投入太多專案資源，形成資源沒有妥善分配，對於其他工作產生排擠效應，造成資源的浪費。

為了有效確保軟體品質，必須有效率的進行專案管理及精確地預估出軟體開發的工作量才能夠協助專案管理者在專案開發的早期即有效地控管開發中所有的資源，也才能提昇日後開發完成的軟體品質。因此建構能夠精確地預估專案規模大小與成本，以提供專案管理者決策與管理之用。近年來，專案管理手法與學說方興未艾，儼然成為一個軟體系統成功與否與能否達到所預期的功能要求的關鍵。然而在專案管理過程中工作分解結構(Work Breakdown Structure, WBS)扮演相當重要的角色，工作分解結構中包含了專

案的工作項目、工作項目的時程安排、工作項目的相關執行人員等專案所需的重要資料。這些資料也常被用來作為預估專案規模的參考依據。

目前軟體開發商在估算軟體大小規模時，大部分是依據資深工程師經驗或是類似已完成專案的經驗為基礎來度量估計軟體大小及複雜度，再依當時可供投入人力與相關資源初步估算所需之工作人月與人員生產力，進而推算開發成本。因此預估值的精確度就受到很大的質疑。早期度量軟體規模與複雜度的方法，為計算程式碼的行數(Line of Code, LOC)，可是軟體系統的LOC無法在專案開始之前得知，必須要等到軟體系統發展完成後才能知道。且和所使用的程式語言與各個程式設計師的習慣而有所不同，所以亦不適合用來預測軟體發展所需的時間、人力與成本。

所以本文提出一套結合風險管理的專案預估方法來進行專案各項工作風險分析評估與規模預估。因此，如果能夠精確快速地預估軟體開發所需的工作量，以及預先找出可能遭遇到的專案風險不僅可以協助專案管理的規劃與監控等工作更有效進行，更可幫助專案管理者在軟體發展生命週期(Software Development Life Cycle, SDLC)各階段妥善地規劃時程、人力、資源配置與時間控管，進而達成使用者對軟體品質的要求。

## 1.2 研究目的

有鑑於軟體專案大小的預估不易，有許多的軟體開發工作量預估方法被相繼提出，這些方法主要有專案建議(Expert Consulting)、類比預估(Estimation by Analogy)與演算法塑模(Algorithm Modeling)三種方法，其中又以透過統計方法或其他數學方法產生預估模式的演算法塑模較能提供客觀的預估結果[4]，在演算法塑模中又以統計方法最被廣泛應用，例如功能點分析法就是使用統計分析法所產生的預估模式，而這種方法是由1970年代中期開發出來，試圖克服以程式碼行數來估計系統規模所存在的難度，開發出一種能夠預期軟體發展相關工作量的機制，提出了以軟體功能為觀點的一種測量方式。這個測量方式稱為功能點(Function Point, FP)的測量。目前提倡功能點分析方法的機構，主要以IFPUG、Mark II、COSMIC等組織為主。其中IFPUG已盛行美國

多年，Mark II則在歐洲流行，COSMIC則是近年才成立的組織。

透過文獻探討現有的功能點分析方法與風險管理流程，本文提出一套結合風險管理的專案預估方法。本研究的主要目的有：(1). 建構軟體專案風險的評估架構；(2). 制定「軟體專案規模與成本估算之標準作業」以利其他軟體資訊服務業者導入風險管理及進行專案規模估算時的重要參考依據；(3). 驗證加入風險因子之專案規模估算模型是否有助於軟體專案規模預估準確度的改善。

本研究預估模型的推導與建立是利用個案研究的方式，分析過去所完成的各類專案歷史資料，以統計迴歸(Statistical Regression)的方法推導出專案規模大小估算模式，進而得知專案團隊估計工作量，如此一來，便能夠協助軟體資訊服務業者，在專案規劃階段，利用現有專案管理工作項目、人力、時程與識別的風險因子，合理的預估軟體專案之開發成本，藉以妥善的分配資源，適時的掌握、控制並管理各項資源及預算。

## 1.3 研究範圍與限制

### 1.3.1 研究範圍

本研究探討以投入人天數為專案大小估算基礎的專案規模預估模型，引用Linda Wallace的研究調查所蒐集的軟體風險因子[44]，以及Boehm[28]的軟體風險管理架構進行風險分析作業，結合Randy K. Smith的工作指派模型[49]，提出本研究預估模型。雖然本文探討事先的風險分析如何有助於提昇專案規模預估的準確度，但隨著專案類型的不同，其所面臨的風險因子也就不盡相同，而風險因子影響專案的程度也不同，因此，本研究所得到的專案規模預估模型只適用於軟體開發類型的專案。

### 1.3.2 研究限制

一、由於軟體開發專案的種類相當多，隨者軟體專案大小的不同，在使用案例(Use Case)大小的認定上常因人而異，為讓使用案例大小認定上趨於

- 一致，本研究假設專案WBS中主要工作項目為使用者案例。
- 二、由於軟體開發專案的種類相當多，在發展過程中所遭遇的風險會有很大的差異；為使風險分析作業標準化，所以本研究引用2004年Linda Wallace針對507位具有PMP證照的專案管理者進行研究調查所蒐集整理的軟體風險因子，評估風險對專案的影響，以避免專案管理者的主觀認知差異，使得識別出的風險因子過於發散而影響風險評估，進而減低軟體規模估算品質。
  - 三、本研究是探討軟體專案規模預估模型，對於專案管理者、系統開發者的軟體開發教育背景、軟體開發專業經驗和功能點教育訓練程度，都會影響功能點數計算精確度，進而影響到軟體專案規模估算結果，將不包含在本文的探討範圍。
  - 四、隨著地區、文化以及環境的不同，不同區域的專案風險管理差異甚大，其所面臨的狀況也不同。因此，本研究所提出的預估模型與風險分析評估結果對於專案時程影響度，主要探討的對象是台灣地區某大型研究機構之軟體發展專案，由於專案風險會隨著個人認知上的不同而產生差異，因此，本研究的結果並不包含專家對風險認知的差異性。

## 1.4 研究流程

本研究採用個案探討方式，以WBS與2001年Randy K. Smith提出的專案工作指派模型(Task Assignment Patterns)為本研究之軟體專案規模估算模型基礎[49]，並採用Boehm[28]提出的風險分析架構，依照風險識別、風險評估與風險分級三個步驟的風險評估機制，引出軟體專案發展過程中所面臨的風險因子，將風險因子的觀念加入到專案工作指派模型中，以人力工時配置概念來衡量風險所產生的工作時程影響，進而建立專案風險評估流程，以降低企業在軟體專案風險發生的機率，或是能在風險發生時，將風險對專案的衝擊降至最小。本文假設專案WBS中主要工作項目為使用者案例，應用Karner, G.於1993年所提出使用者案例功能點分析方法[42][53]，以使用者觀點來度量專案複雜度。

如何建構能有效預估專案規模，以提供專案管理者決策與管理之用，成

為一個軟體系統成功與否，以及能否達到所預期要求的關鍵因素。然而在專案管理過程中WBS扮演相當重要的角色，WBS中包含了專案的工作項目、工作項目的時程安排、工作項目的相關執行人員等專案所需的重要資料。這些資料也常被用來作為預估專案規模參考依據。在相關的研究文獻中指出一般專案含有五項潛在因素，即所謂的人員平均年資、專案參與人數、時程緊密度、人員並行程度和時間離散度[23][27][32][37][38]。本研究整合這五項影響因素，並加入專案風險評估，依據2004年Linda Wallace提出之風險構面與風險因子，進一步識別出影響軟體專案的潛在風險因素，衍生Randy K. Smith估算模型[49]，提出以專案WBS為基礎的專案規模估算模型。

首先，將所蒐集而得的歷史資料，或稱為訓練資料(Training Data)，以複迴歸分析法建立專案規模評估方式，計得檢測用的估算參數；同樣地以相同的歷史資料，依照Randy K. Smith提出的估算模型分析方式，統計求得Randy K. Smith估算模型分析檢測用的估算參數。然後，再將欲估算的實際資料，或稱為檢測資料，分別代入本研究預估模型與Randy K. Smith估算模型等兩種專案規模預估方式，經由上述訓練階段所得到的參數和模式中，測得專案規模估算值，並求得兩種方式各自的準確度。

最後，針對本文所提出的模型及Randy K. Smith估算模型進行線性迴歸模式適合性(Goodness of Fit)與估算品質(Quality of Estimation)的測試，進一步驗證了解本研究模型估算的能力。透過這兩大類的測試，可以了解導入風險評估機制後歷史資料與所建立的模型契合程度，以及本研究模型和Randy K. Smith估算模型各自所測得的準確度，進行此兩種模式估算專案規模能力的比較，也可藉此得知本文提出之估算模型對於新資料的估算品質及準確度改善情形。本研究模型適用範圍大則涵括整個專案生命週期，從專案起始化、專案規劃、專案執行、專案監控、專案驗收，小則僅涵蓋某些階段。使得專案管理者能於軟體發展生命週期早期，使用此模型來估算軟體專案人力，用以軟體專案人力配置的參考指標。此舉，能協助專案管理者先行了解軟體專案的人力狀況，降低軟體專案失敗的可能性。本文研究流程如下：

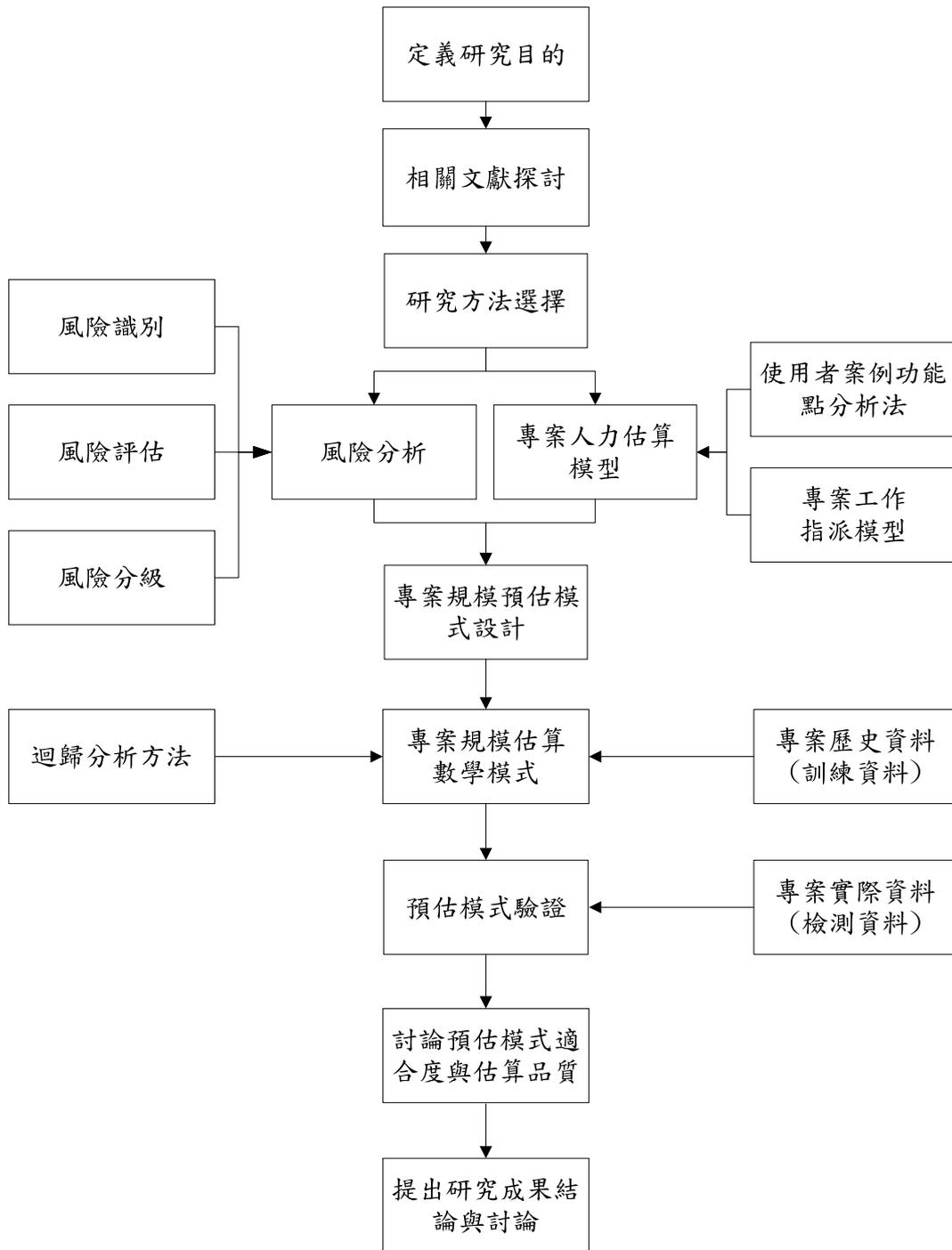


圖 1-1：研究流程

## 1.5 研究架構

本論文共分為七章，各章內容摘要以及研究架構內容概述如下：

### 第一章 緒論

敘述本研究論文的研究背景、研究動機、目的、範圍、限制，並對研究之架構及流程進行說明，最後略述本文各章之內容。

### 第二章 文獻探討

探討目前一般常見的軟體規模預估方式，論述多位學者度對於專案的定義與說明何謂專案風險，接著談論何謂工作分解結構(WBS)，並闡述變異數分析法，以用於後續驗證本研究之實驗設計，最後訴諸於實證軟體工程的應用。

### 第三章 專案風險與分析方法

本文的專案風險分析方法是透過研究個案內部的專案流程，配合文獻探討及採用Linda Wallace提出的風險因子與Boehm提出的風險分析架構，以專案歷史資料為統計主體加以分析而成。利用這個專案風險分析方法來進行風險識別與評估，最終得到專案規模估算影響因素之一的專案風險因子：工作項目的風險影響時程。

### 第四章 專案規模預估模型設計

專案規模估算數學模型的設計，基礎於使用案例功能點分析法與Randy K. Smith所提出的工作指派模型並加入專案風險因子，以專案人力配置所花費的工作人天為專案規模之計算單位，設計出專案規模估算數學模型。

### 第五章 個案研究

以台灣地區某機構為案例，探討個案組織內之流程與專案特

性，並於該個案2005年至2008年間所執行的專案選擇出符合該單位之發展方向與歷年重大成果之代表性專案，當成本研究進行迴歸分析的訓練資料來源，共53筆，經過迴歸統計運算與變異數分析等步驟，得到本研究之專案規模估算數學模型。

## 第六章 研究結果

針對研究結果進行線性迴歸模式的適合性測試，以及估算品質測試。透過這些測試可以了解歷史資料與所建立的模型契合程度，也可以知道模型對於新資料的估算品質，最後得到專案預估的準確度。

## 第七章 結論與建議

總結研究的成果，提出一個可供其他組織參考的專案規模估算模式與施行步驟；闡明此本研究成果的貢獻與價值，及提出一些未來在企業實務上或學術界上，仍值得去努力探討的空間。

第一章	緒論	研究動機與背景 研究目的 研究範圍與限制 研究流程 研究架構
第二章	文獻探討	軟體規模預估方式探討 專案風險 工作分解結構 變異數分析 實證軟體工程
第三章	專案風險 分析方法	軟體風險因子 風險管理 專案風險識別 專案風險評估 專案風險分級
第四章	專案規模 預估模型設計	線性規劃 專案規模預估模型 估算流程與方法 預估模型與企業入口系統整合
第五章	個案研究	個案描述 個案組織內部流程探討 資料來源 資料分析
第六章	研究結果	分析結果 估算模型適合性測試 估算品質測試
第七章	結論與建議	研究貢獻 後續研究建議

圖 1-2：研究架構

## 第二章、文獻探討

本章節文獻探討之順序，先針對目前一般常見的軟體規模預估方式進行探討，緊接著論述多位學者對於專案的定義與說明何謂專案風險，為第三章論述風險分析分法之背景知識，接著談論何謂工作分解結構(WBS)，藉由WBS之說明幫助讀者於後續章節中了解如何由WBS中取得相關專案規模預估因子，再來闡述變異數分析法，讓讀者了解這種已被廣泛推廣至各種實驗領域的實驗設計工具(分析相關數據將用於後續本文實驗設計之驗證)，最後訴諸於實證軟體工程探討與步驟，本文預估模型也將參照Kitchenham等學者們所提出之實證軟體工程步驟[43]進行實驗設計驗證。

### 2.1 軟體規模預估方式探討

軟體規模估算是估計軟體發展的工作量、成本與資源需求的基礎，通過規模與其他度量資料還可以度量專案的生產率、缺陷密度，目前常見估算是代碼行估算方法和功能點分析方法(Function Points Analysis, FPA)。代碼行估算方法是一種經驗估算方法，通常會採用PERT sizing方法和DELPHI方法，估計結果與估計的人員、使用的開發工具緊密相關，而功能點分析法則比較客觀。FPA最早由IBM的工程師Allan Albrecht於1970年代提出[26]，隨後被國際功能點用戶協會(The International Function Point Users Group, IFPUG)提出的IFPUG方法繼承。IFPUG功能點分析方法在美國盛行多年，這種方法主要適合於資訊系統的規模估算。基於Allan Albrecht的功能點方法，又發展出了多種方法，相關功能點方式開發歷程以及目前被ISO組織接受為國際標準的功能點分析方法有4種，如表2-1、圖2-1所示：

表 2-1：ISO 國際標準的功能點分析法

提出單位	ISO 國際標準	標準名稱
國際功能點用戶協會 (IFPUG)	ISO/IEC 20926:2003	Software engineering - IFPUG 4.1 Unadjusted functional size measurement method - Counting Practices Manual
荷蘭軟體度量協會 (NESMA)	ISO/IEC 24570:2005	Software engineering - NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis
英國軟體度量協會 (UKSMA)	ISO/IEC 20968:2002	Software engineering - Mk II Function Point Analysis - Counting Practices Manual
通用軟體度量國際協會 (COSMIC)	ISO/IEC 19761:2003	Software engineering - COSMIC-FFP - A Functional Size Measurement Method

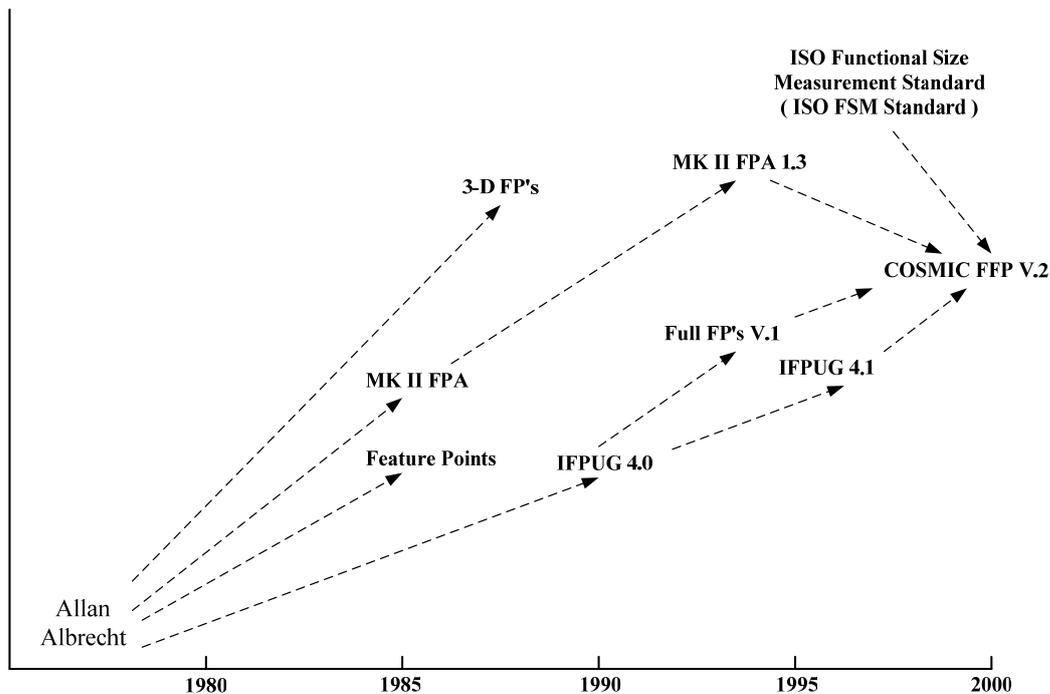


圖 2-1：功能點分析法發展歷程

軟體規模預估方式無論是演算法、專家判斷法或類比法等，必須配合資料的收集。數據愈明確、數量愈豐富，則估算的準確性就隨之提高。資訊組織過去的軟體規模估算，幾乎全採行專家判斷法或類比法。倘若一個資訊組織已建立良好的制度，並能就以往軟體開發專案的數據加以整理，即可採行演算法估算[4]。演算法是以迴歸分析的方法來估計，近年來人工智慧的發展，

也提供了其他的方法論，如類神經網路法與基因演算法等。軟體規模預估方式種類繁多，分類無一定標準，本文將針對功能點分析方法中的IFPUG功能點分析法、Mark II功能點分析法、建構成本模型（COCOMO）、使用案例功能點分析法，以及專家判斷法、類比法進行說明，下列各小節將針對這些常見的軟體規模預估方式進行說明。

### 2.1.1 IFPUG 功能點分析法

在1970年代末期，IBM公司認為有必要開發一套與程式語言無涉，衡量軟體發展成果的方法，它的一位研究人員Allan Albrecht，於是發展出功能點估算。80年代早期，IBM公司GUIDE部門，將功能點技術整理並出版了計算功能點的手冊。80年代末期一個以美國、加拿大為主的國際性學術組織組成了國際功能點用戶協會IFPUG並提出IFPUG功能點分析法，到了2003年IFPUG出版了Counting Practice Manual 4.1版[39]，同時宣稱它一直繼承了20年前Allan Albrecht原設計的理念，即「功能點是一種量測，用來衡量電腦應用系統或者軟體專案的大小。這種衡量是從功能或者使用者的觀點來看，與所使用的程式語言工具、系統開發的方法論及專案開發團隊的技術與能力無涉」。其表示的單位為五個主要構面功能點總和乘上14項通用系統特徵可調整因子，又稱為14項通用系統特徵[39]。

#### ◆ 五個主要的功能點分析構面

功能點分析構面元件的界定以一個應用系統為界限，內含有五個估算構面主要分為二大類：動態交易類和靜態檔案類，而每一類型所包含的構面及定義簡述如下：

#### ◆ 動態交易類，包括以下三個構面：

1. 外部輸入(External Inputs, EI)：當資料藉由輸入畫面直接輸入或來自其他應用系統，這些資料是由該系統的外部帶至系統裡面，並且用來維護一個以上的內部資料檔，這些資料便屬於外部輸入構面。

2. 外部查詢(External Inquiry, EQ)：當資料擷取內部檔案或外部檔案資料。以電腦畫面或報表型式顯示，與外部輸出不同是只擷取資料不能對任何的內部檔案和外部檔案進行維護，不產生經由邏輯運算或計算公式衍生出來的資料，這類資料便屬於外部查詢構面。
3. 外部輸出(External Outputs, EO)：當資料由多個內部檔案或外部檔案經過邏輯運算或計算公式所衍生出來，不僅可以用來更新內部檔案，並且可以報表形式輸出，或輸出到其他應用系統的檔案，這類資料使屬於外部輸出構面。

◆ 靜態檔案類，包含下述二個構面：

1. 外部介面檔(External Interface Files, EIF)：外部介面檔案不歸屬於該應用系統，乃是由其他的應用系統維護並擁有，亦即為另一個應用系統的內部邏輯檔。因此，外部介面檔至少生被一個外部輸入、外部輸出或外部查詢所使用。
2. 內部邏輯檔(Internal Logical Files, ILF)：由一群有意義且可辨識的資料組成，永久保存在該應用系統內部，透過外部輸入來進行新增、更新或刪除等維護作業。因此，內部邏輯檔至少會被一個外部輸出或者外部查詢所使用。

◆ 14項通用系統特徵可調整因子為：

表 2-2：14 項通用系統特徵說明

14項通用系統特徵		說明
1	資料通訊	運作在資料傳輸上的設備與複雜度
2	分配式資料處理	資料或功能是否分配處理？
3	績效衡量	是否滿足使用者特殊的需求，如回應時間在數秒之內？
4	硬體平台負載度	是否將在一部作業擁擠的電腦上執行，因此程式編寫要特別考慮？
5	交易頻率	交易頻率是每日、每週，或者每月等？
6	線上輸入比例	線上資料輸入的比例
7	使用者效率	是否特別顧慮使用者的效率訴求？

8	線上更新	多少個內部檔案是採線上更新？
9	處理的複雜度	系統邏輯與運算是否具有延伸性？
10	多人使用性	是否多人可同時使用？
11	安裝親和度	是否有強調轉換與安裝的便利性？
12	操作親和度	是否考慮啓動、備援及復原程序？
13	多地區/多平台	是否會給好幾個電腦中心使用？
14	可轉移性	是否會要求特殊設計，以便於修改？

◆ 主要計算流程為：

◆ 步驟一：決定功能點計算型態

功能點計算的型態有新開發專案功能點計算、修改重整專案功能點計算與維護現有應用系統功能點計算。

◆ 步驟二：決定應用系統的範圍

應用系統範圍的確認，可從系統規格書、使用者需求規範和資料流程圖等系統開發文件取得相關資訊，經與提出需求者洽談溝通確認。

◆ 步驟三：確認動態交易類構面

依據各構面評價條件予以評價，以求得交易類未調整功能點。動態交易類構面包含有外部輸入(EI)、外部輸出(EO)及外部查詢(EQ)等構面，除了確認所屬交易類構面外，尚須統計每一構面含有多少資料項型態(Data Elements)，而這些資料項涉及了那些檔案數例如：內部邏輯檔案數(ILF)或是外部介面檔案數(EIF)，根據各構面評價條件，即以資料項數(Data Elements)對照檔案數條件之不同予以低、中、高功能需求評價，再依照各構面不同功能需求賦予不同的未調整功能點。

◆ 步驟四：確認靜態檔案類構面

依據各構面評價條件予以評價，以求得檔案類未調整功能點。靜態檔案類構面有內部邏輯檔(ILF)、外部介面檔(EIF)等構面，除了確認應用系統涵蓋了幾個內部邏輯檔，或者涵蓋了幾個外部介面檔外，還須要統計所有的交易

使用了該檔案多少資料項(Data Elements)及記錄數(Record Elements)，而這個記錄數係指使用者能分辨的資料項子集合。根據各構面評價條件，即以資料項數對照記錄數條件之不同予以低、中、高功能需求評價，再依照各構面不同功能需求賦予不同的未調整功能點。

◆ 步驟五：決定功能點調整因子

功能點調整因子(Value Adjustment Factor, VAF)，係以十四項通用系統特徵(General System Characteristics, GSC)作為調整因子，計算出調整因子數值。依照每一項通用系統特徵作業方式和程序複雜度分別給予0到5的量值(Degrees of Influence, DI)，分別代表其影響程度從沒有影響(No Influence)至很大的影響(Strong Influence)，加總後代入功能點調整因子VAF的公式： $(65 + \text{通用系統特徵GSC的影響值和})/100$ ，結果求得調整因子值(VAF)。

$$VAF = 0.65 * 0.01 * \sum_{i=1}^{14} DI_i$$

$DI_i$ ，表示14項通用系統特徵GSC的影響值

◆ 步驟六：計算調整後功能點

將未調整功能點總計(UAF)乘以功能點調整因子(VAF)，所得結果即為調整後功能點(FP)。有關功能點的計算公式如下，而計算過程可以表列方式呈現，如下表所示：

表 2-3：功能點分析計算表

FPA構面	構面複雜度			
	低	中	高	總計
外部輸入(EI)	$FP_{EI} * 3 = FP1_{EI}$	$FP_{EI} * 4 = FP2_{EI}$	$FP_{EI} * 6 = FP3_{EI}$	$\sum_{i=1}^{n=3} FPi_{EI} = UFP1$
外部輸出(EO)	$FP_{EO} * 4 = FP1_{EO}$	$FP_{EO} * 5 = FP2_{EO}$	$FP_{EO} * 7 = FP3_{EO}$	$\sum_{i=1}^{n=3} FPi_{EO} = UFP2$
外部查詢(EQ)	$FP_{EQ} * 3 = FP1_{EQ}$	$FP_{EQ} * 4 = FP2_{EQ}$	$FP_{EQ} * 6 = FP3_{EQ}$	$\sum_{i=1}^{n=3} FPi_{EQ} = UFP3$
內部邏輯檔(ILF)	$FP_{ILF} * 7 = FP1_{ILF}$	$FP_{ILF} * 10 = FP2_{ILF}$	$FP_{ILF} * 15 = FP3_{ILF}$	$\sum_{i=1}^{n=3} FPi_{ILF} = UFP4$
外部介面檔(EIF)	$FP_{EIF} * 5 = FP1_{EIF}$	$FP_{EIF} * 7 = FP2_{EIF}$	$FP_{EIF} * 10 = FP3_{EIF}$	$\sum_{i=1}^{n=3} FPi_{EIF} = UFP5$

未調整功能點加總( $UAF = \sum_{i=1}^{n=5} UFP_i$ )	UAF
調整因子(VAF)	VAF
調整後功能點( $FP = UAF * VAF$ )	FP

功能點分析的計算方式除了上述現有應用系統維護外，對於現有應用系統功能新增加強、新開發軟體系統專案或是重整軟體系統專案等功能點計算，分述如下：

- ◆ 現有應用系統功能新增加強，功能點計算公式：

$$FP = AUFP * VAF$$

AUFP, 表示新增部分未調整前功能點

- ◆ 新開發軟體系統專案·功能點計算公式：

$$FP = (UFP+CFP) * VAF$$

UFP, 表示新開發系統部分未調整前功能點

CFP, 表示現有應用系統作業未調整轉換功能點

- ◆ 重整軟體系統專案，功能點計算公式：

$$FP = (AUFP+CUFP-DUFP+CFP) * VAF$$

AUFP, 表示新增系統功能部分未調整前功能點

CUFP, 表示變更系統功能部分未調整前功能點

DUFP, 表示刪除系統功能部分未調整前功能點

CFP, 表示現有應用系統作業未調整轉換功能點

## 2.1.2 Mark II 功能點分析法

Mark II 是在歐洲流行的 FPA 方法[55]。英國軟體度量協會所提出的 Mark II 功能點分析是由從傳統功能點分析所演變出來的，由 Charles Symons 所提出，與 IFPUG 相較 Mark II 功能點減少對檔案處理的主觀性以確保整個系統整體度量和部分計算總和的結果一致，目的是提供一個更優質計算系統內部處理的複雜度方法[54]。Charles Symons 認為傳統功能點分析中關於複雜度的

設定過於簡單，將導致較為複雜的項目不能給予正確設定的主因。

Mark II 將整個系統視為一連串的邏輯處理程序(Logical Transactions)，每個處理程序包含了輸入、輸出和程序三個元件。Mark II 是從使用者所觸發的事件來定義一個邏輯處理程序，此與傳統功能點分析切入的觀點不同，Mark II 中每一個邏輯處理程序都包含了含輸入、輸出和處理程序三個元件。例如，(1).建立一個使用者檔案；(2).存儲一個資料檔案；(3).登入一個系統。Mark II 更進一步的延伸了傳統功能點分析其通用系統特徵因子，再額外增加了 5 項影響因子，共 19 項技術複雜度因子(Technical Complexity Adjustment factor，TCF) 中的因子(T1~T19)，其各項因子定義如下：

表 2-4：Mark II 技術複雜度因子說明

19 項技術複雜度因子		說明
1	資料通訊	運作在資料傳輸上的設備與複雜度
2	分配式資料處理	資料或功能是否分配處理？
3	績效衡量	是否滿足使用者特殊的需求，如回應時間在數秒之內？
4	硬體平台負載度	是否將在一部作業擁擠的電腦上執行，因此程式編寫要特別考慮？
5	交易頻率	交易頻率是每日、每週，或者每月等？
6	線上輸入比例	線上資料輸入的比例
7	使用者效率	是否特別顧慮使用者的效率訴求？
8	線上更新	多少個內部檔案是採線上更新？
9	處理的複雜度	系統邏輯與運算是否具有延伸性？
10	多人使用性	是否多人可同時使用？
11	安裝親和度	是否有強調轉換與安裝的便利性？
12	操作親和度	是否考慮啓動、備援及復原程序？
13	多地區/多平台	是否會給好幾個電腦中心使用？
14	可轉移性	是否會要求特殊設計，以便於修改？
15	其他應用介面	是否有其他應用系統介面？
16	特殊安全特性	是否特殊安全特性考量？
17	外掛工具存取需求	是否要透過外掛工具直接存取的需求？
18	使用者特別訓練	是否針對使用者進行特殊訓練？
19	文件化需求	是否需要提供系統相關文件需求？

### 2.1.3 建構成本模型 (COCOMO)

Boehm所發表的建構成本模型(Constructive Cost Model, COCOMO)[29]將軟體開發工作量分成三種等級，即基本模式(Basic)、中級模式(Intermediate)和詳細模式(Advanced)三種等級。每種模式又依軟體複雜度分為三種，分別為有機型(Organic Mode)、中間型(Semi-Detached Mode)、內嵌型(Embedded Mode)；所謂有機型，則是小型、簡單、低複雜度、低風險的專案，開發團隊對於所要開發的系統有豐富的經驗且需求定義明確；中間型為中等規模、中等複雜度的專案；內嵌型則是高複雜度、限制條件嚴格的專案。

基本模式是一個最粗略的估計，只考慮軟體規模，未考慮成本調整因子；中級模式除了考慮軟體規模外加15個成本調整因子，詳細模式更將成本依軟體開發階段來劃分，細分15個成本調整因子對各開發階段的影響，故在不同的軟體開發階段，可分別應用不同的等級，例如在軟體開發的初期，通常會先使用Basic等級來做度量，而當軟體開發到一定的程度，關於專案的資訊越來越多後，即可使用Intermediate和Advanced等級來作度量。在每種等級都將軟體分為三種複雜程度，分別稱為有機型(Organic)、半分離式型(Semidetached)及嵌入式型(Embedded)。無論是軟體等級或是軟體複雜度都有相同的表示法結構，即為

$$E = a(EDSI)^b * EAF$$

$E$ 表示*Effort*，即為專案所花費的人月，估算非註釋程式碼千行數(Estimated Delivered Source Instructions, EDSI)為 $EDSI$ 變數； $a$ 和 $b$ 參數，則隨著不同的軟體開發工作量等級則不同，兩者皆經由專案的歷史資料統計分析而來，都隨著複雜度的提昇而增加， $a$ 和 $b$ 參數可依需求參照表2-5。 $EAF$  (Effort adjustment factor)在基本模式是1，在中級模式和詳細模式中則為15個成本影響因子的總和。在Boehm的研究中發現，中級模式的預測較基本模式來的好，而詳細模式預測顯著性又不如中級模式[48]。中級模式的表示為：

$$E = a(EDSI)^b * \prod_{i=1}^{15} C_i$$

$C_i$  表示第*i*個成本影響因子(表2-6)，由上述式子可知，當 $C_i > 1$  時，表示此第*i*項成本影響因子會有提昇成本的趨勢，反之若 $C_i < 1$ ，則表示此因子有降低成本的趨勢。

表 2-5：COCOMO 公式係數表

	基本模式		中級模式		詳細模式	
	<i>a</i>	<i>b</i>	<i>a</i>	<i>b</i>	<i>a</i>	<i>b</i>
有機型	2.4	1.05	3.2	1.05	3.2	1.05
中間型	3.0	1.12	3.0	1.12	3.0	1.12
內嵌型	3.6	1.20	2.8	1.20	2.8	1.20

(資料來源：Boehm, Barry W.,1981[29])

表 2-6：COCOMO 資料集構面與影響因子說明

構面	影響因子( $C_i$ )	說明
產品屬性	產品要求之可靠度(RELY)	所開發之軟體系統專案對可靠度的要求程度
	資料庫之規模(DATA)	開發之軟體系統專案所需的資料庫規模
	產品之複雜度(CPLX)	軟體系統的複雜度
電腦屬性	執行時間之限制(TIME)	軟體系統在執行總時間中可獲得真正執行的時間比例
	磁碟容量限制(STOR)	軟體系統對電腦磁碟的容量限制程度
	硬體架構變動性(VIRT)	電腦硬體架構變動
	電腦回復時間(TURN)	軟體系統執行所需之總時間
人員屬性	分析師能力(ACAP)	系統分析師之能力
	類似開發經驗(AEXP)	專案人員對開發軟體之應用領域的熟悉程度
	程式設計師能力(PCAP)	程式設計師之程式撰寫能力
	開發環境熟悉度(VEXP)	專案人員對開發平台的熟悉程度
	程式語言經驗(LEXP)	程式設計師對於使用之程式語言的熟悉程度
專案屬性	現代程式設計技巧(MODP)	現代程式設計技巧使用的程度
	軟體工具使用(TOOL)	所使用之軟體工具的功能性
	發展時程(SCED)	軟體系統專案開發時程的緊迫程度

(資料來源：張祚佳,2005[15]；Boehm, Barry W.,1981[29])

COCOMO 是個相當完整的成本估計方式，優點是架構非常完整，其缺點是分類過於詳細，若無龐大的資料庫，則參數估計的誤差會很大[1][18]。

## 2.1.4 使用案例功能點分析法

使用案例(Use Case)的概念始於 1960 年的 Ivar Jacobson，一個使用案例(Use case)意味著『使用者使用到的功能』，也就是以顯而易見的方式描述一個系統的功能和特性要求，這類描述與使用程式語言及開發工具無關。亦為完成某特定目的，系統與外部使用者(Actor)的一系列的的基本活動。Actor 可以視為使用系統的人員，或是其他的外部系統。使用案例是由軟體巨擘 Jacobson 於 1992 年率先發表的[40]，對於近代物件導向的技術具有相當大的衝擊。並且由 Booch、Jacobson、Rumbaugh 所謂的物件導向(Object Oriented)三巨頭所聯合制訂並已交由 OMG 審議的 UML(Unified Modeling Language) 規格中，已納為主要的標準規格之重要一環。這裡列出幾位軟體巨擘們對使用案例的定義：

- ◆ 使用案例是一種敘事性的文件，用以描述某個參與者(Actor)使用系統完成某個事件的流程順序[40]。
- ◆ 使用案例就是一組情節(scenarios)，它們跟系統常見的使用目的互相關連，[36]。
- ◆ 使用案例是一個參與者(Actor) (通常是使用者，但或許可以為一個外部的實體，例如另一個外部系統)，在與內部系統的互動中，來達成一個特定目標的一系列行動[51]。
- ◆ The Unified Modeling Language User Guide一書中，則給予 Use Case 的定義：使用案例描述了一連串的循序，每一個循序表達了外部於系統的事物(參與者)與系統本身(及它的關鍵抽象)之間的互動。

使用案例功能點分析方法為傳統功能點分析和 Mark II 功能點分析的延伸，使用案例功能點分析法對於邏輯處理程序(Logical Transactions)的定義，在概念上可說和 MKII 功能點分析法非常相似，都是從使用者的觀點切入，並使用以使用者觀點的功能面來度量系統複雜度[34]。使用案例功能點(Use

Case Points Method)方法是根據一個系統的使用案例，來測量系統的大小[56]，其表示的單位為使用案例功能點，其主要計算流程說明如下。

◆ 主要計算流程為：

◆ 步驟一：計算使用者複雜度

將使用案例圖中各角色者分類(表 2-7)，加總其權重後得未調整角色權重(Unadjusted Actor Weights, UAW)(表 2-8)。

表 2-7：角色複雜度權重對應表

角色分類	分類說明	權重
簡單(Simple)	透過API與系統互動者	1
中等(Average)	透過網路協定如TCP/IP與系統互動者	2
複雜(Complex)	透過GUI介面或Web介面與系統互動者	3

表 2-8：角色複雜度權重計算表

角色分類	權重	角色數量	計算結果
簡單(Simple)	1	0	0
中等(Average)	2	0	0
複雜(Complex)	3	4	12
未調整角色權重(UAW)加總			12

◆ 步驟二：計算使用案例複雜度

將使用案例圖中各使用案例分類(表 2-9)，使用案例權重透過使用案例中的流程步驟或交易(Transaction)個數決定。透過將所有使用案例分類後，加總其權重後得未調整使用案例權重(Unadjusted Use Case Weights, UUCW) (表 2-10)。在 Karner 所提出的方法，被包含的使用案例或是延伸的使用案例，其權重皆不給予採計。

表 2-9：使用案例分類

使用案例種類	種類說明	權重
簡單(Simple)	簡單的使用者介面 資料庫實體個數為 1 者 案例流程步驟個數小於等於 3 者 實作類別個數小於 5 者	5
中等(Average)	較多的使用者介面設計 資料庫實體個數為 2 者 案例流程步驟個數介於 4 到 7 者 實作類別個數介於 5 到 10 者	10
複雜(Complex)	複雜的使用者介面與處理程序設計 資料庫實體個數大於 3 個以上者 案例流程步驟個數大於 7 者 實作類別個數大於 10 者	15

表 2-10：未調整使用案例權重計算表

使用案例種類	權重	使用案例數量	計算結果
簡單(Sample)	5	7	35
中等(Average)	10	13	130
複雜(Complex)	15	3	45
未調整使用案例權重(UUCW)加總			210

◆ 步驟三：產生未調整功能點

將 UAW 和 UUCW 加總(公式 2.1)，產生未調整功能點(Unadjusted Function Points, UFP)，計算範例如表 2-11 所示。接下來本文將於第四章應用此未調整功能點計算方法，但為了本文後續說明與解釋，之後本文將未調整功能點表示式由 UFP 改成  $UFP_j$ 。以  $UFP_j$  表示式來計算專案工作項目 j 的未調整功能點大小，其中  $j=1\sim N$ ，N 為 WBS 中專案工作項目總數。

$$UFP = UAW + UUCW \quad (2.1)$$

表 2-11：未調整功能點

複雜度	加總
未調整角色權重(UAW)	12
未調整使用案例權重(UUCW)	210
未調整功能點(UFP)	222

◆ 步驟四：設定TCF(技術複雜度因子)

設定技術複雜度調整因子(Technical Complexity Adjustment Factor, TCF)，共有 13 項因子需被考慮(表 2-13)，並由其影響的程度給予不同等級的權重(表 2-12)，TCF 為 T1~T13 各項因子其加權值乘以權重後加總而得，並計算可得 TCF(公式 2.2)，計算範例如表 2-14 所示。

$$TCF = C_1 + C_2 \sum_{i=1}^{13} W_i * F_i \quad (2.2)$$

其中  $C_1 = 0.6$ ,  $C_2 = .01$ ,

$W_i$  為技術複雜因子各項加權值,

$F_i$  為技術複雜度因子

表 2-12：複雜度影響性與權重對應表

影響等級	權重
無影響	0
偶爾影響	1
適度影響	2
平均影響	3
顯著影響	4
強烈影響	5

表 2-13：技術複雜度因子說明與加權值對應表

技術複雜度因子	說明	加權值
T1(分散資料處理)	此系統透過通訊協定或設備處理資料或控制資訊的程度	2
T2(效能目標)	此系統對於資料處理吞吐量需求上的程度	2
T3(終端使用者效率)	此系統在使用者界面的考量程度	1
T4(複雜處理)	此系統內部資料處理的複雜程度	1
T5(程式碼重複行數)	此系統的程式碼被其他系統使用的程度	1

T6(安裝簡易)	此系統對於安裝的便利程度	0.5
T7(使用簡易)	此系統在操作上得簡易程度	0.5
T8(可攜帶)	此系統在可攜帶上的程度	2
T9(容易修改)	此系統因應將來維護或修改的程度	1
T10(同步)	此系統在多個使用者同步操作上程度	1
T11(安全)	此系統在安全類別上的需求程度	1
T12(提供第三者使用)	此系統提供第三方參與者支援程度	1
T13(終端使用者訓練)	此系統提供使用者訓練的程度	1

表 2-14：技術複雜度因子計算表

技術複雜度因子	加權值	影響等級	技術複雜度因子計算
T1	2	1	2
T2	2	3	6
T3	1	3	3
T4	1	3	3
T5	1	0	0
T6	0.5	0	0
T7	0.5	5	2.5
T8	2	0	0
T9	1	3	3
T10	1	0	0
T11	1	0	0
T12	1	3	3
T13	1	0	0
技術複雜度因子加總			22.5

$$TCF = 0.6 + (0.01 * 22.5) = 0.825$$

◆ 步驟五：設定ECF(環境複雜度因子)

設定環境複雜度調整因子(Environmental Complexity Adjustment Factor, ECF)，共有 8 項因子需被考慮(表 2-15)，並由其影響的程度給予不同列影響權重(表 2-16)，並計算 E1~E8 其加權值乘以權重後加總，即可得 ECF，最後利用下列公式可得總 ECF 代表值(公式 2.3)，計算範例如表 2-16 所示。

$$ECF = C_1 + C_2 \sum_{i=1}^8 W_i * F_i \quad (2.3)$$

其中 $C_1 = 1.4$ ,  $C_2 = -0.03$ ,

$W_i$ 為環境複雜因子各項加權值,

$F_1$ 為環境複雜度因子

表 2-15：環境複雜度因子說明與加權值對應表

環境複雜度因子	說明	加權值
E1(流程熟悉)	開發人員熟悉軟體開發流程的程度	1.5
E2(使用應用程式經驗)	開發人員對於所使用的應用程式的熟悉度	0.5
E3(物件導向開發經驗)	開發人員對於物件導向的了解度	1
E4(分析能力)	分析能度的程度	0.5
E5(參與動機)	參與動機上的程度	1
E6(需求穩定)	需求穩定的程度	2
E7(參與的臨時工)	臨時工參與的程度	-1
E8(程式語言困難)	程式語言上手的程度	-1

表 2-16：環境複雜度因子計算表

環境複雜度因子	加權值	影響等級	環境複雜度因子計算
E1	1.5	5	7.5
E2	0.5	0	0
E3	1	5	5
E4	0.5	5	2.5
E5	1	5	5
E6	2	3	6
E7	-1	0	0
E8	-1	0	0
環境複雜度因子加總			26

$$ECF = 1.4 + (-0.03 * 26) = 0.62$$

◆ 步驟六：計算功能點

透過公式即可得到調整後功能點數 (Adjusted Function Points, AFP)，即可獲得以功能點為由公式 2.4 即可得到 AFP，此為以使用案例功能點所表示的軟體複雜度。

$$AFP = UFP * TCF * ECF \quad (2.4)$$

範例說明中未調整使用案例功能點(UFP)為 222；技術複雜度因子(TCF)為 0.825；環境複雜度因子(ECF)為 0.62，最後使用案例功能點(AFP)如下：

$$AFP = 222 * 0.825 * 0.62 = 113.55 \text{ 個功能點或稱為 } 114 \text{ 個功能點}$$

### 2.1.5 專家判斷法

專家判斷法是目前軟體產業界最常使用的軟體專案預估方法，當早期需求不明確或是配合新技術或新方法而無歷史資料可供參考時，就只有以專案經理本身的經驗為基礎，時而輔以類比式(Analogy)的預估方法，即以過去開發過的軟體專案資料作相似度的比對，進而挑選出最相似的歷史專案，來得到主觀的專案預估值。較常用到模式及方法有[15][17][29][30]：

- TRW Wolverton模式：它是透過專案經理依據過去的開發經驗來主觀判斷軟體專案中每個模組的型態、困難度，以選擇影響成本因素的『軟體成本矩陣』(Software Cost Matrix)中之值，再加上預估軟體的程式碼行數一同輸入到計算成本的公式中來算出軟體專案開發所需的成本[57]。
- 德爾非技術 (Delphi Technique)：經由一群此領域的專家，在不互相討論的情形下，自行提供開發專案中的各項預估值及其詳細原因，經由專人回收並統計後，再公佈結果及相關問題給所有的參與專家。一直重複以上的步驟直到大家在預估的項目上達成一定程度的共識。
- 計畫評核術 (Program Evaluation and Review Technique, PERT)：可被用來預估專案中的個別項目，透過加總這些項目的預估值來獲得開發軟體系統的整體預估值。它由專家判斷的三個可能的程式碼大小， $O$ 是最樂觀的預估值、 $P$ 是悲觀的預估值、 $R$ 是最可能的預估值。其預估結果 $\mu$ 可被計算如下：

$$\mu = (O + 4R + P) / 6$$

專家法亦即在估計專案的軟體工作量時，利用對此類型專案有經驗的專家來估計軟體工作量。換句話說，專家法所採用的是一個成熟開發者的個人

經驗來估計軟體工作量。因此在專家給建議之前，專案的管理者必須盡可能對該專案的特徵做描述，然後專家就會根據他過去的經驗針對該專案的描述產生軟體工作量預估值。一般而言，專家也會使用一些工具、模式或是方法來輔助以產生預估值，但專案管理者對這些支援的技術並不需要瞭解，管理者所重視的是專家的品質及經驗的廣度。

### 2.1.6 類比法

類比法亦即為預估者需要將預估的專案與過去的專案進行比較，以從中分析某些相似處和差異處，以推導出預估值。因此當找到最相似的過去專案後，可利用差異處來調整該筆相似專案的過去實際開發工作量，以成為新專案的預估工作量值。此外，類比技術的分析必須文件化以使得往後的預估值也能受惠。因此往後當預估值一直不如預期的精確時，也可做為發現原因的線索依據。

使用類比法的時機，大多是用較多歷史資料可驗證及參考的情況，以提高軟體成本預估的準確，而且並不只是預估一次，通常在軟體開發生命週期的規劃階段就進行整體估計，當進行後設計階段則需再進行修正。此方法的優點是類比法的流程符合人類推理的行為，但缺點則是可能受主觀影響，或有偏誤可能看起來相似的專案，可能有很大的差異。

## 2.2 專案風險

探討專案風險之前，本文先針對學者於「專案」一詞定義進行探討與說明，接續再探討過去相關學者對「專案風險」之定義。由於專案管理一辭已衍生許久，故專案的定義隨著許多研究學者的不同而有些許的差異。一般的專案定義如下(表 2-17)：

表 2-17：專案定義

提出學者/單位	定義
Newman(1972)	專案是指一組完整而特定目標之活動，具有明確的起迄時點與成本限制，內容牽涉到各項技術，具有相當的複雜性，而又不重覆。
Reeser (1973)	牽涉各種技術為達成任務所必要的努力，需要在限定時間內完成者。
Cleland and King(1983)	將專案描述為在特定的時間預算下，為了達成特定的目標所從事的活動，這些活動通常是跨越原來組織中的階層關係，且與原來組織中例行性的活動有所不同。
John M. Stewart (1984)	對專案的構成列出以下條件： (一) 有特定之目的； (二) 其任務為現行組織所不熟悉； (三) 各項活務的相互依存關係甚為複雜； (四) 盈餘或虧損影響甚大； (五) 任務為臨時性質。
Meredith and Mantel (1989)	為一特殊及期限內完成之任務，具技術複雜性，賴不斷協調來控制進度、過程、成本及工作績效。
許榮榕(1995)	認為專案是一具有特定目的 (Goal)，尤其是指非例行性 (non-routine) 及不重複 (non-repetitive) 的工作。此工作需比平常的作業需要更多的人力、財力、物力、時間及技術等資源；而專案與其他作業不同之特性更在於以往未曾做過，而現在必須去進行之事務。
J. M. Juran(1998)	專案是為了解決問題所排定的進度表。
王慶富(1999)	認為專案是指一組完整而有特定目標之非重複性活動，皆具有明確的起迄時間及成本限制，這種活動通常於臨時性組織中完成任務，其內容牽涉各項技術，具複雜性，但其成果難以預測，具冒險性。
羅應浮(2000)	專案具有以下特性： (一) 特定的目標(Goal)； (二) 明確的時程(Schedule)； (三) 限制的成本(Cost)； (四) 複雜性(Complexity)； (五) 非常設(Impermanence)； (六) 無經驗(Unfamiliarity)； (七) 不確定性(Uncertainty)。
美國專案管理協會 (2004)	任何擁有特定起點並於特定時間內完成工作的活動，並認為在實務上的專案施行通常都是具有資源限制的。

(資料來源：修改自藍元志,2003[2][10][23][25][47][50])

綜合以上各學者對於專案之定義，歸納出以下特性：

- (一) 有明確特定之目標；
- (二) 有一定的時間限制；
- (三) 所能運用的資源有限；
- (四) 是暫時性的具非例行性及不重複性。

了解專案定義後，後續探討何謂「專案風險」，雖然我們每天都面對不同的風險，但至今風險並沒有一致的定義，一般來說風險的定義可分為事故發生的不確定性(Risk is Uncertainty)與事故發生遭受損失的機會(Risk is the Chance of Loss)[46]。根據韋伯(Webster)字典的定義，風險為 Possibility of Loss or Injury，由此可知，風險包含了機率(Probability)與損失(Loss)兩項變數，而後續許多學者也根據此來對風險加以定義。

在軟體風險方面，其定義與一般風險並沒有太大的差異。首先就風險(risk)與不確定性(uncertainty)做區分；所謂的不確定性指的是不知道事件發生的機率及事件發生後會帶來的影響，而風險則剛好相反，風險指的是已經切確知道事件會發生之機率以及其所帶來之影響，故不確定性指的是一個較未知的狀態，而風險則是一個已經知道的狀態，所以在管理上，風險較能掌握，而不確定性由於是未知的，所以較難管理。下圖，可表示風險與不確定性之區隔。

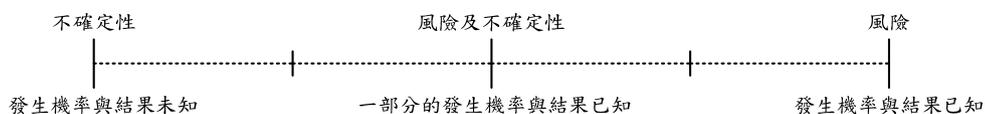


圖 2-2：風險與不確定性區分圖

(資料來源: Kevin J. Bernhardt, “Risk, Uncertainty, and Management Implications”)

接下來便是過去對專案風險意義探討之文獻(表 2-18)：

表 2-18：專案風險的定義

提出學者/單位	定義
Matthew(1983)	一般來說風險的定義可分為事故發生的不確定性(Risk is Uncertainty)與事故發生遭受損失的機會(Risk is the Chance of Loss)。
陳建名(1995)	將軟體專案風險定義為專案進行中可能遭遇的事件，此事件將使專案不順利或產生一定程度的衝擊，並將其分為發生的機率強度與一旦發生所造成的損失兩部分。
許榮榕(1995)	將專案執行之風險(risk)或不確定性(uncertainty)，定義為專案時間、成本及成果達成之機率可能性。
許光華、何文榮(1998)	將風險定義為不希望發生事件發生的機率與這些事件發生造成後造成影響的程度。
葉蘇蓉(1999)	在軟體專案發展過程中，對於可能造成專案時程的延誤、效能不符合需求及超出預算之因素的預測。由此可知，專案風險的定義源自於風險，多半是指專案失敗發生的機率及所造成的影響程度。
英國專案管理協會(1997)	對專案風險的定義為：一個不確定的事件或是一些事件的集合，當這些事件發生時，將會對達成專案目標造成影響。
林大舜(2003)	將風險視為未來不確定性的事件，該事件有可能影響組織目標的達成，包括策略、作業、財務或其他一致性的目標。
Alan Webb(2003)	則將專案風險定義為與預期後果相反的結果，可能是財務的或實體的，導因於在專案過程中所做的決策，或是源於專案存在的環境。
美國專案管理協會(2004)	將專案風險定義如下：風險是一個不確定的事件或狀態，一旦其發生，會對專案目標帶來正面或負面的影響。

(資料來源：修改自張銘仁,2004[3][9][11][12][16])

依照上述研究學者的定義，可以將專案風險定義為：專案受到內部因素或外部環境的干擾，導致成本與時程上的增加，而影響專案的品質或是造成專案開發上困難的因素。

## 2.3 工作分解結構

工作分解結構(Work Breakdown Structrue，WBS)最早是由美國國防部提出[33]，是一種用來分析專案內活動的方法，可以協助專案管理者建立明確的

專案目標。WBS 最初的概念為：

- 是一個以產品導向的樹狀結構，結構內以硬體、軟體、服務、資料等組成。
- 可向下延伸到任何階層。
- 可用以表示和定義最終產品。樹狀結構中包含完成產品所需要的各項工作，用以產生最終產品。

WBS 是一種從總體到細節定義專案最終預期結果的規劃方法，將軟體專案由上而下分成許多專案活動，用以解決專案，完成專案目標。WBS 將一個專案分解成許多較具體的專案活動，專案管理者能更進一步的在此基礎上進行資源的分配、進度計劃並估計專案的成本。一份專案的 WBS（圖 2-3）包含了下列資訊；專案工作項目、專案時程、工作項目使用到的資源與所屬工作人員等。

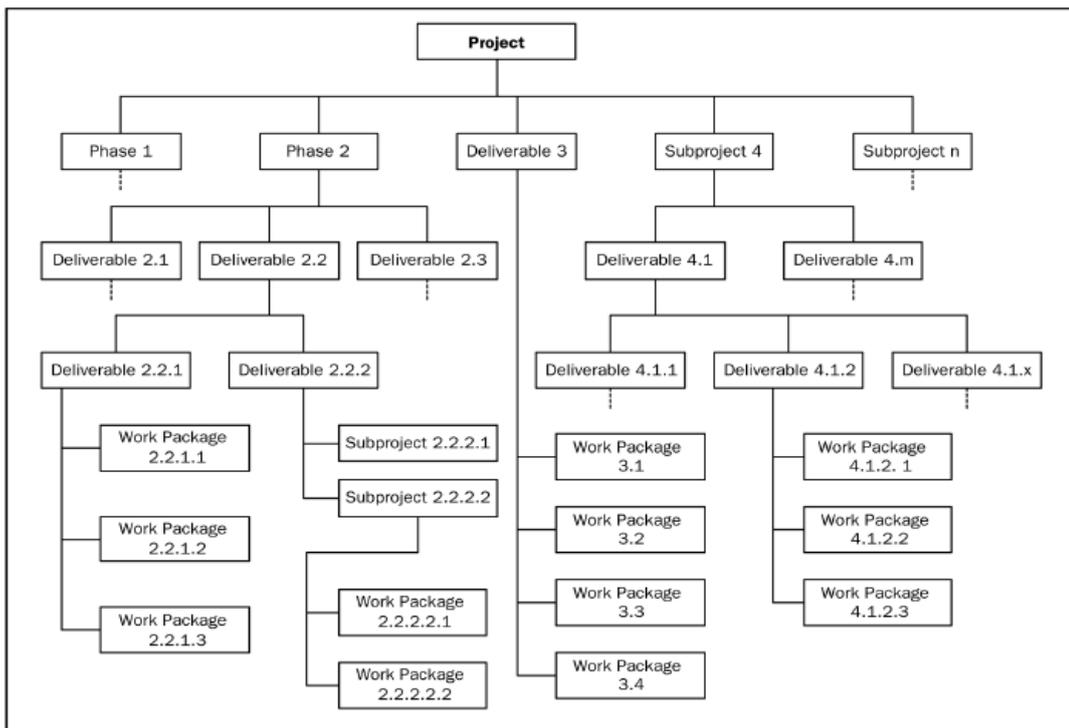


圖 2-3：WBS 範例

## 2.4 變異數分析(ANOVA)

變異數分析(Analysis-of-Variance, ANOVA)為統計學家Fisher首創，最初應用於農業實驗結果之考驗，現則被廣泛推廣適用於各種實驗領域，因此變異數分析法已經成為了實驗設計的重要工具[19]，其理論基礎是基於各類觀察體(observations)常因受某些因素之影響，以致其特性常發生不同之差異。而此種差異在統計學中則以變異(離差平方和, Sum-of-squares, SS)表示之；通常可分為兩方面來說明變異產生的原因[19]：

- 已知原因之變異(自變數引起)。
- 未知原因之變異(抽樣誤差或實驗誤差)。

變異數分析是用來檢定三個或三個以上母群平均數的差異顯著性，若只有一個自變數的變異數分析則稱為單因子變異數分析(One Way ANOVA)，即比較單一種自變項的不同處理方式對某依變項的影響。其分析相關內涵如下表2-19所示。

表 2-19：ANOVA 表

變異來源	平方和(ss)	自由度(ds)	平均平方和(ms)
迴歸	$SSR = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$	$k-1$	$MSR = \frac{SSR}{k-1}$
殘差	$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$	$k(n-1)$	$MSE = \frac{SSE}{k(n-1)}$
總和	$SS_{yy} = \sum_{i=1}^n (Y_i - \bar{Y})^2$	$N-1$	

(資料來源：胡衣臨,2005[6])

其中k為組數，N為總樣本數，此外，檢定統計量(F\*)， $F^* = MSR/MSE$ 。一般定義當 $\alpha$ 值小於0.05時，表示有達顯著[19][20]。若想瞭解兩個自變數A、B間是否有交互作用存在，即A因子的不同水準是否隨著B因子水準不同而有不同的效果，即採用兩因子變異數分析。若交互作用達顯著，則進一步分析其單純主要效果。即A因子在B因子的哪一個水準有顯著效果，以及B因子在A因子的哪一個水準有顯著效果。若單純主要效果顯著，則可比較水準間的差

異。若交互作用未達顯著，則進行自變數的主要效果考驗。若想瞭解三個自變數間是否有交互作用存在，即採用三因子變異數分析。變異數分析根據所使用的樣本可分為二種設計方式：

- 獨立樣本設計

即所有受試者被隨機分派到K個不同的組別，分別接受這一個自變數的K個實驗處理中的一個實驗處理，各受試者之間彼此是毫無關係的。

- 相依樣本設計：(下列三種情況均屬相依樣本設計)

- ◆ 重複量數：以同一組受試者重複接受K種處理，以觀察其表現的差異。
- ◆ 配對組法：選擇K組受試者在某一與依變數有關的特質方面完全相同，控制此配對的條件完全相同而比較K組受試者在依變數上的表現。
- ◆ 同胎法：K個先天條件完全相同的受試者接受不同的處理。

## 2.5 實證軟體工程

軟體專案與人類日常生活行息息相關，舉凡交通運輸、醫藥、超級市場和股市等處處可見。而軟體系統運作結果正確與否不僅浪費巨大的公共財，並且嚴重危及人們生命財產安全，所引發的影響也非常的深遠。為避免上述狀況的發生，本研究除了於軟體專案開發過程中，透過迴歸分析法並採用業界實際個案對這個評估方式進行驗證，有助於提升系統的可靠及可用度，此實驗驗證程序係屬實證軟體工程(Evidence-based Software Engineering, EBSE)應用的一種，藉由與實證軟體工程機制的相輔相成，有助於決策者兼顧經驗法則和人類價值觀，制定最佳決策[41]。

根據學者Sackett指出關於實證研究的文章發表數量。自1992年的1篇成長到1998年大約有1000篇文章，短短的數年間竟有如此快速的成長，這說明了實證研究越來越顯得重要[52]。實證軟體工程的主要目的是從研究領域提供目前最佳的驗證方法。因此，本文參照Kitchenham等學者所提出關於實證軟體工程的五個步驟[43]，將本研究實施實證軟體工程分成下述五個主要步驟：

- 步驟一：確定問題定義

究竟Randy K. Smith估算模型與本研究數學評估模式，在軟體規模估算上的準確度為何？假設本研究數學評估模式估算準確度不亞於Randy K. Smith估算模型估算準確度。

- 步驟二：蒐集最佳的證據

以現行組織所進行過的專案歷史資料為主，依照本研究所應用的評估模式蒐集相關實證資料。所以根據這兩種評估模式所需要的資料，找尋組織內專案的歷史資料，作為最佳的證據。

- 步驟三：嚴謹的實證流程

Randy K. Smith 估算模型於 2001 年 IEEE TRANSACTION ON SOFTWARE ENGINEERING 期刊上發表，指出影響專案的潛在因素，加上 Randy K. Smith 更進一步的分析近四百個專案資料後，證明他所提出的估算模型在準確度上有良好的表現，其模型的相對誤差振幅(Magnitude of Relative Error, MRE)表現上有30.4%的專案落在估算模型的20%界限內，加上影響專案的潛在因素定義上相當嚴謹，不會有重複計值之現象。因此，除了規範各自實證流程外，採用Randy K. Smith估算模型分析結果當作比較驗證基準。

- 步驟四：實踐評估實證

將所蒐集的實際個案歷史資料，分別按照這兩種估算模式實證流程，進行估算。本研究將實作評估分成兩個階段，先是訓練階段，再將訓練階段產生的最佳估算參數代入檢測階段和檢測資料進行專案規模估算。

- 步驟五：實證結果與討論

最後探討上述步驟一至步驟四中，所發現的問題及比較實證結果，提供回饋修正估算模式及調整估算參數參考。

### 第三章、專案風險分析方法

本研究的專案風險分析方法透過研究個案內部的專案流程配合文獻探討及採用Linda Wallace 在2004 提出的風險構面與風險因子[44]以專案歷史資料為統計主體加以分析而成。在分析工具方面，本研究採用Boehm提出的風險分析架構[28]，依照風險識別、風險評估與風險分級三個步驟的風險評估機制，導引出可供專案管理參考的風險評估表。本研究利用這個風險評估機制取得每一個專案工作項目的風險權重值。然後將專案工作項目識別出的各項風險因子，利用Boehm 所提出的風險損失計算公式乘上風險權重值的總和，得到工作項目的風險影響時程，即為風險發生時必須付出的額外花費時間，其將用於本文後續預估專案規模之估算因子之一。本章節將介紹專案風險分析的架構與軟體專案風險中常見的風險構面與風險因子，以及如何透過風險識別與評估的流程得到專案規模估算因子：工作項目的風險影響時程。

#### 3.1 軟體風險因子

自從 McFarlan 在 1981 年提出利用查核表(checklist)來找出軟體專案的風險規模後，風險管理的課題才開始受到軟體產業重視，1991 年，Boehm 針對自身在 TRW 公司的開發經驗，提出了 10 個最重要的風險因子並受到廣泛討論後，識別軟體風險因子的課題開始受到學者們的關注。

在 1993 年，SEI 針對軟體開發的風險因子定義了一個分類架構，該架構將軟體風險分成三個構面，分別是產品工程、開發環境與專案限制。此外，1998 年 Keil 所提出的軟體風險分類架構則是將 53 個軟體風險因子分成四個構面，分別為客戶、需求、環境、執行。近年來，Tom DeMarco 在『與熊共舞』一書[22]中則指出先天的時程錯誤、需求膨脹、人力流失、規格崩潰與低生產力是五個普遍存在於軟體專案開發期間的核心風險。表 3-1 彙整了軟體風險因子的相關文獻資訊。

表 3-1：軟體風險因子相關文獻

提出學者/單位	軟體風險構面	軟體風險因子
McFarlan(1981)	無	53 個
Boehm (1991)	無	10 個
Marvin. J Carr(1993)	3 項	64 個
Mark Keil(1998)	4 項	11 個
James J. Jiang(2001)	11 項	45 個
Tom DeMarco(2003)	無	5 個
Linda Wallace(2004)	6 項	44 個

(資料來源：張家豪、黃世禎、韓文銘, 2006[14])

最後，在 2004 年 Linda Wallace 針對 507 位具有 PMP 證照的專案管理者的研究調查顯示，軟體風險因子可以分成六個構面，共 44 個軟體風險因子。本文則是引用 Linda Wallace 的研究調查所蒐集的軟體風險因子，如表 3-2。

表 3-2：Linda Wallace 提出的風險構面與風險因子

風險構面	描述	風險因子
團隊風險	專案團隊中的成員會增加最後專案成效的不確定性，例如團隊成員的流動、溝通、合作與激勵，以及成員的知識是否充足等議題。	<ul style="list-style-type: none"> <li>● 團隊成員間時常發生衝突</li> <li>● 專案團隊時常流動</li> <li>● 團隊成員不熟悉那些專案工作能夠藉由工具被自動化處理</li> <li>● 團隊成員缺乏專案所需技能</li> <li>● 團隊成員未獲得適當的訓練</li> <li>● 團隊成員對專案缺乏承諾</li> <li>● 團隊成員經驗不足</li> </ul>
組織環境風險	在專案風險中，組織環境風險被視為第二重要的風險構面，例如組織的政策、整個組織環境的穩定性以及組織對專案的支持，都會影響整個專案的最後的績效表現。	<ul style="list-style-type: none"> <li>● 缺乏管理者的支持</li> <li>● 專案執行期間組織管理方式的改變</li> <li>● 專案執行期間組織重新改組</li> <li>● 組織環境的不確定性</li> <li>● 組織政策對於專案造成負面衝擊</li> <li>● 專案間優先權的改變造成專案資源的轉移與流失</li> </ul>
需求風險	在系統發展的專案中，除了經常性需要的改變是與需要有關的問題之外，不正確、不適當以及不清	<ul style="list-style-type: none"> <li>● 不正確的需求</li> <li>● 使用者缺乏對系統能力和極限的理解</li> <li>● 未定義專案成功的標準</li> <li>● 需求間相互衝突</li> <li>● 不明確的需求</li> </ul>

	楚的需要也可能造成軟體發展的問題，提高專案風險。	<ul style="list-style-type: none"> <li>● 系統的輸出與輸入定義不易</li> <li>● 系統需求沒有充分地被發現</li> <li>● 需求持續的變動</li> </ul>
規劃與控制風險	在軟體專案開發的過程中，不適當的規劃和控制會造成不切合實際的預算與時程，也會使專案管理者不容易衡量此一專案是否有在預定階段達到想要達到的目標。在沒有辦法為專案做時程的估計下，管理者會無法得知哪些資源可以為這專案的發展做承諾。	<ul style="list-style-type: none"> <li>● 專案里程碑沒有被明確定義</li> <li>● 專案進度沒有被嚴密監控</li> <li>● 缺乏有效的專案管理方法</li> <li>● 專案管理者的經驗不足</li> <li>● 不好的專案規劃</li> <li>● 專案領導欠缺人性化</li> <li>● 專案溝通不良</li> <li>● 必要的專案資源未被有效的預估</li> <li>● 不切實際的時程估計與安排</li> </ul>
使用者風險	在系統發展期間缺乏使用者的參與是造成風險的一個原因。若使用者對於這新的系統抱持的態度是不友善的，將有可能不配合系統的發展，而造成專案發展失敗率的提升。	<ul style="list-style-type: none"> <li>● 與使用者缺乏合作</li> <li>● 使用者抗拒改變</li> <li>● 未取得使用者對於專案的承諾</li> <li>● 缺乏使用者參與</li> <li>● 使用者間發生衝突</li> <li>● 使用者對於專案的消極態度</li> </ul>
複雜度風險	一個軟體專案複雜度就是此專案在進行時的困難度。可以藉由以下屬性來判斷複雜度的高低，例如當使用這個新軟體技術時處理的過程是否複雜，是否需要很多連結(link)來連接原有已存在的系統(existing system)以及外部實體(external entity)。	<ul style="list-style-type: none"> <li>● 專案技術未被用在被設定的用途上</li> <li>● 與現行系統間存在大量的連結</li> <li>● 高度的技術複雜度</li> <li>● 組織內曾經執行過的最大型專案中是否有試行過此類技術</li> <li>● 專案是否使用新的技術</li> <li>● 許多外部供應商是否參與專案開發</li> <li>● 不成熟的技術</li> <li>● 高度複雜度的工作是否被自動化</li> </ul>

## 3.2 風險管理

Boehm認為有效的風險管理必須透過事先一連串的程序：風險辨識、風險評估、風險分級來進行風險分析；另外，應於專案管理計畫中制定風險計畫，研究風險解決辦法以減緩風險發生時的影響，進行風險控制活動，並且不斷地進行風險監督一直持續至專案結束，並提出了如圖3-1的風險管理架構。本研究將以風險管理架構中屬於風險分析的三大部分：風險辨識、風險評估、風險分級來探討，如何利用事前的風險分析來幫助專案規模的預估，所以風險控制並非本文討論範疇。

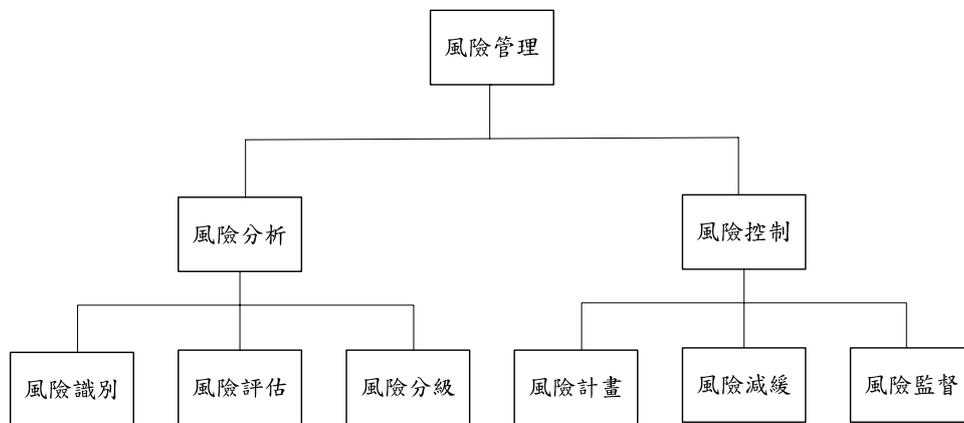


圖 3-1：Boehm 的風險管理架構[28]

## 3.3 專案風險識別

一般常使用的風險識別有幾種方法[4]：

- 分解法：將專案風險分成幾個較小、可處理的部分，然後將每部分的風險詳細地加以分析及判斷，以找出最可能得風險事項。
- 個案參考：對大部分的專案而言，風險的識別是基於歷史的經驗而來，所以參考以往的經驗找出相關的風險事項就是一個很好的方法。
- 腦力激盪：由專案小組以開會的方式相互激盪、刺激靈感，這個方法也常被使用。
- 問卷表：經由相關人員腦力激盪後，利用數個問題來詢問專案小組或是當事人，並讓其作答，藉此可判斷該風險是否存在。

- 理論：一些學者所提出的理論，如 80/20 法則或權重方法，如 AHP、Delphi 分析法。
- 經驗：老祖宗的經驗傳承。

本研究的風險識別採用分解法，利用 WBS 將專案分成幾個較小、可處理的專案工作項目，每個專案工作項目再利用專案工作細項分類表(表 3-3)依照工作屬性細分為多個工作細項，如圖 3-2 所示。然後將每部分利用 Linda Wallace 風險因子表(表 3-2)加以詳細分析及判斷，以找出最可能的風險事項，協助專案小組進行風險識別工作。將專案工作項目細分為多個工作細項，有助於後續進行風險識別與評估作業。以下將以範例說明風險識別作業。

例如，有一個專案 WBS 包含專案執行管控、部落格建置、管理系統擴充、介面開發與整合、資料庫建置等專案工作項目，依據主要工作項目的工作內容及工作屬性細項分類，依照表 3-3 的工作細項與細項內容舉例，專案管理者則可再將專案工作項目分解成多個主要工作項目，如下所示：

- 專案執行管控：可細分為需求管理、專案管理、規劃分析設計、資料處理、資訊管理等五項主要工作項目。
- 部落格建置：可細分為開發製作、規劃分析設計、資料處理、需求管理等四項主要工作項目。
- 管理系統擴充：可細分為開發製作、需求管理、規劃分析設計等三項主要工作項目。
- 介面開發與整合：可細分為開發製作、需求管理、規劃分析設計等三項主要工作項目。
- 資料庫建置：可細分為開發製作、需求管理、規劃分析設計、資料處理等四項主要工作項目。

這些專案工作項目中主要工作項目再將預估開始與完成時間、負責人員設定到 WBS 內容中，範例如圖 3-2 所示。

表 3-3：專案工作細項分類表

工作細項	舉例	工作細項	舉例	工作細項	舉例
專案管理	範疇/整合管理 成本管理 時程管理 溝通管理 人力資源管理 監控度量 品保稽核	教育訓練	內部教育訓練(部門教育訓練、中心 seminar) 受訓 系統教育訓練	資訊管理	建構管理=CM 軟硬體建置 網路維護 系統測試 電腦維修
需求管理	訪談 分析	規劃分析設計	規劃(工作項目內容規劃製作/系統/文宣/電子報內容) 系統分析 系統設計 架構設計 問卷分析 空間分析 各項評估分析事務	資料處理	會議資料(報告書、簡報) 圖資處理 網頁資料處理 資料庫處理 資料收集(系統/電子報/網站) 問卷整理及統計 海報設計與製作
客服	客訴電話處理	開發製作	程式開發 系統維護 介面開發(UI) 電子報製作	人事總務	人才招聘 庶務管理
支援事項	支援教師 職工福利	專案監審	資料檢驗 系統檢驗	財會事項	會計 核銷 採購

(資料來源：本研究)

	WBS	任務名稱	工期	預估開始時間	預估完成時間	負責人員
1	1	1 專案執行控管	120 工作日?	96/10/1	97/3/20	
2	1.1	1.1 需求管理	60 工作日?	96/10/1	96/12/19	Cathychen,Gary
3	1.1.1	1.1.1 訪談	25 工作日	96/10/1	96/11/2	Gary
4	1.1.2	1.1.2 分析	56 工作日?	96/10/5	96/12/19	Cathychen
5	1.2	1.2 專案管理	120 工作日?	96/10/1	97/3/20	Cathychen
13	1.3	1.3 規劃分析設計	30 工作日?	96/10/1	96/11/9	Gary,David
17	1.4	1.4 資料處理	10 工作日?	96/10/1	96/10/12	Claire,Cryil
20	1.5	1.5 資訊管理	74 工作日?	96/11/1	97/2/15	Jeff,Ken
23	2	2 部落格建置	30 工作日?	96/10/1	96/11/9	
24	2.1	2.1 開發製作	30 工作日?	96/10/1	96/11/9	Ken,Yukon
27	2.2	2.2 規劃分析設計	12 工作日?	96/10/1	96/10/16	David
30	2.3	2.3 資料處理	6 工作日	96/10/1	96/10/8	Claire
32	2.4	2.4 需求管理	8 工作日?	96/10/1	96/10/10	Cathychen,Gary
35	3	3 管理系統擴充	68 工作日?	96/10/1	96/12/31	
36	3.1	3.1 開發製作	68 工作日?	96/10/1	96/12/31	Ken,Jane,Yukon
40	3.2	3.2 需求管理	33 工作日?	96/10/1	96/11/14	David,Gary
43	3.3	3.3 規劃分析設計	38 工作日?	96/10/1	96/11/21	Gary,Jeff
46	4	4 介面開發與整合	65 工作日?	96/10/1	96/12/26	
47	4.1	4.1 開發製作	65 工作日?	96/10/1	96/12/26	Ken,Yukon,Jane
51	4.2	4.2 需求管理	50 工作日?	96/10/1	96/12/6	Gary,David
54	4.3	4.3 規劃分析設計	55 工作日?	96/10/1	96/12/13	David,Jeff
57	5	5 資料庫建置	116 工作日?	96/10/1	97/3/14	
58	5.1	5.1 開發製作	116 工作日?	96/10/1	97/3/14	Ken,David
61	5.2	5.2 需求管理	76 工作日?	96/10/1	97/1/11	David,Gary
64	5.3	5.3 規劃分析設計	60 工作日?	96/10/1	96/12/19	David,Jeff
67	5.4	5.4 資料處理	66 工作日?	96/10/1	96/12/27	Jeff,Cryil

圖 3-2：依據專案工作細項分類之 WBS 內容說明

### 3.4 專案風險評估

因為風險是一件無法預期發生的事件與相關事件後果的整合，每個專案均有無法達成目標的風險，要評估專案風險，第一步要先分解專案成細項活動，建立專案的 WBS，再依據軟體生命週期劃分的階段，鑑定出各 WBS 進行的深度與可能的風險[4]。專案風險評估所需研究資料之採取樣本為本文研究個案 2005.01~2008.10 間所簽約並執行完成之軟體專案，共 430 件。為了使評估更為準確，表 3-4、軟體專案風險影響權重表中「專案大小」、「專案所用技能」、「專案開發結構化程度」採用 Lynda M. Applegate 的專案開發風險評估架構[45]，權重部分則採取邀請該本研究個案的九位專案管理者採取團體會商的方式所制定；WBS 專案工作項目風險評估方法，則是將依據專案工作細項分類表(表 3-3)分類後的工作細項，再進行風險因子識別並計算其影響程度及風險影響時程，如表 3-5 所示。最後再彙整專案所有工作細項得到風險評估表，如表 3-6 所示。

表 3-4：軟體專案風險影響權重表

技能分類	經費單位：千元		低結構化		高結構化	
	技能熟識	大專案 (經費>2,000)		中低風險	權重: 40	中低風險
中專案 (經費介於 500~2,000)			中度風險	權重: 50	低風險	權重: 25
小專案 (經費<500)			極低風險	權重: 10	極低風險	權重: 10
技能生疏	大專案 (經費>2,000)		極高風險	權重: 100	中度風險	權重: 50
	中專案 (經費介於 500~2,000)		高風險	權重: 75	中低風險	權重: 40
	小專案 (經費<500)		中度風險	權重: 50	低風險	權重: 25

例如，有一個要進行風險評估工作細項：部落格建置-規劃分析設計，規劃共需 12 個人天數，專案管理者藉由軟體專案風險影響權重表(表 3-4)與軟體風險因子表(表 3-2)中風險識別出三個風險因子(表 3-5)，則此工作細項的風險發生時，將使專案時程增加 $((7.5+45+45)/100)*12=11.7$  人天數。此部分之風險識別與影響時程計算方式，後續將於本研究預估模型中之風險影響時程估算因子(Risk Effort, *RISK*)所應用。範例計算如表 3-5 所示，表 3-5 專案工作細項風險識別表中的風險構面與風險因子，本應為六個構面及 44 個風險因子，本文為方便觀看與說明，所以將未識別出的風險構面與風險因子，省略表示。

表 3-5：專案工作細項風險識別表

風險評估工作細項		部落格建置-規劃分析設計			
風險構面	風險來源 (風險因子)	影響 權重 (Wt <sub>j</sub> )	風險發生 機率 (Prob <sub>j</sub> )	風險 影響性 (PRISK <sub>j</sub> )	風險影 響時程 (RISK <sub>j</sub> )
團隊 風險	<input checked="" type="checkbox"/> 團隊成員經驗不足	75	10%	0.075	12*0.075=0.9
	<input type="checkbox"/> 團隊成員對專案缺乏承諾				
需求 風險	<input type="checkbox"/> 需求持續的變動				
	<input checked="" type="checkbox"/> 不明確的需求	75	60%	0.45	12*0.45=5.4
複雜 度風 險	<input checked="" type="checkbox"/> 高度的技術複雜度	50	90%	0.45	12*0.45=5.4
	<input type="checkbox"/> 不成熟的技術				
	<input type="checkbox"/> 專案是否使用新的技術				
風險影響時程					11.7(天)

### 3.5 專案風險分級

依照風險評估表(表 3-6)，藉由規劃工時與影響程度得到嚴重程度，並以此嚴重程度為索引(數字越大越嚴重)，可以很輕易排序出風險分級。例如，表 3-5 中的風險評估工作細項：部落格建置-規劃分析設計，專案管理者經由風險識別與風險評估步驟計算得到當此工作細項風險發生時，將使專案時程增加 $((7.5+45+45)/100)*12=11.7$  人天數，造成專案時程增加的影響，本文稱之為風險影響時程。當專案工作項目所有工作細項的影響時程都計算出後，就能依據影響時程的長短進行排序，得到風險評估表中的嚴重程度，如表 3-6 所示。若因其內部之組織結構、人力分配、或管理方式等各項因素，導致專案資源衝突時，專案管理者可依據其嚴重程度排定優先處理的專案工作項目，專案風險分級更有助於組織內執行各項專案與專案資源分配。

表 3-6：專案風險評估表

工作項目	工作細項	規劃天數	影響程度	風險影響時程(天)	嚴重程度
專案執行控管	需求管理	60	0.075	4.5	13
	專案管理	120	0.01	1.2	7
	規劃分析設計	30	0.065	1.95	9
	資料處理	10	0.05	0.5	5
	資訊管理	7	0.015	0.105	1
部落格建置	開發製作	30	1.05	31.5	19
	規劃分析設計	12	0.975	11.7	16
	資料處理	6	0.55	3.3	11
	需求管理	8	0.43	3.44	12
管理系統擴充	開發製作	45	0.315	14.175	17
	需求管理	10	0.035	0.35	3
	規劃分析設計	15	0.06	0.9	6
介面開發、整合	開發製作	20	0.325	6.5	14
	需求管理	5	0.075	0.375	4
	規劃分析設計	10	0.025	0.25	2
資料庫建置	開發製作	60	0.435	26.1	18
	需求管理	20	0.085	1.7	8
	規劃分析設計	15	0.15	2.25	10
	資料處理	10	0.95	9.5	15

## 第四章、專案規模預估模型設計

本章節將介紹本研究設計之專案規模預估模型以及如何由WBS取得專案規模預估因子的分析流程與估算方式。首先介紹Randy K. Smith所提出用於估算專案工作人力之工作指派模型，就未調整之功能點、人員平均年資、專案參與人數、工作項時程緊密度、人員並行程度、時間離散程度等構面進行專案工作項目需投入人天數預估。Randy K. Smith的專案工作人力之工作指派模型，其模型樣式如下[49]：

$$\ln E_j = \alpha + \beta_1 \times UFP_j + \beta_2 \times EXP_j + \beta_3 \times TEAM_j + \beta_4 \times INTS_j + \beta_5 \times CONC_j + \beta_6 \times FRAG_j$$

也就是針對單一個專案工作項目 $j$ ，其所花費投入的工作人力 $E_j$ ，其中 $j=1\sim N$ ， $N$ 為WBS中專案工作項目總數。滿足上述式子。其中 $UFP_j$  (Unadjusted Function Point)為此工作項目的未調整功能點大小，本研究應用使用案例功能點分析法中的使用者角色及使用案例的複雜度與權重影響，加總兩者之權重影響值來得到所需的未調整功能點(公式2.1)。 $EXP_j$ 表示所有參與此工作項目的人員平均年資， $TEAM_j$ 為工作項目參與專案參與人數， $INST_j$ 為工作項目時程規劃緊密程度， $CONC_j$ 為此工作項目的團隊成員並行程度， $FRAG_j$ 為此工作項目的時間離散度。接續探討本研究設計之預估模型，本模型建構於Randy K. Smith工作指派模型的基礎上，將專案風險影響因素併入工作指派模型中，利用風險分析取得每一個專案工作項目的風險權重值。然後將專案工作項目識別出的各項風險因子，計算度量得到工作項目的風險影響時程，使得風險發生時造成工作項目執行的人天數變動也能反應在真實的專案成本上，相關軟體規模估算影響構面，如圖4-1所示。本研究改良Randy K. Smith的估算模型，衍生出的專案規模估算數學模型樣式如下：

$$\ln E_j = \alpha + \beta_1 \times UFP_j + \beta_2 \times EXP_j + \beta_3 \times TEAM_j + \beta_4 \times INTS_j + \beta_5 \times CONC_j + \beta_6 \times FRAG_j + \beta_7 \times RISK_j$$

其中 $RISK_j$ 表示在此工作項目中發生識別風險因子之風險時所需額外花費的人天數總和。

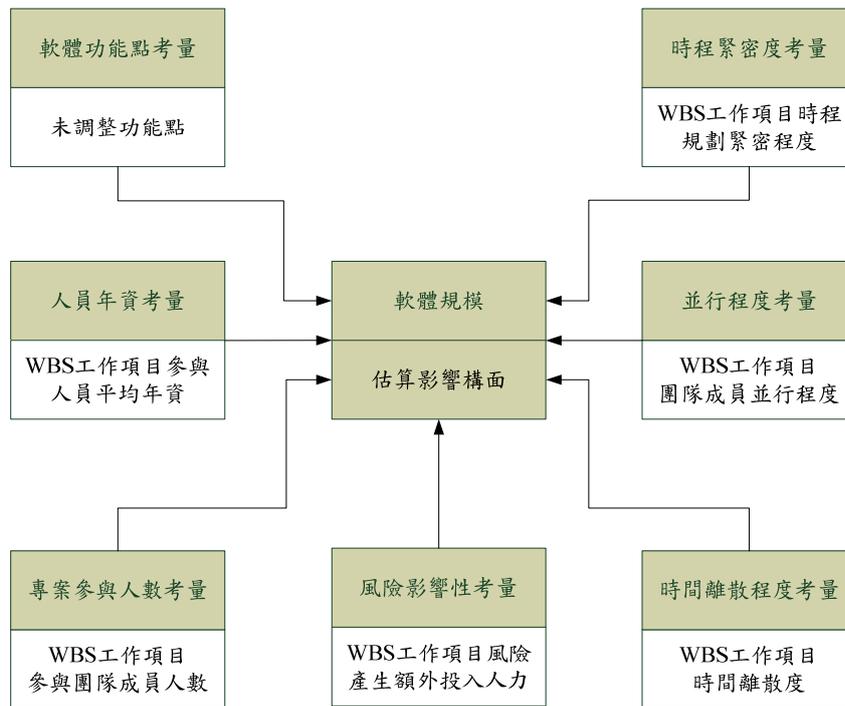


圖 4-1：軟體規模估算影響構面

然後採用作業研究中常見的線性規劃模型求取最小誤差平方和為數學模式原型。模型中的 $\alpha$ 與 $\beta$ 等數值以這七個構面(圖4-1)透過歷史資料進行迴歸分析找出各項參數以求得線性方程式。以便估算專案所需投入之人天數，乘上每單位人日產值即可得知專案所需成本。以下將針對線性規劃模型與本研究預估模型以及估算流程與方法進行說明。

#### 4.1 線性規劃

線性規劃是作業研究這個領域從 1940 年代被提出來之後，最普遍而有效應用的模型之一。在軍事、工業、農業、交通、經濟、醫療系統、以及行為與社會科學等眾多領域都獲致了豐碩的成果，此外可用來解決極大尺度問題的高效率的電腦程式也帶來線性規劃模型在解決實際問題上高度的普及性。除此以外，線性規劃也被廣泛的用來發展其他作業研究問題模型的解決方法，因此對線性規劃的充分瞭解也是對其他作業研究問題模型探討過程中必須具備的基本知識[7]。

線性規劃所能解決的問題通常包含下列特性：(1).其目的為將某個相依變數(dependent variable)極大化或極小化，該相依變數可以由線性的目標函數(objective function)所定義，通常代表某種經濟上或機能上的目標，如利潤、產量、成本或品質指標等等。(2).目標函數中含有許多變數，這些變數的數值描述不同的決策可能性，通常代表某種資源的使用數量如資金、時間、人力、材料等等達成目標所必須使用的項目。(3).對資源的使用有許多條件的限制，其限制內容可以用線性的等式或不等式來描述[6]。

本研究的數學模式，採用作業研究的線性規劃模型，求取最小誤差平方和[5]。當變數不止兩個時，就必須用到複迴歸分析模型[8]。假設此時有三個變數，其代表的資料種類是彼此相關的，無法用線性迴歸兩兩個別處理時，欲求得如下式子：

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i, \quad i = 1, 2, \dots, n \quad (4.1)$$

其中， $\varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$

- 解法一：利用正規方程式求解，以求得其變數間的關係時，此時必須先

將上式擴展成下面三個正規方程式：

$$\sum_{i=1}^n y_i = \beta_0 \cdot n + \beta_1 \sum_{i=1}^n x_{1i} + \beta_2 \sum_{i=1}^n x_{2i} \quad (4.2)$$

$$\sum_{i=1}^n x_{1i} y_i = \beta_0 \sum_{i=1}^n x_{1i} + \beta_1 \sum_{i=1}^n x_{1i}^2 + \beta_2 \sum_{i=1}^n x_{1i} \cdot x_{2i} \quad (4.3)$$

$$\sum_{i=1}^n x_{2i} y_i = \beta_0 \sum_{i=1}^n x_{2i} + \beta_1 \sum_{i=1}^n x_{1i} \cdot x_{2i} + \beta_2 \sum_{i=1}^n x_{2i}^2 \quad (4.4)$$

其中， $n$ 代表資料筆數。將(4.2)、(4.3)、(4.4)聯立可解出 $\beta_0$ 、 $\beta_1$ 、 $\beta_2$ ，代入(4.1)，可得到此三個變數的迴規模型。當變數個為3個以上時，只要利用擴展方程式的方式即可。 $m$ 個( $m \geq 3$ )變數時需要擴展成 $m$ 條方程式，以組成聯立方程式。

● 解法二：以迴歸式求 $\beta_i$ 估計值：

$$\hat{\beta}_1 = \frac{s_{22}s_{1y} - s_{12}s_{2y}}{s_{11}s_{22} - s_{12}^2} \quad (4.6)$$

$$\hat{\beta}_2 = \frac{s_{11}s_{2y} - s_{12}s_{1y}}{s_{11}s_{22} - s_{12}^2} \quad (4.7)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}_1 - \hat{\beta}_2\bar{x}_2 \quad (4.8)$$

$$y = \hat{\beta}_0 + \hat{\beta}_1x_1 + \hat{\beta}_2x_2$$

線性規劃計算求解，將以學生成績舉例說明，例如校方想了解大學轉學考試的數學成績( $x_1$ )與英文成績( $x_2$ )對微積分成績( $y$ )的影響有多高？假設有15筆學轉學生成績資料 $x_1$ 、 $x_2$ 、 $y$ ，又稱為訓練資料，如下表所示，分別以下列求解一、求解二說明線性規劃計算，最後求得線性迴歸式來表示轉學考試的數學成績( $x_1$ )與英文成績( $x_2$ )對微積分成績( $y$ )之間的影响關係。

表 4-1：迴歸分析訓練資料表

座號	$x_1$	$x_2$	$y$	$x_2 \cdot y$	$x_1 \cdot y$	$x_1^2$	$x_2^2$	$x_1 x_2$
1	50	70	86	4300	6020	2500	4900	3500
2	42	89	77	3234	6853	1764	7921	3738
3	61	47	86	5246	4042	3721	2209	2867
4	35	48	63	2205	3024	1225	2304	1680
5	50	58	83	4150	4814	2500	3364	2900
6	39	42	62	2418	2604	1521	1764	1638
7	53	71	85	4505	6035	2809	5041	3763
8	28	49	70	1960	3430	784	2401	1372
9	23	42	63	1449	2646	529	1764	966
10	22	59	72	1584	4248	484	3481	1298
11	70	91	84	5880	7644	4900	8281	6370
12	76	62	85	6460	5270	5776	3844	4712
13	43	53	78	3354	4134	1849	2809	2279
14	40	59	71	2840	4189	1600	3481	2360
15	38	61	79	3002	4819	1444	3721	2318
合計	670	901	1144	52587	69772	33406	57285	41761

方法一：解正規方程式

$$\begin{cases} 1144 = 15\beta_0 + 670\beta_1 + 901\beta_2 \\ 52587 = 670\beta_0 + 33406\beta_1 + 41761\beta_2 \\ 69772 = 901\beta_0 + 41761\beta_1 + 57285\beta_2 \end{cases}$$

$$\begin{cases} \hat{\beta}_0 = 50.5593 \\ \hat{\beta}_1 = 0.3569 \\ \hat{\beta}_2 = 0.1626 \end{cases}$$

迴歸式為  $y = 50.5593 + 0.3569x_1 + 0.1626x_2$

方法二：以迴歸式求  $\beta_i$  估計值

$$\bar{x}_1 = \frac{670}{15} = 44.67 \quad S_{yy} = 1098.93 \quad S_{1y} = 1488.33$$

$$\bar{x}_2 = \frac{901}{15} = 60.07 \quad S_{11} = 3479.33 \quad S_{2y} = 1055.73$$

$$\bar{y} = \frac{1144}{15} = 76.27 \quad S_{22} = 3164.93 \quad S_{12} = 1516.33$$

$$\begin{aligned} \hat{\beta}_1 &= \frac{S_{22}S_{1y} - S_{12}S_{2y}}{S_{11}S_{22} - S_{12}^2} \\ &= \frac{3164.93 \times 1488.33 - 1516.33 \times 1055.73}{3479.33 \times 3164.93 - (1516.33)^2} \\ &= \frac{4710460.2 - 1600835}{11011835 - 2299256.6} \\ &= \frac{3109625.2}{8712578.4} \\ &= 0.3569 \end{aligned}$$

$$\begin{aligned} \hat{\beta}_2 &= \frac{S_{11}S_{2y} - S_{12}S_{1y}}{S_{11}S_{22} - S_{12}^2} \\ &= \frac{3479.33 \times 1055.73 - 1516.33 \times 1488.33}{3479.33 \times 3164.93 - (1516.33)^2} \\ &= \frac{3673233 - 2256799.4}{11011835 - 2299256.6} \\ &= \frac{1416433.6}{8712578.4} \\ &= 0.1626 \end{aligned}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}_1 - \hat{\beta}_2\bar{x}_2 = 50.5593$$

迴歸式為  $y = 50.5593 + 0.3569x_1 + 0.1626x_2$

## 4.2 專案規模預估模型

專案工作項目一般會被指派給一位以上專案成員執行，也就是說，在大多數的專案開發過程中，一個人往往同時會被分派到一個以上的工作項目，而一個工作項目，同時仰賴一個以上成員來完成。而將工作項目分配給專案成員的方法，隨著工作項目或人員的不同而有所差異。常見的分配工作的例子像是將一項工作項目，被委派給一個包含數個成員的團隊來完成，而這個團隊在這項工作項目完成後，才會進行下一項工作項目。或是將工作項目依據複雜度排序，將複雜度較高的工作委派給經驗豐富團隊來執行。

在相關的研究報告中指出一般專案含有五項潛在因素，即所謂的人員平均年資、專案參與人數、工作項目時程緊密度、人員並行程度和時間離散程度[27][32][37][38]。本研究整合這五項潛在因素與加入的風險影響程度潛在因素，結合Randy K. Smith所提出的軟體專案人力估算模型，可用來估算專案投入之人力。接下來本文將針對此五項潛在因素的定義來描述說明。

說明此五項潛在因素定義之前，先針對說明潛在因素所需用到的變數進行定義：

- 變數 $|M|$ ：表示WBS中專案工作項目總數。
- 變數 $j$ ：表示單一專案工作項目 $j$ ，其中 $j=1\sim|M|$ 。
- 變數 $|T|$ ：表示整個專案所需的所有時間單位。
- 變數 $|P|$ ：表示整個專案所有參與的團隊成員數目。
- 變數 $|RF|$ ：表示所識別出的風險因子總數。

此五項潛在因素分別定義如下：

### (1). 人員平均年資(Experience, $EXP$ )

人員平均年資( $EXP_j$ )為參與專案工作項目 $j$ 的所有團隊成員平均年資。此

類資料通常不包含在專案資料及WBS中，需要透過其他管道來取得，一般可經由人資部門取得專案相關人員年資，再予平均求得。

(2). 專案參與人數(Team Size, *TEAM*)

專案參與人數(*TEAM<sub>j</sub>*)表示在專案工作項目j中，累計所有參與的團隊成員數目。專案所有工作項目(*W<sub>ijk</sub>*)所規劃的時程中，任何參與的成員(*Q<sub>ij</sub>*)，都算在此工作項目的專案參與人數中，一個成員最多只能被算一次。

$$Q_{ij} = \begin{cases} 1 & \text{if } \sum_{k=1}^{|T|} W_{ijk} \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

專案參與人數(*TEAM<sub>j</sub>*)的定義如下：

$$TEAM_j = \sum_{i=1}^{|P|} Q_{ij}$$

(3). 工作項目時程緊密度(Intensity, *INST*)

工作項目時程緊密度，即為工作項目時程規劃的緊密程度，定義為專案工作項目j的活動時間單位(*A<sub>j</sub>*)與所規劃開始到結束的連續時間單位(*D<sub>j</sub>*)的比值，即為非空白時程個數除以總時程數。單一項工作項目的時程緊密度為此工作項目在其所規劃開始到結束的連續時間單位中，有多少個單位有一個人員以上進行該開發。

$$INTS_j = \frac{|A_j|}{|D_j|}$$

(4). 團隊成員並行程度(Concurrency, *CONC*)

此程度可以視為在一個工作項目中，所有人員參與的程度。此項程度以此工作項目於各個活動時間單位其開發人員數的平均數表示。例如一個工作項目共有四個活動時間單位，前兩個時間單位分別有四個人員回報參與，後兩個時間單位分別只有兩人參與，則並行程度為 $(4+4+2+2)/4=3$ ，數值3 即為此工作項目人員參與的程度，此即為團隊成員並行程度(*CONC<sub>j</sub>*)。

$$CONC_j = \frac{\sum_{k=1}^{|T|} \sum_{i=1}^{|P|} W_{ijk}}{|A_j|}$$

(5). 時間離散程度(Fragmentation, *FRAG*)

此程度表示在一工作項目中的所有成員，每位成員分別在不同時間點上，同時手頭上正在進行的工作項目個數，此即為時間離散程度(*FRAG<sub>j</sub>*)。

每位成員在時間單位內的時間離散程度(*PFRAG<sub>j</sub>*)定義如下：

$$PFRAG_{ik} = \sum_{j=1}^{|M|} W_{ijk}$$

每位成員在一工作項目的時間離散程度(*PFRAGSUM<sub>ij</sub>*)定義如下：

$$PFRAGSUM_{ij} = \sum_{k=1}^{|T|} (PFRAG_{ik} * W_{ijk})$$

所有參與成員在此工作項目的時間離散程度(*FRAG<sub>j</sub>*)的定義如下：

$$FRAG_j = \frac{\sum_{i=1}^{|P|} (PFRAGSUM_{ij} * Q_{ij})}{TEAM_j}$$

例如，有一個工作項目共有五個成員，每個成員分別在不同的時間點上同時在處理多個工作項目，範例計算如下表。

表 4-2：工作項目的時間離散程度計算範例

	時間單位1	時間單位2	時間單位3
成員A	2	2	3
成員B	3	2	0
成員C	1	3	2
成員D	4	1	0
成員E	0	2	5
此工作項目的時間離散程度為 $((2+2+3)+(3+2+0)+(1+3+2)+(4+1+0)+(0+2+5))/5=6$			

(6). 風險影響時程(Risk Effort, *RISK*)

表示在一工作項目中發生識別風險因子( $RF_j$ )之風險時所需額外花費的人天數總和，即為工作項目風險影響程度與工作項目規劃總人日值( $WBSEffort_j$ )的乘積，亦可稱之為風險影響時程( $RISK_j$ )。

每一個識別風險發生時對於工作項目的影響程度( $PRISK_j$ )定義為識別風險影響權重( $Wt_j$ )與風險發生機率( $Prob_j$ )乘積如下：

$$PRISK_j = \sum_{j=1}^{|RF|} Wt_j * Prob_j$$

工作項目的風險影響時程( $RISK_j$ )的定義如下：

$$RISK_j = \sum_{j=1}^{|T|} (PRISK_j/100) * WBSEffort_j$$

例如，有一個工作項目規劃共需12個人天數，但風險識別出三個風險影響因子，則此工作項目的風險影響時程會增加 $((7.5+45+45)/100)*12=11.7$ 人天數，範例計算如表3-5所示。

### 4.3 估算流程與方法

在專案WBS中專案管理員將專案拆解成子工作項目結構並給予規劃，這些工作項目的規劃包含了子工作項目的時程、工作內容、工作項目人天數、項目的負責人等等(圖4-2)。本研究提供一個從WBS中取得專案參與人數、時程緊密度、人員並行度與時間離散程度的流程方法，如圖4-3。透過此流程可以協助專案管理者從WBS中取得軟體規模與專案成本的估算值。

	WBS	任務名稱	工期	預估開始時間	預估完成時間	負責人員
1	1	1 專案執行控管	120 工作日?	96/10/1	97/3/20	
2	1.1	1.1 需求管理	60 工作日?	96/10/1	96/12/19	Cathychen, Gary
3	1.1.1	1.1.1 訪談	25 工作日	96/10/1	96/11/2	Gary
4	1.1.2	1.1.2 分析	56 工作日?	96/10/5	96/12/19	Cathychen
5	1.2	1.2 專案管理	120 工作日	96/10/1	97/3/20	Cathychen
6	1.2.1	1.2.1 溝通管理	120 工作日	96/10/1	97/3/20	Cathychen
7	1.2.2	1.2.2 範疇/整合管理	120 工作日	96/10/1	97/3/20	Cathychen
8	1.2.3	1.2.3 監控度量	120 工作日	96/10/1	97/3/20	Cathychen
9	1.2.4	1.2.4 品質稽核	120 工作日	96/10/1	97/3/20	Cathychen
10	1.2.5	1.2.5 人力資源管理	120 工作日	96/10/1	97/3/20	Cathychen
11	1.2.6	1.2.6 時程管理	120 工作日	96/10/1	97/3/20	Cathychen
12	1.2.7	1.2.7 成本管理	120 工作日	96/10/1	97/3/20	Cathychen
13	1.3	1.3 規劃分析設計	30 工作日	96/10/1	96/11/9	Gary, David
14	1.3.1	1.3.1 架構設計	14 工作日	96/10/1	96/10/18	Gary
15	1.3.2	1.3.2 規劃	30 工作日	96/10/1	96/11/9	David
16	1.3.3	1.3.3 有關工作項目之各項評估分析事務	10 工作日	96/10/18	96/10/31	David
17	1.4	1.4 資料處理	10 工作日	96/10/3	96/10/16	Claire, Cryil
18	1.4.1	1.4.1 會議資料(報告書、簡報)	10 工作日	96/10/3	96/10/16	Claire
19	1.4.2	1.4.2 資料收集	3 工作日	96/10/11	96/10/15	Cryil
20	1.5	1.5 資訊管理	74 工作日?	96/11/1	97/2/15	Jeff, Ken
21	1.5.1	1.5.1 建構管理	55 工作日	96/11/1	97/1/15	Jeff
22	1.5.2	1.5.2 系統測試	7 工作日?	97/2/1	97/2/15	Ken
23	2	2 部落格建置	30 工作日?	96/10/1	96/11/9	
35	3	3 管理系統擴充	68 工作日?	96/10/1	96/12/31	
46	4	4 介面開發與整合	65 工作日?	96/10/1	96/12/26	
57	5	5 資料庫建置	116 工作日?	96/10/1	97/3/14	

圖 4-2：WBS 結構內容

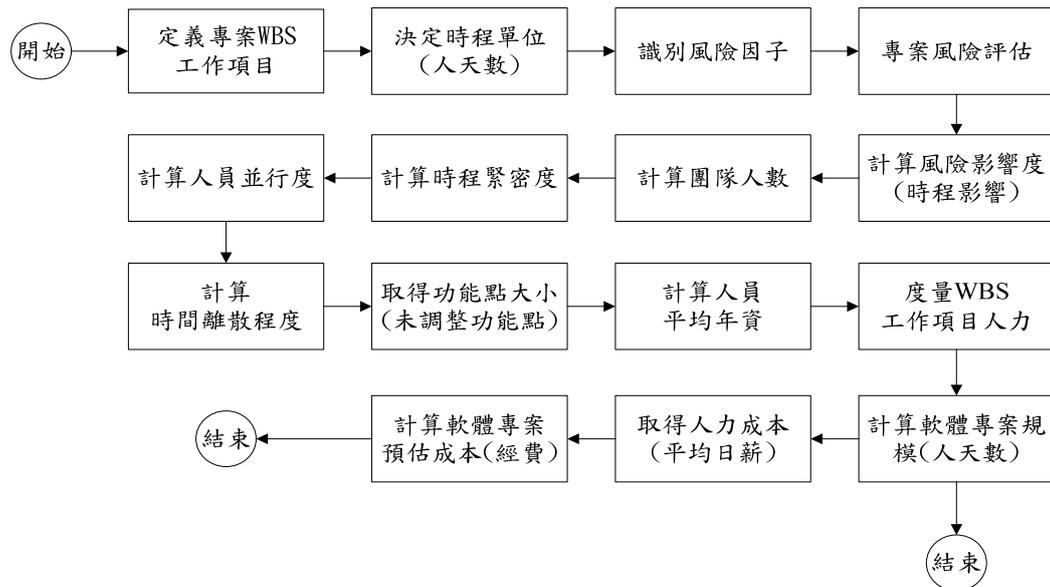


圖 4-3：軟體專案規模與成本估算流程

針對估算流程(圖4-3)之軟體專案規模估算(前十二項活動)以及軟體專案預估成本(最後三項活動)，各項活動說明如下：

- 定義專案WBS工作項目：

以WBS以樹狀結構來表示工作項目，這個活動乃是要讓專案管理者定義每個階層的工作項目為主要工作項目，須留意主工作項目下的子工作項目不得為另一主工作項目。

- 決定時程單位：

每項工作項目在WBS的規劃中都有開始與結束的時間，而常見的時程單位有人天、人週或人月，專案管理者選擇適當的時程單位後，可以計算主要工作項目的時程個數，本研究的時程單位為人天。

- 識別風險因子：

每項工作項目再依照工作屬性細分為多個工作細項(表3-3)。然後將每部分利用Linda Wallace風險因子表(表3-2)加以詳細分析及判斷，以找出最可能的風險事項，識別可能發生的風險因子。

- 專案風險評估：

每項工作項目細項識別出的各項風險因子，依據專案經費、系統結構化程度與技術成熟度，取得風險影響權重表(表3-4)的風險權重值，再藉由風險權重值與風險發生機率乘積總和，得到工作項目細項的風險影響值。

- 計算風險影響度：

根據工作項目細項風險因子的風險影響值計算風險影響時程，即發生風險時所需額外花費的人天數。

- 計算專案參與人數：

在此的專案參與人數為參與此工作項目的總成員數，各成員在此工作項目中只能被計算一次，也就是即使某成員分別出現在所屬的數個子工作項目中，但也只能被計算一次。

- 計算時程緊密度：

此活動主要是取得專案時程中是否有空白的時程，以計算時程的緊密程度。

時程緊密度公式為：

$$\text{時程緊密度} = \frac{\text{非空白時程個數}}{\text{總時程數}}$$

- 計算人員並行度：

此活動主要是在各個專案工作項目透過蒐集各工作時程中的參與人員數量，並取得各工作時程的參與人員平均數，即為人員並行度。

- 計算時間離散程度：

在一般的工作項目規劃中，一個人可能在同一時段被分配到許多工作項目，透過計算一個人在某時程下同時進行的工作項目數量，可以計算出他的工作時程的離散程度。例如某人於某時程同時進行四個工作項目，則此成員於此時程離散程度為4。

- 取得功能點大小：

工作項目的功能點為此工作項目複雜度的一種度量結果，本研究使用使用案例功能點為度量單位，此乃以使用者觀點的功能面來度量專案複雜度。

- 計算人員平均年資：

成員的平均年資為一工作項目中各團隊成員的平均年資。

- 度量WBS工作項目人力：

當取得各因子的結果後，即可代入估算模型，即可得到工作項目的人力估算結果。

- 計算軟體專案規模：

專案WBS中各工作項目皆估算完後，得到一個專案最終的人力，這些工作項目的人力總和，透過這項活動專案管理者可以得到所預測之人力結果，即為軟體專案規模。

- 取得人力成本：

由管理部門取得專案管理者、系統分析設計者、系統開發者與其他專案人員角色的平均日薪資。

- 計算軟體專案預估成本：

專案規模中的投入人天數乘上參與專案人員角色的平均日薪資，得到專案預估經費成本。

#### 4.4 預估模型與企業資訊入口系統整合

有鑑於預估模式須借重歷史資訊的蒐集，本研究預估模型與本研究個案內部的企業資訊入口系統(Enterprise Information Portal, EIP)相互整合，本研究EIP系統具有提供專案歷史資料之能力，保留珍貴之專案歷史資料以為後續專案進行預測與協助預估模式之進行，作為專案時程預估及分析之用。EIP系統以專案管理服務為基礎蒐集專案資訊：專案工時回報記錄、專案工時統計、專案成員工時統計、工作項目統計、專案Loading統計、專案工時分配比率與專案實際成本等專案規模資訊，取得專案實際規模、成本、人力投入數量、預估品質與準確程度，便於未來調整預估模型估算參數與人力平均成本。相關專案管理服務如下說明：

- 專案管理服務

專案管理者新增專案相關資訊，包含專案類型(Project Type)、專案名稱(Project Name)、專案簡稱(Abbreviation)、委託單位(consignor)、預估經費(Budget)、執行期間(Duration)與派任專案經理、專案CM、專案QA。系統實作畫面如圖4-4。

圖 4-4：專案研擬單

由專案管理者研擬新專案並加以管理之服務，經過高階管理者審核流程通過成立專案。由專案管理提出專案研擬申請，經專案部門經理→主任辦公室部門主管→財務組主管→管理部部門主管逐層審核，審核包含專案預估經費

(Budget)、專案執行期間(Duration)與派任專案經理、專案CM、專案QA人選，如圖4-5。



圖 4-5：專案研擬成立審核流程圖

- 建立專案項目管理服務

審查通過之專案依照專案工作細項分類表(表3-3)建立專案工作項目並進行細項分類、指派參與人力與設定工作項目起迄日期，如圖4-6、圖4-7。建立完成的專案項目管理服務可直接將專案工作項目與細項工作分類匯出至 Microsoft Project 應用軟體中，便於專案管理者進行專案管理作業。

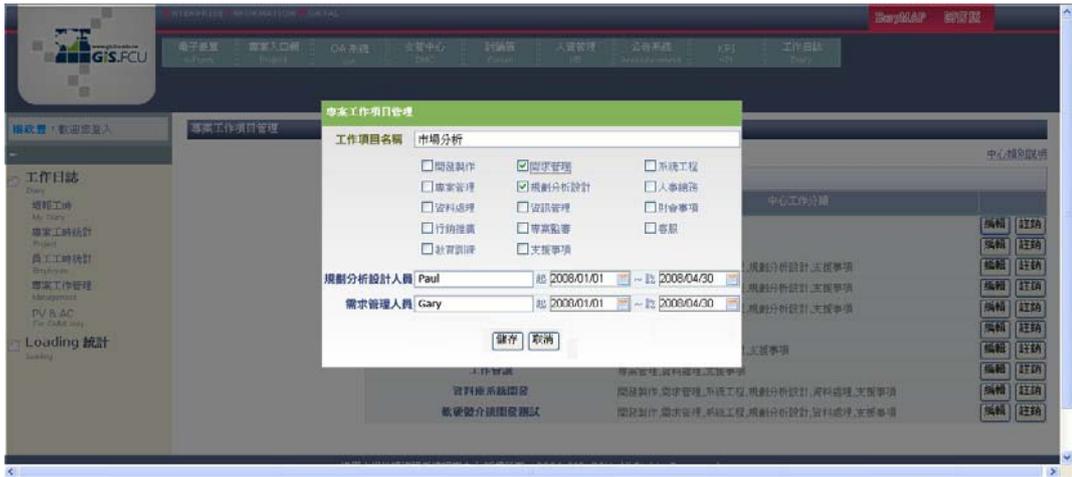


圖 4-6：專案工作項目細項分類設定



圖 4-7：專案項目管理服務

● 計算軟體專案規模

由專案WBS中專案管理員將專案拆解成子工作項目結構並給予規劃，這些工作項目的規劃包含了子工作項目的時程、工作內容、工作項目人天數、項目的負責人等，分別依據本文之估算流程與方法，藉由專案WBS內容取得相關專案規模估算因子(如圖4-8)，如專案參與人數、工作項時程緊密度、人員並行程度、時間離散程度，最後配合功能點、專案參與人員平均年資等專案規模估算因子，代入本文預估模型即可得到軟體專案規模(人天數)，再將投入成員之平均日薪乘上預估人天數，就可以得到專案預估成本(經費)。

	WBS	任務名稱	工期	預估開始時間	預估完成時間	負責人員
1	1	1 系統開發建置	107 工作日?	96/1/1	96/5/26	
2	1.1	1.1 規劃分析設計	7 工作日	96/1/1	96/1/9	Cool, Three
3	1.1.1	1.1.1 架構設計	3 工作日	96/1/1	96/1/3	Cool
4	1.1.2	1.1.2 系統分析	4 工作日	96/1/2	96/1/5	Three
5	1.1.3	1.1.3 系統設計	4 工作日	96/1/4	96/1/9	Cool
6	1.2	1.2 開發製作	8 工作日	96/1/4	96/1/14	Sky
7	1.2.1	1.2.1 核心模組開發	2 工作日	96/1/4	96/1/5	Sky
8	1.2.3	1.2.3 管理介面開發	2 工作日	96/1/12	96/1/14	Sky

圖 4-8：由專案 WBS 內容可取得相關專案規模估算因子

● 編輯專案人員工時記錄服務

提供專案成員填寫工時系統的介面(圖4-9)，以網路服務為基礎的填寫介面，在於方便使用者無論在何處都能輸入工作相關資訊。

圖 4-9：專案成員工時回報介面

● 工時記錄查詢與審核服務

提供專案各成員查詢已輸入的工作時程資訊，專案管理者可以對專案所有參與人員的工作時程進行查詢並審核其工作紀錄；而專案一般成員僅能查詢自身已輸入之工作時程紀錄，如圖4-10。

工作日誌											
2003		2004		2005		2006		2007		2008	
一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
今日工作日誌											
2008 11/4 (二)		8		<input checked="" type="checkbox"/> 加班 <input type="checkbox"/> 簽退 日誌 8 水利三期 至一河局協助驗收遠端監控系統				公假 8 至一河局協助驗收遠端監控系統			
日期	工時	詳細資訊								請假記錄	
1 (一)	8	日誌 2.5 (97)天眼公網 技術會議 日誌 5.5 (97)學術研究(事業) 第十九屆物件導向技術及應用研討會投稿論文格式修正與上傳									
2 (二)	8	日誌 3 保林資訊更新(2) 至林務局南投管理處(祕書林桂煌代為申請) 日誌 5 保林資訊更新(2) 與jeff討論系統擴充功能規劃設計								公假 3 至林務局(祕書林桂煌代為申請)	
3 (三)	8	日誌 6 保林資訊更新(2) 系統擴充報價與驗收功能討論確認 日誌 2 (97)天眼公網 至文心機房與加安電腦測試資料庫同步軟體								公假 2 至文心機房與加安電腦測試資料庫同步軟體	
4 (四)	8	日誌 4 (97)學術研究(事業) 投稿論文資料準備 日誌 4 保林資訊更新(2) 至南投處與孫宗志確認保固款請領前修正功能最後確認								公假 4 至南投處與孫宗志確認保固款請領前修正功能最後確認	
5 (五)	8	日誌 8 台大保林國台 系統功能討論與規劃									
6 (六)	例假日										
7 (日)	例假日										
8 (一)	8	日誌 1 (97)天眼公網 技術會議 日誌 2.5 (97)其他行政(事業) 辦公室空間調整 日誌 4.5 台大保林國台 系統功能討論與規劃									
9 (二)	8	日誌 2 台化擴充 專案工作會議 日誌 3 (97)訓練課程(事業) 8th Manage 專案管理軟體介紹 日誌 3 (97)天眼公網 經銷商MOU草擬									

圖 4-10：專案成員工時回報記錄

● 相關專案工作管理

提供專案工作項目與細項分類工時分析統計圖，如圖4-11、圖4-12，讓專案管理者在專案開發過程中，能隨時對專案進度的監控，以即時反映出專案時程績效(Schedule Performance Index, SPI)，進而使其對專案的預估有更精準的結果。工時分析統計圖能夠協助專案管理者日後於相似專案中訂定工作期程、安排人力與風險評估之參考依據。透過即時的工時資料收集，從工時的角度來幫助專案的管理，以及預估模型的實際檢測資料蒐集，有助於日後專案規模估算因子的調整。提供給專案管理者更有效的預測軟體專案規模，減少專案開發上可能產生的時程或效能偏差與，與提供有效的工時歷史資料，增進未來工時預估的準確度。

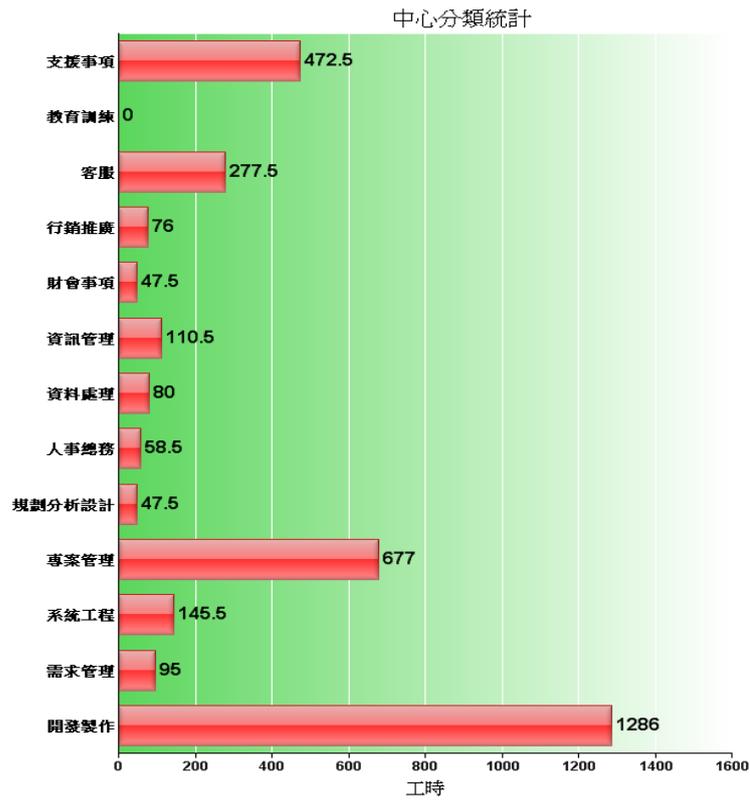


圖 4-11：專案工作細項分類工時投入統計圖

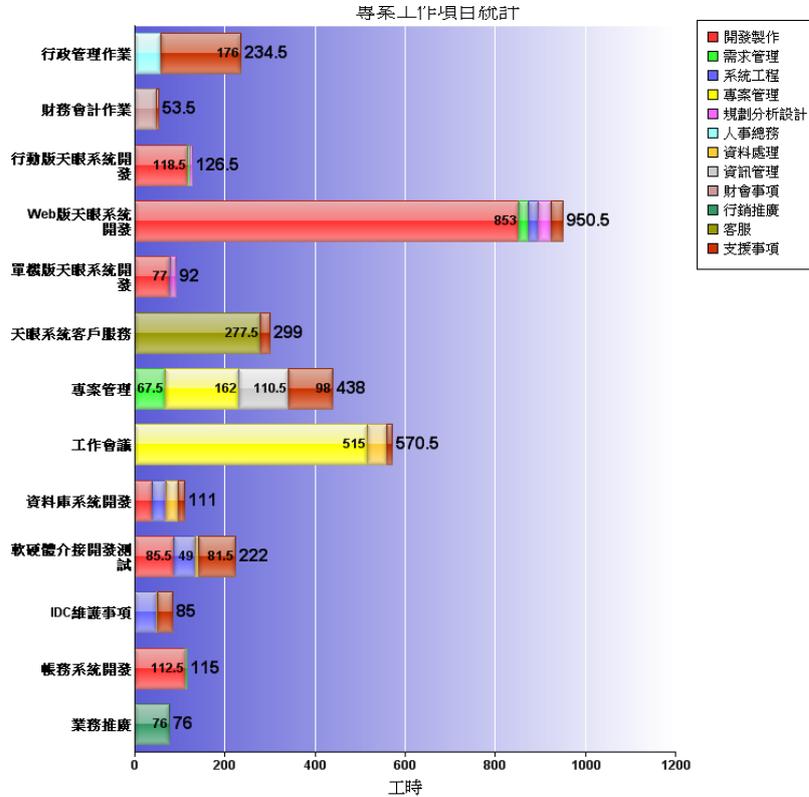


圖 4-12：專案工作項目細項分類工時比率與工時統計圖

● 專案工時統計

提供專案整體性的專案工作項目與團隊成員工時投入明細統計資料(圖4-13)以及專案工作項目投入工時分佈比率(圖4-14),藉由歷史資料的蒐集與分析,讓專案管理者瞭解專案工作細項分類與軟體專案類型、經費之關係,以為日後調整風險影響權重表(表3-4)之參考依據。

水視所KM			
工作項目	中心分類	工作時數	加班時數
<b>專案管理</b>		<b>825.5</b>	<b>15</b>
	需求管理	164.5	2
	專案管理	485.5	5.5
	規劃分析設計	0	0
	資料處理	175.5	7.5
	資訊管理	0	0
	行銷推廣	0	0
<b>單一入口及介面開發、整合</b>		<b>605</b>	<b>18.5</b>
	開發製作	604.5	18.5
	需求管理	0	0
	規劃分析設計	0.5	0
<b>知識管理系統擴充</b>		<b>273</b>	<b>6</b>
	開發製作	253.5	6
	需求管理	4.5	0
	規劃分析設計	15	0
<b>知識物件盤點</b>		<b>29</b>	<b>0</b>
	需求管理	5.5	0
	規劃分析設計	0	0
	資料處理	23.5	0
<b>圖資發佈軟體及資料庫建置</b>		<b>91</b>	<b>0</b>

日誌列表 -- 網頁對話			
填寫人	填寫日期	工作內容	工時
陳雅鈴	2008/07/04	盤點資料整理	1.5
	2008/07/15	河川課專業領域分析	6.5
	2008/08/04	知識盤點問卷分析	4
	2008/10/06	各課室盤點表整理	8
	2008/10/30	各課室盤點結果整理	3.5

圖 4-13：專案工作項目與團隊成員工時投入明細統計圖

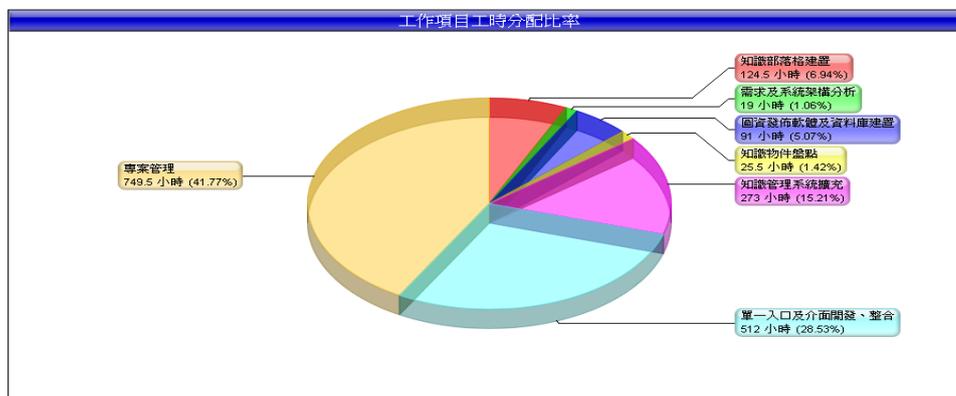


圖 4-14：專案整體工作項目投入工時分佈比率圖

## 第五章、個案研究：某學術研究中心

接著在個案研究部分，首先介紹個案的背景以便了解資料的來源，在以實例介紹軟體專案規模與成本估算流程(圖4-3)中的活動過程，並對所蒐集到的真實案例資料進行分析。這裡的分析主要是針對本研究之估算模型進行統計迴歸分析，自變數為功能點數及專案執行潛在因素，共計有功能點數、團隊成員平均年資、專案參與人數、時程緊密度、人員並行程度時間離散度和風險影響時程七個構面，與依變數人天數(即為專案規模，人天數為專案規模之計算單位)之間相關性分析，利用迴歸分析求解的方法，找出數學模式中的各項估算係數，並對實際個案執行專案規模估算。

### 5.1 個案描述

該學術研究中心創立於民國八十年，歷經多年努力，積極與相關學術單位、政府機構及產業界等合作，推動國土資訊等相關計畫，進行不同領域之研發，累積豐碩成果，深得許多產、官、學界肯定。同時，該單位年年不斷擴充人員與環境設備，至今已達30多位支援教師、90多位全職研究同仁及數十位博、碩士生之組織規模，承接專案之質與量皆穩定成長，並且跨足資訊整合領域，提供業務資訊化整體解決方案，除專案計畫執行及技術研發外，更榮獲多項專利與獎項肯定。

該單位自成立以來，不斷接受公、民營機構委託承辦各類型研發計畫及專案，實際完成公民營機構各類研發計畫與專案七百餘件，並於各學報、期刊(64篇)及國內外研討會不定期發表研究成果(314篇)，撰寫地理資訊系統(GIS)及遙感探測(RS)等相關教材，供各階層學習及教學之參考，每年舉辦相關教育訓練課程，亦多次辦理國中、小「數位地球科技研習營」及高中生GIS競賽活動，為未來科技人才紮根。多年來協助政府推展國土資訊系統之應用，舉辦大型研討會及國土資訊成果展示會，積極推廣GIS、RS、GPS之技術，讓社會各界能運用優質的知性技術，洞悉無風險、無障礙的生存空間。

其主要發展方向以地理資訊系統(Geographic Information Systems, GIS)、

全球衛星定位系統(Global Positioning Systems, GPS)及遙感探測(Remote Sensing, RS)等空間專業知識為核心，配合現代網路化、行動化、視覺化之技術輔助，協助解決現有環境決策問題，近年更著力於「資源管理」、「計畫管考」、「便民服務」、「數位學習」、「車隊監控」、「災害防救」、「專案管理」、「資料倉儲」、「3D GIS應用」及「農村行銷」等領域研究與成果推廣。

## 5.2 個案組織內部流程探討

民國87年間，當時該中心有鑑於品質系統管理為未來研究發展之趨勢，開始推動ISO 9001國際品保制度，並於次年8月正式通過ISO 9001國際品保認證，成為國內第一家通過ISO 9001國際品保認證之學術研究中心，並於民國92年3月再度通過ISO 9001：2000年版品保認證，為達成對客戶提供更高品質的承諾以及因應政府提昇國內產業流程與增進國際競爭力，民國九十五年底遂行導入能力成熟度整合模式(Capability Maturity Model Integration, CMMI)並整合ANS/PMI 99-001-2004(Project Management Body of Knowledge, PMBOK)國際專案管理標準，並於民國97年5月通過CMMI Maturity Level 2認證。該單位期望藉由CMMI的導入建立組織內專案紀律，提供專案所有成員在執行專案中統一遵循的程序，以改善專案流程、提升專案執行品質與能力成熟度。本研究探討之個案專案生命週期與專案流程，如下圖。

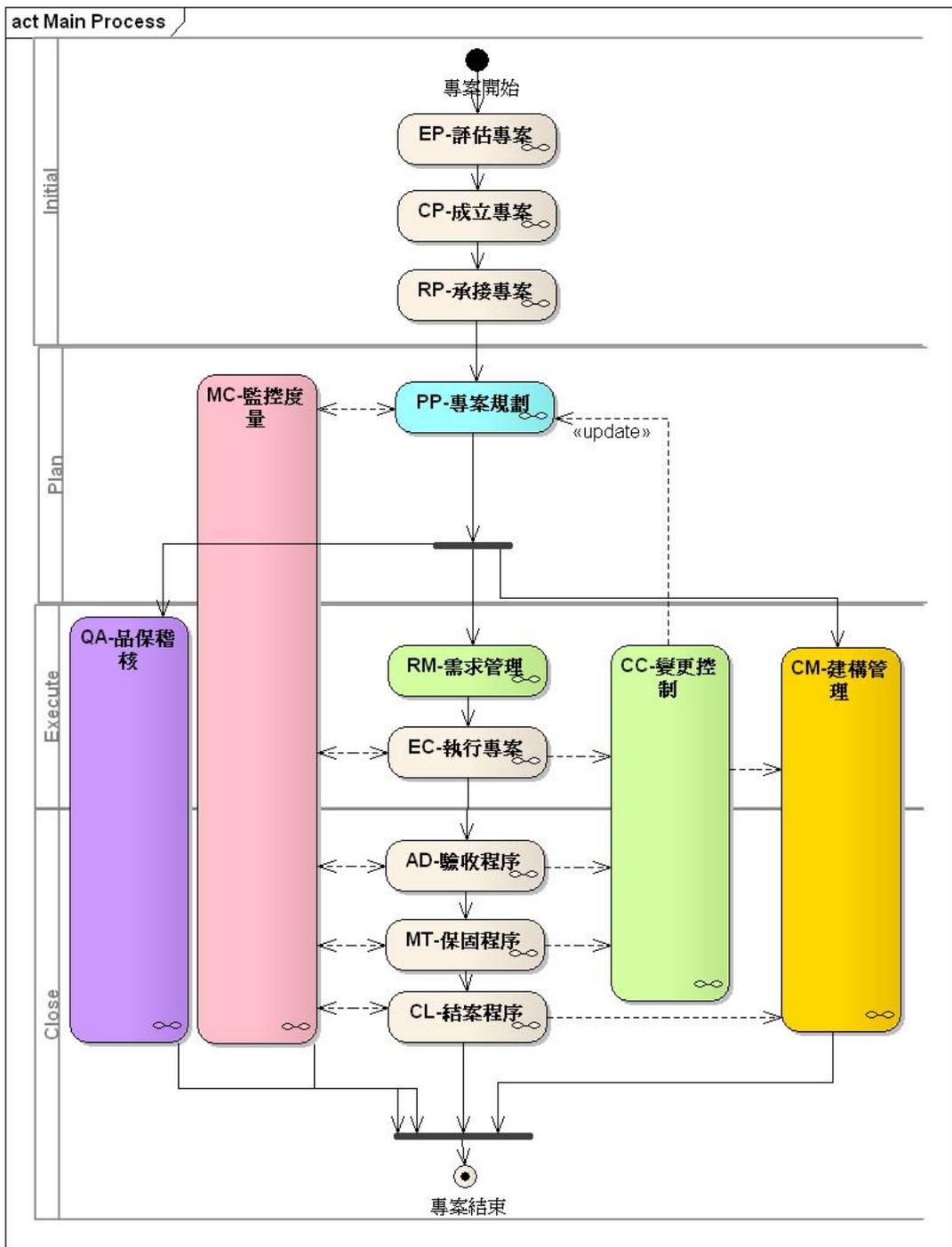


圖 5-1：個案專案生命週期與專案流程

(1). 專案生命週期：

個案承接專案整體執行生命週期，從開始階段獲知可能專案訊息成立專案評估小組開始到結束階段報請驗收，至結案申請單審核通過，可分為以下四階段：

- 開始階段(Initial)：

開始階段簡寫為「I」，從獲知可能專案訊息，並經單位最高負責人、部門主管指派專人組成評估小組進行專案評估作業，評估作業完成決定承接專案後，填寫專案研擬單開始成立專案，並授權指派專案管理者組成專案團隊進行專案承接前置作業及投標，至專案承接完成議價止。

- 規劃階段(Plan)：

規劃階段簡寫為「P」，從專案完成議價後，專案管理者與專案團隊進行專案管理細部作業規劃至完成第1版專案管理計畫書，並經甲方確認止。

- 執行階段(Execute)：

執行階段簡寫為「E」，由專案管理計畫書第1版完成發佈後，開始依據專案管理計畫書各項規劃執行需求訪談分析與工作項目執行，工作項目執行完成後交付甲方，報請甲方發出驗收公文為止。

- 結束階段(Close)：

結束階段簡寫為「O」，甲方發出報請驗收公文，經由甲方依合約內容驗收通過後，進入專案保固程序，最後保固期滿後請領保固金後，由專案管理者填寫結案申請審核通過為止。

(2). 個案內部專案流程：

個案內部專案流程概述說明如下：

- 評估專案流程(EP)：

專案自委託單位提出專案需求，經由該單位最高負責人責成專人組成評估小組進行專案評估作業。

- 成立專案流程(CP)：

評估作業完成決定承接專案後，授權專案管理者成立專案，籌組專案團隊進行備標與投標作業。

- 承接專案流程(RP)：

專案投標評選，得標並與甲方議價簽訂專案合約。

- 專案規劃流程(PP)：

舉凡專案涉及「資源管理」、「計畫管考」、「便民服務」、「數位學習」、「車隊監控」、「災害防救」、「專案管理」、「資料倉儲」、「3D GIS 應用」及「農村行銷」等專案類型時，由專案管理者撰寫並透過專案成員協助，共同訂定產出具有實際執行價值且一致性的【專案管理計畫書】，針對成本、時程、人力與工作項目，進行細分與結構化，建構專案執行紀律，使所有專案成員可依據【專案管理計畫書】所規劃內容與規則，如期如質完成各項工作成果。同時，作為專案執行過程，進行監控與度量的目標及基準。

- 需求管理流程(RM)：

自第一版【專案管理計畫書】完成後開始執行此項流程領域，依據專案特性與委託方需求，由專案管理者制定需求提供及接受準則做為需求接受依據，需求訪談者依據訂定需求提供/接受準則進行需求訪談，並針對需求進行審查，核定之需求彙整並填寫至需求規格書、需求追溯表及進行及維護需求的雙向追溯性。一旦需求提供與接受的雙方達成協議，專案成員全體須依據需求提供及接受準則文件中之需求規格審查準則進行最後確認並簽名認可，以取得所有專案團隊成員對於需求的承諾。並藉由變更控制流程來管理需求變更與紀錄變更理由及使用需求追溯表維護原始需求與所有專案交付產出和專案中所有組成項目之間的雙向追溯性。

- 變更控制流程(CC)：

變更控制流程主要在於由專案在執行階段至結束階段期間，變更項目申請、評估、審查、發佈、執行的流程及準則，做為專案成員在執行變更控制

流程的依據。當發生專案需求變更時，專案管理者依據變更請求單位之變更請求，並提出影響評估及可能解決方案之說明，並填製變更影響評估表，經由變更控制委員會成員依其權責審查變更請求，變更通過後登錄變更至變更管理系統並通知專案成員。最後依據變更的事項修改專案管理計畫及變更影響程度更新專案基準，並將變更的事項公布，讓專案成員了解及執行變更內容。

- 品質稽核流程(QA)：

藉由客觀的審查流程與產品品質保程序及活動，確保專案確實依據規劃流程、步驟及規範執行；並持續追蹤未落實部份直至完成。此流程之目的為讓專案團隊成員與管理階層能夠客觀洞察流程與專案相關工作產品，藉由客觀的流程與產品品質保證活動，了解所有專案人員是否依照所制定的工作流程、標準、作業程序等執行工作。在專案執行的過程中，執行流程與產品品質保證的結果與發現，能夠讓專案成員及管理階層及早了解現行專案作業與規定不符合之處，及早處理問題，或針對未來可能面臨之風險，尋求對策之參考。

- 建構管理流程(CM)：

比較基準之建立為專案系統開發及執行依據，建構管理紀錄歷史異動內容以維持比較基準之一致性。依專案計畫工作需求，識別專案生命週期各階段需要建構管理之建構項目，並根據準則來選擇建構項目或工作產品之組成項目，以建立建構項目清單。依據不同建構項目，確定是否需建立比較基準(如功能的基準、產品的基準等)。比較基準建立時須涵蓋上一次比較基準之內容，並以建立日期為存放比較基準之目錄名稱。

- 監控度量流程(MC)：

監控度量之主要目的係針對專案進行時可能影響專案品質的各項度量，進行定期與不定期之監控，以確保專案品質之一致性。專案於規劃階段時，便必須針對該專案需要度量之項目進行設定、度量項目之取得方式以及矯正

措施之施行。其重點乃為監控專案之各項參數，確保專案如期如質如預算完成，並能回饋做為組織流程資產。由專案管理者識別影響專案之時程、成本、品質、團隊技能、風險、建構管理、關鍵人員參與，並將其量化成度量項目，依度量監控頻率監控其變化。最後，由專案管理者蒐集監控度量值產生監控度量記錄表，定期分析結果於每次會議與專案團隊檢討並採取矯正措施直到專案結束，於專案結束後，將歷次監控度量記錄彙整，並針對專案進行期間各項度量項目進行檢討(Lesson Learned)，針對足以影響專案品質、期程及預算的各項參數，進行定義與設計，加以蒐集、分析、解釋與行動，做為提升專案品質、縮短專案期程、提高專案利潤的依據，並回饋作為組織流程資產。

- 執行專案(EC)、驗收程序(AD)、保固程序(MT)、結案程序(CL)流程：

依據委託方確認之需求規格書，進行工作項目規劃、設計與執行，工作項目執行完成後交付委託方，報請委託方發出驗收公文，經由委託方依合約內容驗收通過後，進入專案保固程序，最後保固期滿後請領保固金後，專案管理者填寫結案申請，並針對專案進行期間各項度量項目召開專案結案討論會議進行檢討，針對足以影響專案品質、期程及預算的各項參數，進行定義與設計，加以蒐集、分析、解釋與行動。最後，由專案管理者彙整相關專案資料燒錄並繳交成果報告回饋作為組織流程資產。

### 5.3 資料來源

這個實際研究個案，內部具有完整的企業資訊入口包含工作日誌系統、OA 系統、文管系統、人資系統、專案電子表單系統，完整蒐集了由2005年1月至2008年10月間，該單位執行的430件專案歷史資料中委由個案組織內部之高層專案管理者根據部門發展方向、技術核心、研發創新、研究教學等角度，由430件專案歷史資料選出具有代表性的五十三筆(附錄一)專案WBS主要工作項資料(EIP 工時記錄、專案風險評估記錄、專案甘特圖、專案人力架構資料、專案結案記錄、人員年資等)，其資料涵括「資源管理」、「計畫管考」、「便民服務」、「數位學習」、「車隊監控」、「災害防救」及「資料倉儲」

等專案，資料分佈各別為「資源管理類專案：八筆」，「計畫管考類專案：七筆」、「便民服務類專案：八筆」、「數位學習類專案：七筆」、「車隊監控類專案：八筆」、「災害防救類專案：八筆」及「資料倉儲類專案：七筆」。藉由內部系統統一化WBS工作項細項類別，專案經理透過工作日誌系統進行專案WBS工作項規劃，專案執行時相關成員依據規劃WBS工作項填報工時，如圖5-3。因為所有專案有一致的工作細項分類，如圖5-2，有助於提昇專案規模預估的精確性。

專案工作項目管理

工作項目名稱 知識部落格建置

開發製作       需求管理       支援事項  
 專案管理       規劃分析設計       人事總務  
 中心工作分類  
 資料處理       資訊管理       財會事項  
 教育訓練       專案監審       客服

儲存 取消

圖 5-2：工作日誌系統-專案 WBS 工作項規劃

所有列表 最常使用

管理部 水規所KM  
 推廣部 金車e化  
 巨鵬辦公室 水利三期  
 主任辦公室 保林資訊更新(2)  
 研發部 中華郵政監控  
 環資部 (96)台塑維護  
 科管部 涸闌農村  
 事業部 二所農村推廣  
 魅力農村120  
 華山行銷(3)

專案工作項目 中心工作分類

單一入口及介面開發、整合      開發製作  
 知識管理系統擴充      需求管理  
 知識物件盤點      規劃分析設計  
 圖資發佈軟體及資料庫建置  
 既有圖資盤點及詮釋資料建立  
 需求及系統架構分析

專案階段 I (初期) P (規劃) E (執行) O (結案)

加班內容 整合介面模組開發

開始時間 17:30

0.5	1	1.5	2	2.5	3	3.5	4
4.5	5	5.5	6	6.5	7	7.5	8
8.5	9	9.5	10	10.5	11	11.5	12
12.5	13	13.5	14	14.5	15	15.5	16
16.5	17	17.5	18	18.5	19	19.5	20
20.5	21	21.5	22	22.5	23	23.5	24

儲存 取消

圖 5-3：工作日誌系統-工時填寫回報

## 5.4 資料分析

本研究分析工具使用統計軟體 SPSS10.0 中文版來執行統計分析的部分，計算所有自變數的迴歸係數，目的是為了驗證這七個不同構面：未調整功能點、人員平均年資、專案參與人數、工作時程緊密度、人員並行程度、時間離散程度、風險影響時程(時間成本)等自變項對依變數：人天數(專案規模)是否有顯著的估算作用？進而分析究竟這七個構面自變數對依變數人天數(專案規模)的估算能力為何？由於本研究自變數不只一個，有七個自變數，係屬迴歸分析中的多元迴歸分析或稱複迴歸分析[19]。因此，必須了解變數之間的相關情形，並驗證自變數七個構面對依變數人日的估算是否顯著，最後探究自變數七個構面對依變數人天數(專案規模)的估算能力？為此五十三筆訓練資料進行迴歸相關分析，可得到本研究模型之相關係數表，如表 5-1。以及看出自變數與依變數之間相關的強弱程度及方向。本研究模型與 Randy K. Smith 估算模型的相關係數 $\alpha$ 與 $\beta$ 數值，如下所示：

Randy K. Smith 的估算模型

$$\ln E_j = 2.524 + 0.02269 \times UFP_j - 0.146 \times EXP_j + 0.167 \times TEAM_j - 0.776 \times INTS_j - 0.454 \times CONC_j + 0.01139 \times FRAG_j$$

本研究模型

$$\ln E_j = 2.091 + 0.005191 \times UFP_j - 0.07996 \times EXP_j + 0.124 \times TEAM_j - 0.552 \times INTS_j - 0.315 \times CONC_j + 0.008505 \times FRAG_j + 0.0616 \times RISK_j$$

表 5-1：迴歸分析表

係數(a)					
模式	未標準化係數		標準化係數	t	顯著性
	B 之估計值	標準誤	Beta 分配		
(常數)	2.091	.271		7.723	.000
1 未調整功能點	5.191E-03	.011	.025	4.70	.641
人員平均年資	-7.996E-02	.039	-.135	-2.064	.045
專案參與人數	.124	.042	.188	2.935	.005
工作項時程緊密度	-.552	.182	-.189	-3.035	.004
人員並行程度	-.315	.116	-.267	-2.717	.009
時間離散程度	8.505E-03	.001	.848	8.671	.000
風險影響時程	6.160E-02	.009	.413	6.887	.000

a 依變數\：ln(人日)

不僅如此，從表 5-2 迴歸係數的變異數分析(ANOVA)摘要表中，顯示當自變數：未調整功能點、人員平均年資、專案參與人數、工作項目時程緊密度、人員並行程度、時間離散程度、風險影響時程(時間成本)等七個構面均投入迴歸模式中，如表 5-2 所示，整體迴歸模式之 F 值等於 53.147， $p=0.000 < 0.05$  達顯著水準，表示自變數：未調整功能點、人員平均年資、專案參與人數、工作時程緊密度、人員並行程度、時間離散程度、風險影響時程(時間成本)等七個構面，確實可以有效估算出軟體規模對依變數：ln(人日)所進行的估算。總而言之自變數七個構面對依變數：ln(人日)的估算顯著。

表 5-2：迴歸係數的變異數分析(ANOVA)摘要表

變異數分析(b)						
模式		平方和	自由度	平均平方和	F 檢定	顯著性
1	迴歸	21.440	7	3.063	53.147	.000(a)
	殘差	2.593	45	5.763E-02		
	總和	24.033	52			

a 預測變數：(常數), 風險影響時程, 專案參與人數, 未調整功能點, 人員並行程度, 人員平均年資, 工作項時程緊密度, 時間離散程度  
b 依變數\：ln(人日)

表 5-3：模式摘要表

模式摘要				
模式	R	R 平方	調過後的 R 平方	估計的標準誤
1	.945(a)	.892	.875	.24006011459814

a 預測變數：(常數), 風險影響時程, 專案參與人數, 未調整功能點, 人員並行程度, 人員平均年資, 工作項時程緊密度, 時間離散程度

雖然自變數：未調整功能點、人員平均年資、專案參與人數、工作項目時程緊密度、人員並行程度、時間離散程度、風險影響時程(時間成本)等七個構面對依變數：ln(人日)的估算作用顯著，但是究竟這些自變數對依變數：ln(人日)的估算能力為何？關於這個問題，可以在表 5-3 模式摘要表中找到所要的解答，在表 5-3 內顯示相關係數 R 等於 0.945，決定係數 R 平方( $R^2$ )等於 0.892，迴歸模式的估計標準誤為 0.2401，這些數值明顯地表示自變數：未調整功能點、人員平均年資、專案參與人數、工作項目時程緊密度、人員並行程度、時間離散程度、風險影響時程(時間成本)等七個構面，可解釋依變數：ln(人日)總變異量的 89.2%。總而言之，在這個研究個案中的自變數：未調整功能點、人員平均年資、專案參與人數、工作項目時程緊密度、人員並行程度、時間離散程度、風險影響時程(時間成本)等七個構面對依變數：ln(人日)有 89.2%的解釋力，而依變數不能被自變數預測的部份僅僅為 10.8 % ( $1-0.892=0.108$ )。

## 第六章、研究結果

線性迴歸模式的適合性(Goodness of Fit)測試，此測試的目的是在於了解模型中迴歸曲線與歷史資料的契合程度，而估算品質(Quality of Estimation)的測試，即模型對於新資料的估算品質，了解模型估算的能力。透過這些測試可以了解歷史資料與所建立的模型契合程度，也可以知道模型對於新資料的估算品質[13][21]，最後得到專案預估的準確度。

### 6.1 分析結果

由5.4節資料分析可得知，人員平均年資、專案參與人數、工作項目時程緊密度、人員並行程度、時間離散程度、風險影響成本(時間成本)，六項影響因子在本研究模型中對於預測結果的影響程度，如表6-1所示。

表 6-1：各項因子於本研究模型中的人天數影響性

因子	關係係數	P-value	影響性	說明
人員平均年資	-0.07996	0.045	負相關	隨著年資增加助於縮短專案執行人天數。
專案參與人數	0.124	0.005	正相關	參與人數因子對於預測結果具有正相關性，也就是人數的增加所需投入專案管理成本提升，相對增加預測人天規模。
工作項目時程緊密度	-0.552	0.004	負相關	時程緊密度主要是測量工作項目的時程規劃緊湊程度。當時程緊密度增加時，表示團隊成員能夠有專注的時間在指定的工作項目上，有助於提高工作效率，縮短專案時程。
人員並行程度	-0.315	0.009	負相關	當人員並行程度提昇時(此提昇反應了團隊在合作關係上深度與默契的增加)，工作項目的人力隨著合作關係增加而縮短，在團隊合作程度提昇的同時，會使得團隊成員花較少的時間在手頭的工作項目上，此現象說明了透

				過團隊合作來提高工作效率。
時間 離散程度	0.00851	0.000	正相關	時間離散度反應了工作項目中的團隊成員，其在工作時間上被其他工作項目影響的程度，由結果可知當時間離散度提昇時，相對會提昇預測規模，故兩者存在正相關性。
風險 影響時程 (時間成本)	0.0616	0.009	正相關	可表示當工作項目規劃中識別出的風險因子愈多時，專案應變成本也會隨之提昇，預測人天規模也會增加。

由表6-1結果得知，平均年資因子對於預測結果具有負相關性，也就是隨著團隊成員年資的增加，有助於累積專案執行與專業技能經驗值，助於縮短專案執行時間人天數(專案規模)。

專案參與人數因子對於預測結果具有正相關性，也就是參與人數的增加所需投入專案管理成本也會跟著提升，所以預測專案規模也會增加。

人員並行程度對於預測結果具有負相關性，當人員並行程度提昇時(此提昇反應了團隊在合作關係上深度與默契的增加)，工作項目的人力隨著合作關係增加而縮短，在團隊合作程度提昇的同時，會使得團隊成員花較少的時間在手頭的工作項目上，此現象說明了透過團隊合作來提高工作效率。

時程緊密度主要是測量工作項目的時程規劃緊湊程度。當時程緊密度增加時，表示團隊成員能夠有專注的時間在指定的工作項目上，時程緊密度降低表示團隊成員在規劃期間中，有許多空白時間，工作項目非在連續的時間單位內完成。由結果可得知當時程緊密度的增加時，會使預測結果降低，與學者DeMarco的研究發現相似[32]。

時間離散度反應了工作項目中的團隊成員，其在工作時間上被其他工作項目影響的程度，由結果可知當時間離散度提昇時，相對會提昇預測結果，

故兩者存在正相關性。學者DeMarco建議專案管理員在團隊成員完成一個工作項目後，才規劃指派下一個工作項目[32]。

風險影響時程因子與預測結果有正相關性，可表示當工作項目規劃中識別出的風險因子愈多時，投入專案應變成本也會隨之提昇，以因應未知的專案風險，所以預測人天規模也會增加。

## 6.2 估算模型適合性測試

線性迴歸模式的適合性(Goodness of fit)測試主要是為了解模型中的回歸曲線與輸入資料的契合程度，使用到的參考指標有R平方( $R^2$ )、調過後的R平方(Adjusted R Square)。R平方於迴歸分析中表示模型透過X去預測Y的預測釋力，即Y變項被自變項X所解釋的比率。由於以樣本統計量推導出來的R平方來評估整體模式的解釋力，並進而推論到母群體時，可能會有高估的傾向即樣本數越小，越容易高估，解釋力膨脹效果越明顯，樣本數越大，膨脹情形越輕微，故透過調過後的R平方(Adjusted R Square)，可以減輕因為樣本估計帶來的R平方膨脹效果。R平方與調過後的R平方能反應自變項與依變項所形成的線性迴歸模式的適合性。F 統計量(F Static)目的是為判定因變數與所有自變數所成的集合間是否有顯著關係，其計算方法為均方迴歸(Mean Square Regression, MSR)除以均方誤差(Mean Squared Error, MSE)[31]。另一項測試線性迴歸模式的適合性指標為均方根誤差 (Root Mean Square Error, RMSE)，其計算公式如下：

$$RMSE = \sqrt{\frac{\sum_{j=1}^{|M|} (E_j - EA_j)^2}{|M|}}$$

公式中|M|表示在資料集中的主工作項目總數， $E_j$  為主要工作項目 j 其預測出人力， $EA_j$  為主要工作項目 j 其實際所使用的人力，此人力即透過分析開發人員時間紀錄後所得的真實人力大小，一個模型的均方根誤差愈小，則線性迴歸模式的適合性愈高。表 6-2 為原模型與本研究模型 R 平方( $R^2$ )、調過

後的 R 平方與均方根誤差的測量結果。由結果可知本研究模型在 R 平方(R<sup>2</sup>)與校調過後的 R 平方的表現上皆比原模型來的高，即當增加工作項目的風險影響時程(時間成本)因子後，本研究模型能夠有較好的人力預測釋力。而本研究模型在均方根誤差上比原模型來的小，表示本研究模型與原模型在擁有相同的專案歷史資料下，本研究模型的能夠有較高的線性迴歸模式的適合性。而本研究模型在 F 統計上也較原模型高，表示本研究模型中估算人力與七項因子的整體關係較原模型更為顯著，證明本研究模型因為導入風險評估機制所取得的專案工作項目風險影響時程(時間成本)因子，能夠顯著的提昇人力的估算。

表 6-2：線性迴歸模式的適合性測量結果

	原模型	本研究模型	比較結果
適合性(Goodness of fit)分析			
R 平方(R <sup>2</sup> )	77.8%	89.2%	本研究有較好的人力預測釋力
調過後的 R 平方	74.9%	87.5%	本研究加入風險因子後有較好的人力預測釋力
均方根誤差 (RMSE)	0.34	0.24	本研究在真實人力與估算人力上誤差較低，估算模型適合性較好

### 6.3 估算品質測試

一個好的估算模型，除了本身在線性迴歸模式的適合性上有好的表現，對於新的資料也能夠有好的估算能力。因此測量一個模型對於新進資料的估算品質是很重要與實際的。透過一個模型的相對誤差振幅(Magnitude of Relative Error, MRE)可以得知模型對於新資料的估算的品質，相對誤差振幅公式如下[31][35]：

$$MRE_j = \left| \frac{E'_j - EA_j}{EA_j} \right|$$

公式中 $E_j'$ 為主要工作項 $j$ 在模型中所預測出人力，唯此模型建立使用的歷史資料中，不含工作項 $j$ 的歷史資料，即工作項 $j$ 相對於估算模型為新的資料。 $EA_j$ 為主要工作項 $j$ 其真實的人力，即透過分析開發人員時間紀錄的真實人力大小。表 6-3 為原模型和本研究模型估算品質測量結果。本研究模型的 MRE 平均值較原模型的平均值為低，表示本研究模型透過加入主工作項目風險影響時程(時間成本)的因素後，對新資料的預測平均上能夠比原模型有更好的表現。而由 MRE 的標準差可了解模型的穩定性，原模型與本研究模型在標準差上本研究模型的數值較低，故本研究模型穩定性較加。最後一項計算 MRE 小於或等於 0.2 的個數占整體的比例[31][35]，而本研究模型的表現上較為優異，表示原模型與本研究模型在擁有相同專案歷史資料下，本研究模型對於新資料的測量結果品質高者較原模型來的多。

表 6-3：估算品質測試結果

	原模型	本研究模型	比較結果
相對誤差振幅 (Magnitude of Relative Error, MRE)			
Average	0.3103	0.1989	本研究模型平均上有較佳的估算品質
Within 20%	0.4038	0.5769	本研究模型估算性能較佳

(資料來源：本研究個案資料, 附錄二、三)

## 第七章、結論與建議

本研究提出之專案規模預估數學模式可由從專案WBS進行專案所需投入人力之估算。研究並使用專案參與人數、時程緊密度、人員並行程度、時間離散度、使用案例功能點和團隊成員平均年資等因子，及整合專案風險因子因素，建立出以專案人力投入基礎的軟體規模預估模式。經資料驗證的結果顯示本研究模型在迴歸模式適合性上相較於Randy K. Smith估算模型有較好的預測釋力，並且對於未知、新的專案WBS有較好品質的估算結果，能夠提昇專案估算的準確度，避免專案人力資源因沒有妥善分配，造成資源投入不足或資源浪費。本研究之預估模型也實際整合了EIP系統，藉由系統的輔助進行工時資料的蒐集，最後規劃與設計了工時日誌與專案工作管理服務深入進行廣泛的研究統計，提供專案相關工時歷史資料，協助專案管理者從工時的角度來管理專案。

### 7.1 研究貢獻

對於本論文的研究成果，具有實用性、學術性貢獻與價值，具體貢獻說明如下：

- 使用一般功能點分析法的最主要缺點，須對該方法論有相當瞭解，一般非資訊背景的專案管理者不易上手，易造成預估結果因為個人主觀因素而造成落差過大。本研究利用迴歸分析求出的軟體規模估算模式，提供專案管理者一套比較簡易的程序，可藉由 WBS 的工作項目掌控各項專案資源規劃與管理。相對功能點分析法的繁複程序，實施的可行性增高了。
- 本研究結合風險分析的評估流程與研究結果，建立一套簡單的軟體專案風險分析的標準流程，不但提供專案管理者自行評估的功能，且可很快的得知各風險因子的重要程度。另外，本研究也可提供管理者評估專案現況風險，使專案管理者能及時掌控風險管理的資訊。
- 在專案風險評估的問題本身就具有主觀、模糊及不明確性，故本研究將 Linda Wallace 提出的風險構面與風險因子引入採用 Boehm 風險分析架構中並配合制式化 WBS 工作項目分類的設計，使專案管理者能整體性思考

風險因素影響專案的程度，且又能反應風險問題的特性，減少人為過於主觀的判斷。

- 本研究以個案實際的管理數據，建立起一套軟體規模估算模式。不僅可用於公司組織內部資源的分配，亦可使用於軟體開發中各階段快速預估人力投入分析，幫助軟體開發者可快速累積蒐集之歷史資料，供後續學術研究的參考。
- 提供「軟體專案規模與成本估算流程」之標準化作業，以協助軟體產業在進行軟體專案預估之導入時的重要參考依據，以降低軟體廠商建置軟體專案預估流程與工具所需的時間與成本。
- 將正規化之軟體專案預估模式技術導入軟體產業軟體開發工作量、成本與時間預估之精確度，使軟體產業可以精確地控制開發生命週期各階段的預算、資源分配以及開發進度，以確保軟體的品質。
- 本預估模型整合 EIP 系統針對專案資訊進行蒐集與分析，提供多面向的專案資訊，在專案工作項目、細項工作分類、團隊成員投入工時的分析報表上有深入的研究統計，能提供有效的工時歷史資料，助於後續專案規模預估模型的預測準確度提昇與估算因子的調整。
- 透過不斷累積蒐集之歷史資料，以協助軟體產業界在後續軟體專案預估模式精確性的提昇。

## 7.2 後續研究建議

本研究的分析個案內部具有完整的企業資訊入口，包含工作日誌系統、OA 系統、文管系統、人資系統，專案電子表單系統，以及內部定期辦理相關專案管理、工作分工結構之教育訓練，並運用本研究模型建置提供自動化規模預估平台，協助專案管理者可於專案備標、需求規劃、測試驗收與結案階段(歷史資料蒐集)執行規模估算，此乃有效提昇專案規模估算準確度的關鍵。因為影響軟體專案的風險因素非常多且不易控管，此研究將涉及到整個軟體產業未來的市場規模。本研究標的僅止於前段之風險評估流程，另外屬於中段的風險計畫、風險減緩、風險監督等風險控制及危機處理，以及後段的專

案執行的經驗學習等，皆為值得後續研究的課題。建議未來研究的方向有：

- 可在專案工作項目的風險評估上加入人力、品質、預算限制以及時間工作順序之限制，來做綜合的衡量，使風險評估更趨完善。
- 本研究中所提出的風險評估機制之專案風險影響權重訂定方法與專案分類方式尚有許多改進之空間，也是一個值得研究的方向。
- 發展軟體專案規模預估及風險評估控制系統，結合本研究之專案規模預估模型與風險評估程序，建立一套簡單的軟體專案規模計算以及風險評估與監控系統，不但提供專案管理者自行評估的功能，且可很快的得知各風險因子於 WBS 工作項目中的重要及影響程度。另外，該系統也可提供專案管理者於軟體開發生命週期(Software Development Life Cycle, SDLC)從專案備標、專案規劃、設計階段、專案結案階段的規模估算及專案現況風險評估，使專案管理者能及時掌控專案規模與風險管理的資訊。
- EIP 系統針對資料的收集與分析再加強，提供更多面向的工時資訊，作更深入更廣泛層面的研究統計，藉由更多層面的專案工時預估歷史資料，增進未來工時預估的準確度。

最後，本研究目前針對軟體專案進行人力投入規模估算。因為，本研究模型中功能點因子採用使用案例功能點分析並不涉及軟體開發相關技術，使得本模型的應用領域適用性並不侷限軟體專案，更可適用於非軟體類專案，如人力派遣、產品行銷、系統工程類型專案等。能夠協助專案管理者精確快速地預估專案所需的時間成本不僅有助於專案管理的規劃與監控等工作更有效進行，更可幫助專案管理者在規劃與開發階段妥善地規劃開發時程與資源配置與時間控管，進而達成使用者對專案品質的要求。希望本研究成果能使軟體專案管理能更上軌道，但未來仍須不斷回饋更多的數據及定期調整估算參數，使估算模式能契合開發環境的變化；亦希望能誘使其他更進一步的學術研究投入，使軟體工程的規模估算領域更實務化。

## 參考文獻

### 中文文獻

1. 王德榮，「以功能點估算軟體規模之研究」，碩士論文，國立中山大學資訊管理學系碩士在職專班，2003。
2. 王慶富，「專案管理」，聯經出版事業公司，1996。
3. 林大舜、張啟昌，「建構資訊推動計劃風險管理模式之研究」，管理與系統，第十卷第三章，p303-328，2003。
4. 林信惠、黃明祥、王文良，「軟體專案管理」，智勝出版，2002 初版。
5. 李茂興 譯，Michael E. Hanna 著，「作業研究」，揚智文化，2000。
6. 胡衣臨，「軟體功能點數計算精確度的影響因素與提昇之研究」，碩士論文，國立臺灣科技大學資訊管理研究所，2005。
7. 施宣光，線性規劃(Linear Programming)，  
<http://course.ad.ntust.edu.tw/AD5505701/OR/linear.htm>，2008。
8. 黃登源 著，「應用迴歸分析」，華泰文化事業公司，1998。
9. 陳建名，「軟體專案之風險分析」，國立中山大學資管所 碩士論文，1995。
10. 許榮榕，「系統方法專案管理」，天一出版社，1995。
11. 許光華、何文榮，「專案管理-理論與實務」，華泰書局，1998。
12. 葉蘇蓉，「建立軟體專案風險衡量模式之研究-加入組織慣性變項」，暨南大學資管所 碩士論文，1999。
13. 張文貴、楊政豐、黃映瑞、穆青雲，「加入風險分析之軟體規模預估模式的研究」，第十九屆物件導向技術及應用研討會，p114~121，2008。
14. 張家豪、黃世禎、韓文銘，「軟體風險因子之動態辨識」，資訊管理學術與實務研討會，p258~264，2006。
15. 張柞佳，「應用模擬退火法於軟體工作量預估之研究」，碩士論文，國立臺灣科技大學資訊管理研究所，2005。
16. 張銘仁，「運用模糊層分析法於軟體專案風險之研究」，碩士論文，國立屏東科技大學工業管理研究所，2004。
17. 張鴻林，「動態軟體工作量估算模式之研究」，碩士論文，國立臺灣科

- 技大學資訊管理研究所，2001。
18. 梁雅菁，「軟體開發委外計價流程模式」，碩士論文，國立臺灣科技大學資訊管理研究所，2003。
  19. 張紹勳、張紹評、林秀娟，「SPSS For Windows 統計分析—初等統計與高等統計(下冊)」，松崗電腦圖書公司，2002。
  20. 鄭惟厚 譯，Larry Gonick. & Woollcott Smith. 著，「看漫畫學統計」，天下文化，2004。
  21. 榮思哲，「以 WBS 為基礎之人力估算模型與功能點分析服務」，碩士論文，國立中央大學資訊工程研究所，2007。
  22. 錢一一 譯，Tom DeMarco, Timothy Lister 著，「與熊共舞：軟體專案的風險管理 (Waltzing with Bears-Managing Risk on Software Project)」，經濟新潮社，2004。
  23. 錢一一 譯，Frederick P. 著，「人月神話 (The Mythical Man-Month)」，經濟新潮社出版，2004。
  24. 藍元志，「專案風險管理回應策略選擇模式之建立」，國立中央大學工業管理研究所 碩士論文，2003。
  25. 羅應浮，「專案管理的失效模式與效應分析」，中華大學工業工程管理研究所 碩士論文，2000。

## 英文文獻

26. Albrecht, A. J., "Measuring application development productivity," In Proc. Of the Joint SHARE/GUIDE/IBM Application Development System, October 1979.
27. Basili, V. and Reiter, R., "An Investigation of Human Factors in Software Development," Computer, vol. 12, no.12, p21-38, 1979.
28. Boehm, Barry W., "Software Risk Management : Principles and Practices," IEEE Software, p32-41, January 1991.
29. Boehm, Barry W., "Software Engineering Economics," Prentice-Hall, Inc., 1981.
30. Conte, S. D., "Software Engineering Metrics and Models," Menlo Park CA:

- Benjamin/Cummings, 1986.
31. Cougar, J. and Adelsberger, H., "Comparing Motivation of Programmers and Analysts in Different Socio/Political Environments : Austria Compared to the United States," *Computer Personnel*, vol. 11, no.4, p13-17, 1988.
  32. DeMarco, T. and Lister, T., "Peopleware: Productive Projects and Teams," Dorset House, 1987.
  33. DOD(USA), "Department of Defense Handbook Work Breakdown Structure," Chapter 1, p4, 1998.
  34. Gautam Banerjee, "Use Case Point : An Estimation Approach," August 2001
  35. Fenton, N. and Pfleeger, S. "Software Metrics: A Rigorous & Practical Approach," PWS Publishing, 1997.
  36. Fowler, M., with Scott K., "UML Distilled - Applying the Standard Object Modeling Language," Addison Wesley Longman, 1997.
  37. Frederick P., "The Mythical Man-Month," Addison-Wesley, 1975.
  38. Fried, L. , "Team Size and Productivity in Systems Development," *Information Systems Management*, vol. 8, no.3, p27-41, 1991.
  39. IFPUG, "Function Point Counting Practices Manual Release 4.1," 2003.
  40. Jacobson, I., "Object-Oriented Software Engineering," Addison Wesley, 1992.
  41. Jorgensen M., Dyba, T., Kitchenham, B., "Teaching Evidence-Based Software Engineering to University Students," *Software Metrics*, 11th IEEE International Symposium. 2005.
  42. Karner, G., "Metrics for Objectory," Diploma thesis, University of Linköping, Sweden. No. LiTHIDA-Ex-9344:21, December 1993.
  43. Kitchenham, B., Jorgensen, M., Dyba, T., "Evidence-based Software Engineering," *International Conference on Software Engineering(ICSE)*, Edinburgh, p273-281, 2004.
  44. Linda Wallace, Mark Keil, Arum Rai, "Understanding software project risk : a cluster analysis," *Information & Management*, vol. 42, Issue 1, p115-125, 2004.
  45. Lynda M. Applegate, F. Warren McFarlan, James L. McKinney, "Corporate Information Systems Management: Text and Cases," Irwin; 5th Edition, p461-487, 1996.
  46. Matthew, L., "Risk management Manual Volume I, California: Merrit

- Company,” U.S.A., p1-2, 1983.
47. Newman, William H., Charles E. Summer, and E. Kirby. Warren, “The Process of Management: Concept, Behavior and Practice,” New York, NY: Prentice-Hall, 1972.
  48. Qing Hu, Robert T. Plant, David B. Hertz, “Software cost estimation using economic production models,” J. Management Information Systems, vol.15, no.1, p143-163, 1998.
  49. Randy K. Smith, “An Empirical Study Using Task Assignment Patterns to Improve the Accuracy of Software Effort Estimation,” Member, IEEE Computer Society, 2001.
  50. Reeser, C., “Management : Function and Modern Concepts,” New York, NY: Scott, Foresman and Company, 1973.
  51. Rosenberg, D. and Scott, K., “Use Case Driven Object Modeling with UML,” Addison Wesley, 1999.
  52. Sackett, D.L., Straus, S.E., Richardson, W.S., Rosenberg, W., and Haynes, R.B. “Evidence-Based Medicine: How to Practice and Teach EBM,” Second Edition, Churchill Livingstone: Edinburgh, 2000.
  53. Smith, J., “The Estimation of Effort Based on Use Cases,” Rational Software, White paper, 1999.
  54. Symons, C. R., John Wiley & Sons., “ Software Sizing and Estimating: Mk II FPA,” 1991.
  55. UKSMA, “Mk II Function Point Analysis - Counting Practices Manual,” 2002.
  56. Vinsen, K., Jamieson, D. and P.G. Callender, “Use Case Estimation –The Devil is in the Detail,” 2004.
  57. Wolverton, R. W., “The Cost of Developing Large-Scale Software,” IEEE Transactions on Computers, C-23, p615-636., Jun 1974.

## 附錄

### 附錄一 專案主要工作項訓練資料

編號	未調整 功能點	人員平均 年資	專案參與 人數	工作項時 程緊密度	人員並行 程度	時間離散 程度	風險影響 時程	人日
1	6	2.50	2	1.00	1.00	3.50	1.05	3
2	6	2.50	2	1.00	1.00	13.00	2.60	5
3	16	1.25	2	0.61	1.00	41.00	26.24	36
4	6	3.50	2	1.00	1.00	45.50	5.46	10
5	6	2.67	3	1.00	1.00	12.33	2.96	7
6	6	4.00	2	1.00	1.00	3.50	1.61	2.5
7	6	4.00	2	1.00	1.00	12.00	1.68	3
8	6	4.00	2	1.00	1.00	23.00	2.76	5
9	6	3.50	2	1.00	1.00	49.00	5.88	10.5
10	6	4.00	2	1.00	1.00	20.00	1.20	3.5
11	6	4.00	2	1.00	2.55	287.00	19.90	41.5
12	6	4.00	1	1.00	1.00	199.00	11.94	28
13	6	4.00	1	0.90	1.00	199.00	1.99	19.5
14	6	4.44	8	0.07	1.07	26.50	5.97	19.5
15	6	3.50	2	1.00	1.00	99.50	5.97	15.5
16	11	4.00	1	0.05	0.15	24.00	1.65	10
17	6	5.00	2	0.02	0.02	1.50	5.31	9
18	6	2.75	2	1.00	2.00	85.00	1.70	8
19	6	2.75	2	1.00	1.00	42.50	5.10	8.5
20	6	2.75	2	1.00	2.00	85.00	1.70	8
21	6	2.75	2	1.00	1.00	42.50	5.10	8.5
22	11	2.75	2	1.00	2.00	85.00	7.65	13
23	6	2.75	2	1.00	1.00	42.50	3.40	7
24	6	2.75	2	1.00	1.20	71.50	8.50	13
25	6	2.75	2	1.00	1.00	53.50	1.07	6
26	6	4.17	3	1.00	1.00	5.33	0.96	3.5
27	6	2.75	2	1.00	1.00	30.00	1.80	5
28	11	2.75	2	1.00	1.00	17.00	0.68	3.5
29	6	4.17	3	1.00	1.00	2.67	0.40	3
30	6	5.50	2	1.00	1.04	73.00	1.41	7
31	23	3.50	2	1.00	1.00	45.50	0.91	5
32	6	4.13	4	1.00	1.00	4.50	0.36	4

33	6	5.00	2	1.00	1.00	42.50	0.85	5
34	16	2.75	2	1.00	1.00	42.50	4.25	8
35	6	2.00	2	1.00	1.00	8.00	0.16	2.5
36	6	2.75	2	1.00	1.00	18.50	1.11	3.5
37	6	2.75	2	1.00	1.45	61.50	1.70	6.5
38	6	2.75	2	1.00	1.00	53.50	2.14	6.5
39	6	4.17	3	1.00	1.00	5.33	0.16	3
40	6	2.75	2	1.00	1.00	30.00	3.00	6
41	12	2.75	2	1.00	1.19	50.50	4.25	8
42	6	2.75	2	1.00	1.00	42.50	4.25	8
43	6	2.75	2	1.00	1.00	8.00	0.80	3
44	6	4.00	1	1.00	1.00	37.00	1.85	4
45	10	2.75	2	1.00	1.00	29.50	1.18	4.5
46	7	2.75	2	1.00	1.00	89.00	3.56	10
47	7	8.00	5	0.80	0.02	0.80	3.42	8
48	7	5.83	3	0.60	0.04	2.67	3.94	6.5
49	6	2.75	2	1.00	2.00	178.00	3.56	15
50	6	2.75	2	1.00	3.45	267.00	5.34	21.5
51	6	5.50	2	1.00	2.00	178.00	5.34	16
52	6	2.00	2	1.00	2.00	178.00	1.78	13.5
53	6	2.50	2	1.00	1.00	1.00	3.50	6

附錄二 本文預估模型檢測資料估算品質測量數據

筆數 編號	實際值	估算值 (檢測值)	誤差值  實際值-檢測值	相對誤差振幅  (估算值-實際值) /實際值	Within 20%	誤差值平方
1	3	2.3547	0.6453	0.2151	0	0.4164
2	4	4.7938	0.7938	0.1985	1	0.6302
3	30	35.0446	5.0446	0.1682	1	25.4482
4	11.5	10.0178	1.4823	0.1289	1	2.1971
5	7	6.2433	0.7567	0.1081	1	0.5726
6	3.5	2.3456	1.1545	0.3298	0	1.3328
7	3	3.0715	0.0715	0.0238	1	0.0051
8	7	5.1053	1.8947	0.2707	0	3.5899
9	11.5	10.7493	0.7507	0.0653	1	0.5635
10	5	3.1510	1.8490	0.3698	0	3.4187
11	25	40.1048	15.1048	0.6042	0	228.1561
12	24	27.5152	3.5152	0.1465	1	12.3565
13	18.5	16.5105	1.9895	0.1075	1	3.9582
14	20	17.1779	2.8221	0.1411	1	7.9642
15	20	14.7020	5.2980	0.2649	0	28.0687
16	7	6.6163	0.3837	0.0548	1	0.1472
17	13.5	9.6342	3.8658	0.2864	0	14.9445
18	8	7.0113	0.9887	0.1236	1	0.9775
19	15	9.7049	5.2951	0.3530	0	28.0377
20	7.5	7.0113	0.4887	0.0652	1	0.2388
21	11.5	9.7049	1.7951	0.1561	1	3.2222
22	17	13.9902	3.0098	0.1770	1	9.0590
23	11	7.8247	3.1753	0.2887	0	10.0823
24	15.5	14.1932	1.3068	0.0843	1	1.7076
25	5	6.0871	1.0871	0.2174	0	1.1817
26	5.5	2.8687	2.6313	0.4784	0	6.9237
27	5	5.1014	0.1014	0.0203	1	0.0103
28	5	3.2690	1.7310	0.3462	0	2.9965
29	3.5	2.0459	1.4541	0.4155	0	2.1145
30	9	6.7215	2.2785	0.2532	0	5.1914
31	5.5	5.2997	0.2003	0.0364	1	0.0401
32	5	3.3310	1.6690	0.3338	0	2.7856
33	6	5.0044	0.9956	0.1659	1	0.9911
34	12	9.5612	2.4388	0.2032	0	5.9476

35	2	1.6090	0.3911	0.1955	1	0.1529
36	5	3.4608	1.5392	0.3078	0	2.3691
37	8	6.4214	1.5786	0.1973	1	2.4919
38	5.5	7.2705	1.7705	0.3219	0	3.1346
39	2	1.9839	0.0161	0.0081	1	0.0003
40	7.5	6.4286	1.0714	0.1429	1	1.1479
41	13.5	8.9656	4.5344	0.3359	0	20.5604
42	8.5	8.7648	0.2648	0.0312	1	0.0701
43	2.5	2.3168	0.1832	0.0733	1	0.0336
44	5	3.9950	1.0050	0.2010	0	1.0099
45	5	4.3775	0.6225	0.1245	1	0.3875
46	14	11.6293	2.3707	0.1693	1	5.6203
47	11	10.7681	0.2319	0.0211	1	0.0538
48	12	9.0198	2.9802	0.2484	0	8.8818
49	19	16.1644	2.8357	0.1492	1	8.0409
50	35	22.7477	12.2523	0.3501	0	150.1181
51	14.5	16.9808	2.4808	0.1711	1	6.1543
52	16	14.5099	1.4901	0.0931	1	2.2203
平均相對誤差振幅						0.1989
相對誤差振幅(MRE)小於或等於 0.2 的個數占整體的比例						57.69%

附錄三 Randy K. Smith估算模型檢測資料估算品質測量數據

筆數 編號	實際值	估算值 (檢測值)	誤差值  實際值-檢測值	相對誤差振幅  (估算值-實際值) /實際值	Within 20%	誤差值平方
1	3	2.7395	0.2605	0.0868	1	0.0679
2	4	3.8320	0.1680	0.0420	1	0.0282
3	30	20.6194	9.3806	0.3127	0	87.9962
4	11.5	6.4195	5.0805	0.4418	0	25.8115
5	7	5.1367	1.8633	0.2662	0	3.4720
6	3.5	1.0145	2.4855	0.7101	0	6.1777
7	3	1.9920	1.0080	0.3360	0	1.0161
8	7	3.2570	3.7430	0.5347	0	14.0100
9	11.5	6.8220	4.6780	0.4068	0	21.8837
10	5	2.9120	2.0880	0.4176	0	4.3597
11	25	29.4618	4.4618	0.1785	1	19.9081
12	24	21.9240	2.0760	0.0865	1	4.3098
13	18.5	21.9240	3.4240	0.1851	1	11.7238
14	20	17.5378	2.4622	0.1231	1	6.0626
15	20	12.6295	7.3705	0.3685	0	54.3243
16	7	13.8624	6.8624	0.9803	0	47.0921
17	13.5	7.1529	6.3471	0.4702	0	40.2860
18	8	9.6195	1.6195	0.2024	0	2.6228
19	15	6.9370	8.0630	0.5375	0	65.0120
20	7.5	9.6195	2.1195	0.2826	0	4.4923
21	11.5	6.9370	4.5630	0.3968	0	20.8210
22	17	14.6195	2.3805	0.1400	1	5.6668
23	11	6.9370	4.0630	0.3694	0	16.5080
24	15.5	8.7674	6.7326	0.4344	0	45.3277
25	5	8.2020	3.2020	0.6404	0	10.2528
26	5.5	2.6067	2.8933	0.5261	0	8.3714
27	5	5.4995	0.4995	0.0999	1	0.2495
28	5	9.0045	4.0045	0.8009	0	16.0360
29	3.5	2.3000	1.2000	0.3429	0	1.4400
30	9	7.2038	1.7962	0.1996	1	3.2263
31	5.5	7.2820	1.7820	0.3240	0	3.1755
32	5	4.1318	0.8682	0.1737	1	0.7539
33	6	6.9370	0.9370	0.1562	1	0.8780
34	12	16.9370	4.9370	0.4114	0	24.3740

35	2	2.9695	0.9695	0.4848	0	0.9399
36	5	4.1770	0.8230	0.1646	1	0.6773
37	8	8.1362	0.1362	0.0170	1	0.0186
38	5.5	8.2020	2.7020	0.4913	0	7.3008
39	2	2.6067	0.6067	0.3033	0	0.3680
40	7.5	5.4995	2.0005	0.2667	0	4.0020
41	13.5	7.4419	6.0581	0.4487	0	36.7001
42	8.5	6.9370	1.5630	0.1839	1	2.4430
43	2.5	2.9695	0.4695	0.1878	1	0.2204
44	5	3.2940	1.7060	0.3412	0	2.9104
45	5	5.4420	0.4420	0.0884	1	0.1954
46	14	13.2845	0.7155	0.0511	1	0.5119
47	11	11.9371	0.9371	0.0852	1	0.8782
48	12	8.7208	3.2792	0.2733	0	10.7535
49	19	20.3145	1.3145	0.0692	1	1.7279
50	35	28.3445	6.6555	0.1902	1	44.2957
51	14.5	17.1520	2.6520	0.1829	1	7.0331
52	16	21.1770	5.1770	0.3236	0	26.8013
平均相對誤差振幅						0.3103
相對誤差振幅(MRE)小於或等於 0.2 的個數占整體的比例						40.38%