

國立臺東大學資訊管理學系

碩士論文

Department of Information Science and Management Systems

National Taitung University

Master Thesis

具三種故障內容函數特性並考慮不完美除錯之軟體可靠度成長模型  
General Software Reliability Model for Imperfect Debugging with Three  
Fault Content Functions.

研究生：謝濱壕 撰

Graduate Student : Pin-Hao Hsieh

指導教授：廖國良 博士

Advisor : Gwo-Liang Liao, Ph.D.

中華民國 108 年 6 月

July, 2019

國立臺東大學  
學位論文考試委員審定書  
系所別：資訊管理學系碩士班

本班 謝濱壕 君

所提之論文 具三種故障內容函數特性並考慮  
不完美除錯之軟體可靠度成長模型

業經本委員會通過合於 碩士學位論文 條件

論文學位考試委員會：

黃建裕

(學位考試委員會主席)

謝忠銘

蔣國良

(指導教授)

論文學位考試日期：108年6月3日

國立臺東大學

附註：1. 本表一式二份經學位考試委員會簽後，正本送交系所辦公室及註冊組或進修部存查。

2. 本表為日夜學制通用，請依個人學制分送教務處或進修部辦理。

# 國立臺東大學博、碩士學位論文授權書

本授權書所授權之論文為本人在 國立臺東大學 資訊管理學 系(所)

組 107 學年度第 2 學期取得 碩 士學位之論文。

論文名稱：具三種故障內容函數特性並考慮不完美除錯之軟體可靠度成長模型

本人具有著作財產權之論文全文資料，授權予下列單位：

同意	不同意	單位
<input checked="" type="checkbox"/>	<input type="checkbox"/>	國家圖書館
<input checked="" type="checkbox"/>	<input type="checkbox"/>	本人畢業學校圖書館
<input checked="" type="checkbox"/>	<input type="checkbox"/>	與本人畢業學校圖書館簽訂合作協議之資料庫業者

得不限地域、時間與次數以微縮、光碟或其他各種數位化方式重製後散布發行或上載網站，藉由網路傳輸，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

同意 不同意 本人畢業學校圖書館基於學術傳播之目的，在上述範圍內得再授權第三人進行資料重製。

## 電子檔公開時程

立即公開	一年後公開	二年後公開	三年後公開
			✓

本論文因申請專利或發表(未申請者本條款請不予理會)需延後公開，依「教育部100年7月1日臺高(二)字第1000108377號函」，延後公開合理期限最多不超過五年。

本論文已申請專利並檢附證明，專利申請案號：\_\_\_\_\_；

本論文準備以上列論文投稿期刊。

延後公開日期：民國\_\_\_\_\_年\_\_\_\_\_月\_\_\_\_\_日。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，本人同意視同同意授權。

指導教授姓名：廖國良 (親筆簽名)

研究生簽名：謝濱壕 (親筆正楷)

學 號：10601306 (務必填寫)

日 期：中華民國 108 年 6 月 18 日

1. 本授權書(得自 [http://portal.lib.nttu.edu.tw/lp.asp?CtNode=714&CtUnit=148&BaseDSD=52&mp=1&xq\\_xCat=10](http://portal.lib.nttu.edu.tw/lp.asp?CtNode=714&CtUnit=148&BaseDSD=52&mp=1&xq_xCat=10) 下載)請以黑筆撰寫並影印裝訂於書名頁之次頁。

2. 依據91學年度第一學期第一次教務會議決議：研究生畢業論文「至少需授權學校圖書館數位化，並遲於三年後上載網路供各界使用及校內瀏覽。」

授權書版本:2018/03/16

## 謝誌

兩年的研究所生涯就快結束了，回首過去，彷彿一切就像是昨天剛剛發生一樣。六年的時間過去了，我也從大學、研究所畢業，學到了許多事情，無論是課業，還是做人處事方面，師長與同儕一向都是良好的學習對象。

這篇論文能夠順利地完成，必須感謝指導教授廖國良老師的教導。從最一開始找不到研究方向到完成這篇論文，老師總在我不懂時耐心與細心地指導我，培養我思考及運算能力，在此感謝老師六年來的教導與良苦用心。感謝口試委員謝昆霖老師及黃建裕老師在百忙之中前來指導，口試前還花了許多時間與心力進行事前審查，並在口試當天給予諸多明確的建議，讓論文能夠更加的完善。

研究期間感謝哲宇、仁豪、維豪，當我遇到研究瓶頸時總能協助我度過，即使不懂也能陪我聊聊天解壓力。感謝班上同學大家的彼此鼓勵打氣，讓我能夠更加有力氣繼續做完研究。感謝多年來的好朋友們，感謝你們經常透過訊息聊天陪我抒發心情，有你們的支持鼓勵，讓我有動力盡快做完研究。感謝最重要、最親愛的家人，在我遇到瓶頸挫折懊惱時，透過電話關心我並給予我鼓勵，因為知道我的忙碌，時常叮囑我要好好休息，有你們給我的支持，讓我可以無後顧之憂地完成學業。

學生階段就快結束，透過這幾年的經驗累積，讓我更有勇氣面對接下來的關卡。最後以此篇論文獻給我的家人、老師、同學及這段日子以來幫助、關心我的人，謝謝您們！

謝濱壕 謹誌  
國立臺東大學資訊管理學系碩士班  
中華民國一〇八年六月

## 摘要

軟體與我們的生活息息相關，不論是在醫療、監控、發電廠、軍事和空中交通管制等方面，而隨著科技的發展以及進步，軟體的開發也更加快速，為了確保這些軟體的品質以及使其能夠正常運作，專家學者使用軟體可靠度預測軟體中的錯誤數，使軟體品質獲得極高的提升，也因此軟體可靠度越來越受到重視。本研究提出使用非齊次卜瓦松過程(Non-homogeneous Poisson Process)為基礎，並考量不完美除錯的可靠度成長模型(Software Reliability Growth Model)。當模型建構完畢之後，使用最小平方估計法求出所需之參數，並透過多組實際測試之數據，評估本研究提出之模型擬合度，再與過去專家學者所提出之模型相互比較，比較結果顯示，本研究所提出之模型在預測軟體錯誤數方面比現行的模型具有更高的準確性。

**關鍵字：**軟體可靠度成長模型、不完美除錯、非齊次卜瓦松過程

## Abstract

This paper has proposed a non-homogeneous Poisson process (NHPP) based software reliability growth model (SRGM) for imperfect debugging with three fault content functions. The development of technology makes software also have a fast growing, and those software are often related to our life, for example, in medical, surveillance, nuclear power station, real time military and air traffic control, etc. To ensure software quality the experts propose SRGMs to predict the failure behavior of systems that makes software reliability getting more and more importance. This paper has proposed a non-homogeneous Poisson process based SRGM and consider about the imperfect debugging. We use Least Squares Estimation to find the unknown parameter of the propose model, then use three known software failure data sets and compare with six existing model to measure the fitness of proposed model. Experimental results shows proposed model have better fit to the real failure data sets than existing SRGMs.

*Keywords: Software Reliability Growth Model; Non-Homogeneous Poisson Process; Imperfect Debugging*

## 目次

謝誌.....	i
摘要.....	ii
Abstract.....	iii
目次.....	iv
表目次.....	vi
圖目次.....	vii
第一章 緒論.....	1
第一節 研究背景與動機.....	1
第二節 研究目的與範圍.....	1
第三節 研究方法與流程.....	2
第二章 文獻探討.....	4
第一節 可靠度.....	4
第二節 硬體可靠度與軟體可靠度.....	6
第三節 卜瓦松過程與非齊次卜瓦松過程.....	11
第四節 非齊次卜瓦松過程模型.....	12
第五節 不完美除錯.....	19
第六節 學習曲線與團隊學習.....	21
第三章 研究方法.....	24
第一節 研究架構.....	24
第二節 資料來源.....	24
第三節 參數估計法.....	25
第四節 模型評估標準.....	26
第四章 模型建構.....	29
第一節 問題描述.....	29
第二節 軟體可靠度模型.....	29
第三節 資料分析與模型比較.....	31
第四節 小結.....	49

第五章 結論與建議.....	50
第一節 結論.....	50
第二節 建議.....	50
參考文獻.....	52
中文部分.....	52
英文部分.....	54



## 表目次

表 1	美國可靠度發展簡介.....	4
表 2	各國可靠度發展概要.....	5
表 3	NHPP 軟體可靠度模型與均值函數表.....	18
表 4	實際失效數據來源.....	25
表 5	資料集.....	36
表 6	模型參數估計 (資料集一).....	37
表 7	模型比較結果.....	40
表 8	模型參數估計 (資料集二).....	41
表 9	模型比較結果.....	44
表 10	模型參數估計 (資料集三).....	45
表 11	模型比較結果.....	48

## 圖目次

圖 1	研究流程圖 .....	3
圖 2	浴缸曲線圖 .....	7
圖 3	軟體失效與硬體失效之差異 .....	8
圖 4	兩種類型之除錯環境 .....	20
圖 5	學習曲線圖 .....	22
圖 6	PROPOSE 模型之適配度曲線 (資料集一) .....	38
圖 7	G-0 模型之適配度曲線 (資料集一) .....	38
圖 8	DSS 模型之適配度曲線 (資料集一) .....	38
圖 9	YEI 模型之適配度曲線 (資料集一) .....	39
圖 10	YLI 模型之適配度曲線 (資料集一) .....	39
圖 11	PNZ 模型之適配度曲線 (資料集一) .....	39
圖 12	ROY 模型之適配度曲線 (資料集一) .....	40
圖 13	本研究推出模型與現有模型之適配度比較圖 (資料集一) .....	40
圖 14	PROPOSE 模型之適配度曲線 (資料集二) .....	42
圖 15	G-0 模型之適配度曲線 (資料集二) .....	42
圖 16	DSS 模型之適配度曲線 (資料集二) .....	42
圖 17	YEI 模型之適配度曲線 (資料集二) .....	43
圖 18	YLI 模型之適配度曲線 (資料集二) .....	43
圖 19	PNZ 模型之適配度曲線 (資料集二) .....	43

圖 20	ROY 模型之適配度曲線 (資料集二)	44
圖 21	本研究推出模型與現有模型之適配度比較圖 (資料集二)	44
圖 22	PROPOSE 模型之適配度曲線 (資料集三)	46
圖 23	G-0 模型之適配度曲線 (資料集三)	46
圖 24	DSS 模型之適配度曲線 (資料集三)	46
圖 25	YEI 模型之適配度曲線 (資料集三)	47
圖 26	YLI 模型之適配度曲線 (資料集三)	47
圖 27	PNZ 模型之適配度曲線 (資料集三)	47
圖 28	ROY 模型之適配度曲線 (資料集三)	48
圖 29	本研究推出模型與現有模型之適配度比較圖 (資料集三)	48



# 第一章 緒論

## 第一節 研究背景與動機

在過去的幾十年間，軟體工程師以及相關領域的研究者們皆投入了大量的時間與精力，試圖以最少的錯誤創造出軟體。在最近幾年，快速發展的電腦設備以及通訊傳輸技術已經大大的改變了我們的生活。尤其在軟體開發的方面更為明顯，日新月異的技術使得軟體的數量急劇地增加，由於軟體經常在系統中運行，從而影響我們的日常生活，因此人們對於軟體的依賴程度日漸增高，例如在交通管制、核電廠、飛機、軍事、製造過程控制、汽車與安全控制以及醫院的病人監測系統等方面，就是因為軟體已經如此地貼近我們的生活，卻依然無法避免軟體出現錯誤，才更加彰顯了軟體可靠度的重要性。

在現代軟體開發的過程當中，品質的保證也變得越來越重要(Ibarra & Munoz, 2018)，因此使用軟體可靠度進行評估幾乎是不可避免的。而在過去的數十年當中，已經有眾多的專家學者提出了許多不同的軟體可靠度成長模型 (software reliability growth model ,SRGM) (Huang & Kuo, 2017)。其中又以非齊次卜瓦松過程 (Non-Homogeneous Poisson Process ,NHPP) 為基礎所建立的軟體可靠度成長模型為多數，即便已經有如此多的可靠度模型被提出來，只要軟體還有出錯的一天，就會持續有研究者為了提高軟體的品質與可靠度，進而提出更加複雜與精確的模型，因此本研究期望能夠藉由文獻的蒐集與探討，以期找出一個比現有模型更加可靠的軟體可靠度模型，並可作為軟體評估之參考。

## 第二節 研究目的與範圍

過去的學者所提出的軟體可靠度模型很多是以非齊次卜瓦松過程為基礎，並且假設錯誤會遵循著某種機率分佈，另外有一些學者是以不同的方法或者針對錯誤分佈進行不同假設來進行軟體可靠度的相關研究，例如：神經網路

(Lakshmanan & Ramasamy, 2015)、貝葉斯法 (Cai, 2000)、以及考慮非線性錯誤分布 (Wang & Wu, 2016) 等方法。關於錯誤移除方面的假設，最常見的是將檢測到的錯誤立即消除，這種假設稱之為完美除錯 (Perfect Debugging)，但是錯誤可能會在前一個錯誤被移除時出現，因此在實際的除錯過程中，這種假設並不合理，也不太可能發生。而另外一種假設，即是在除錯的過程中，可能產生新的錯誤，由於與前一種假設剛好相反，因此該假設程稱為不完美除錯 (Imperfect Debugging)。根據以上的文字，本研究希望藉由以非齊次卜瓦松過程為基礎，結合不完美除錯假設條件來建構可靠度模型。

早期的軟體可靠度研究大多都以單一概念以及簡單的假設為基礎，少有結合多種概念或因素的研究或模型，但近期的模型卻發現若模型當中結合多種其他模型，則可靠度會顯著提升，因此本研究希望以非齊次卜瓦松過程為基礎，並且結合三種不同模型當中的故障內容函數所形成之軟體可靠度成長模型，評估能否更準確地預測軟體失效行為。

### 第三節 研究方法與流程

本研究的模型以非齊次卜瓦松過程為基礎，結合另外三種模型的特性於故障內容函數中，並考量不完美除錯過程的軟體可靠度成長模型。當模型建構完成後，利用最小平方估計法求得模型所需參數值，並透過三組實際失效資料集，評估本研究的模型之適合度，並與過去專家學者提出之模型進行比較，評估本研究提出之模型在預測軟體失效行為上是否優於先前之模型。

本研究的流程一開始先針對問題找出研究動機及目的，接著蒐集並整理相關文獻，從中尋找如何解決問題。並從過去公開發表的論文當中蒐集適合資料集，模型建構則以 NHPP 模型為基礎，結合專家學者過去提出的模型特性於故障內容函數中，並考量不完美除錯過程。模型建構完成後，透過參數估計值與多組實際失效資料集進行測試，評估本研究提出之模型是否合適，若結果不合適則針對參

數估計這一項目進行重新試算與修正，直到符合所需的數值範圍為止，最後進行模型評估比較，並提出結論與建議。本研究流程如下圖所示。

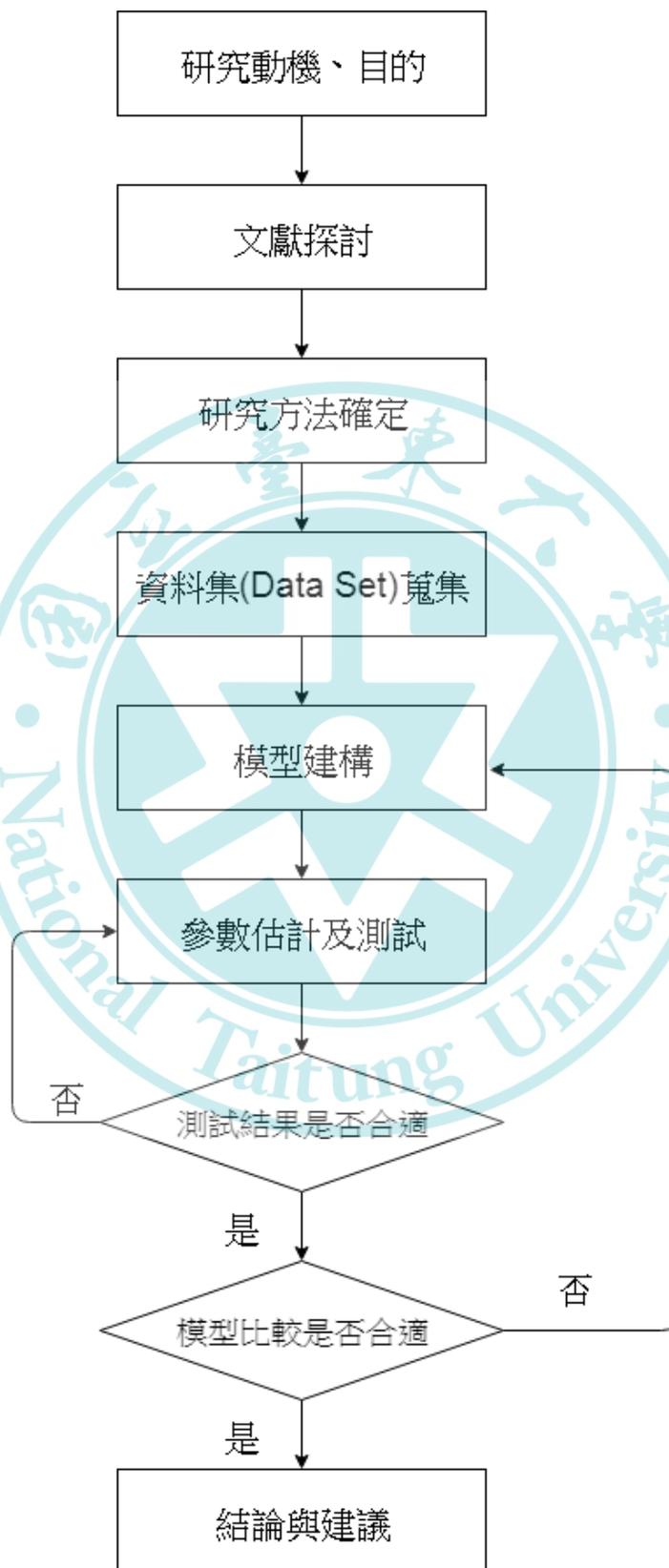


圖 1 研究流程圖

## 第二章 文獻探討

### 第一節 可靠度

可靠度 (Reliability) 這一概念最早可以追溯到第二次世界大戰時期，德國所製造的 V-1 火箭，以串聯模式計算可靠度與應用 (徐銘鴻，2008)，但是真正有組織性的活動則是由美國於二戰期間開始進行，逐步走向材料方面，由以下表格將美國的可靠度發展做個簡單的介紹。

表 1 美國可靠度發展簡介

階段	年份	組織	事件/功能/特點
萌芽時期	1943	電子管研究委員會	針對電子管的可靠度進行研究。
	1949	無線電工程師學會	成立可靠度技術組。
創建時期	1950	電子設備可靠度專門委員會	於 1952 年提出建議成立永久性可靠度委員會。
	1952	電子設備可靠度顧問組	即 AGREE，由美國國防部成立，並由軍方、工業界、學術界組成。
	1957	電子設備可靠度顧問組	發表著名報告《軍用電子設備的可靠性》。
全面發展時期	1965	-----	發布《系統與設備的可靠度大綱要求》。
	1966	羅姆航空發展中心	發布《維修性大綱要求》、《維修性鑑定、驗證及評估》、《維修性預計》等標準。
深入發展時期	1975-	-----	建立統一的可靠度管理機構。 成立全國統一的可靠性數據交換網。 改善可靠性設計和試驗方法。 展開軟體可靠度研究

資料來源：整理自 (楊志飛，可靠性概念講義)

除了美國以外，其他國家也大致於 20 世紀 70 年代以後開始進行或引進可靠度的相關研究，整理於表 2 之中。

表 2 其他國家可靠度發展概要

國家	年代	事件
蘇聯	1962	出版教科書《可靠性及質量控制的統計方法》。
	未知	建立可靠度組織機構與相關實驗室。 研究成果為 K-S 檢驗法與馬可夫過程。 大量引用美國軍用可靠度標準。
法國	未知	成立“電訊委員會”，用以協調、監督可靠度要求。
	1962	於國立電訊研究中心成立可靠度中心，負責各項可靠度事務。
日本	未知	從美國引進可靠度工程，並將軍事領域的研究成果應用至民用工業中。
	1965	成立“電子元件可靠度中心”，用以蒐集、分析元件的可靠度數據。
中華人民共和國	1955	“中國電子產品可靠性與環境試驗研究所”成立。
	1965	建立可靠度管理研究所。
	1984 ~	體系逐漸成形。

資料來源：整理自（楊志飛，可靠性概念講義）

現今有關產品的可靠度定義，一般多以美國國防部成立的顧問小組（Advisory Group on the Reliability of Electronic Equipment）所提出的報告《軍用電子設備的可靠性》當中的定義為基礎（陳登寶，2015），其中可靠度的定義為：「物品於既定的時間內，在特定的使用（環境）條件下，執行特定性能或功能，圓滿成功達成任務的機率」，由於大多數國家的可靠度工程幾乎都是自美國引進，再針對本土氣候、民情進行本土化，因此大多數國家的可靠度定義都會與美國相似，而當任務無法成功完成時，我們便稱之為失效(Failure)。

而除了理論方面的可靠度，也有許多人在進行硬體相關的可靠度研究，例如最近興起的電動車，就有許多的相關可靠度研究，不論是動力系統(楊宗展,2019)、鋰電池(王耀祥,2015)，也有結合機械安全，討論工具機安全系統的可靠度(林正平,2019)，如此多的論文，足以見得，可靠度已經不僅僅只是一種概念存在於理論當中，而是結合貼近人們的生活。

## 第二節 硬體可靠度與軟體可靠度

### 一、硬體可靠度

任何一個系統，都是由數個系統組成的，細分到最細就是元件，而只要是系統，就一定會提到可靠度，硬體可靠度的定義與概念是由可靠度延伸過來的，也就是在指定的操作環境與條件下，硬體在一段規定的時間內正確執行要求功能的能力。硬體失效的原因通常有元件損耗、元件疲勞、環境溫度、組裝工藝或是設計的問題，並且根據失效的時間長短可以大致分為兩類(Teng et al., 2006)：

#### (一) 永久性硬體失效：

即是硬體失效或退化，且復原不容易完成，當系統中的一個元件無法正常運作時，就是永久性故障。而永久性硬體故障又可以進一步分為以下兩類：

1. 總硬體失效：整體硬體元件停止運作，無論元件工作重做次數多寡。例如：汽車引擎燒毀，則汽車無法前進，直到引擎更換才能完成。
2. 局部硬體失效：部分硬體元件停止運作。例如：汽車是由許許多多的零部件所組成的，而且通常不會一起發生失效，假設汽車的某一部件發生失效，但汽車仍然可以運作，並完成任務，則稱此為局部硬體失效。

#### (二) 暫時性硬體失效：

指的是被操作環境的干擾所引起的失效，並且通常於重新啟動或任務重作後將失效移除。例如：引擎內部環境溫度超出可容許範圍，就有可能使引擎會暫時性的停止工作。而讓引擎休息一陣子之後，汽車又可以重新發動，

由於屬於高溫的操作環境干擾而引起的失效，並且能夠再重新啟動之後移除失效，因此可以算是暫時性硬體失效之一。

而與硬體可靠度的相關理論當中，另一知名的理論便是“浴缸理論”，此理論表示，某一產品其壽命中的失效會分為三個階段，亦即：早期失效、隨機失效、老化失效。

早期失效 (Decreasing Failure Rate)：初始階段失效率極高，主要原因大多是零件品質不良或輸送問題，導致失效率極高，而經過一段時間之後，由於零件逐漸磨合，將會使的失效率會隨時間增加而遞減。

隨機失效 (Constant Failure Rate)：此一階段失效率近乎一常數，而失效原因大多為不可預料的意外。

老化失效 (Increasing Failure Rate)：在此階段的失效率會隨時間的增加而遞增，主要原因大多為產品經過長時間使用，導致零件逐漸老化以及磨損進而導致失效率上升。



圖 2 浴缸曲線圖

## 二、軟體可靠度

軟體可靠度，即字面上的意思，是用來檢測軟體的可靠度，基本概念是由硬體可靠度發展而來的，軟體可靠度與硬體可靠度都屬於隨機過程，並且都可以藉由機率分配來進行解釋。而軟體與硬體最主要的差別便在於軟體並不會磨損、消耗與惡化...等物理層面的損耗，但是軟體中的某些部件的行為可能會影響可靠度（Chondamrongkul et al., 2019），也因此軟體的失效曲線並不會像硬體可靠度所產生的浴缸曲線圖，而是如圖 3 的虛線所示：

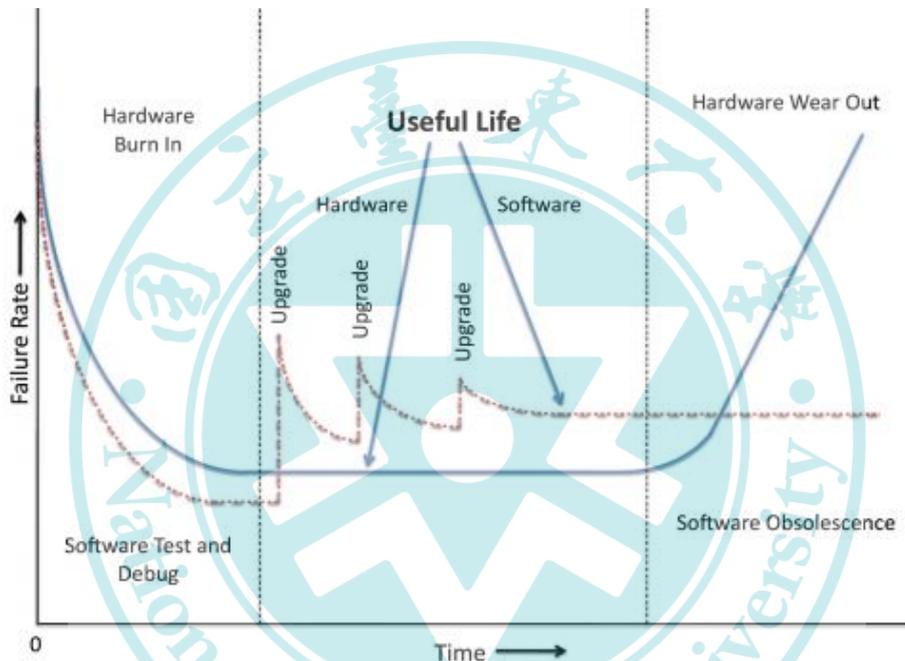


圖 3 軟體失效與硬體失效之差異

資料來源：William et al., (2012)

在圖 3 中我們可以明顯發現三階段的失效率呈現出很大的差異，並在下方說明：

發售前：

軟體失效率：最高，但經過除錯而隨時間降低。

硬體失效率：最高，但經過燒機而隨著時間下降至最低。

發售後：

軟體失效率：隨著每次的更新，軟體複雜度也會提升，連帶使失效率短暫上升，並隨時間降低，直到下次更新，如此形成一個循環。

硬體失效率：由於經過燒機使得失效率在上市之後幾乎為一常數。

產品生命週期結束：

軟體失效率：由於生命週期結束，軟體不再更新與除錯，因此不會有所上升或下降，並在最後趨近於一常數，直到此產品完全淡出人們的眼中，但仍會比剛上市的時候還要高上許多。

硬體失效率：在生命週期的最後，由於硬體的磨損與老化導致失效率會急速上升，直到報廢為止。

而軟體會發生失效的原因，幾乎就是因為程式碼，由於程式碼的增加會導致新的錯誤產生，換句話說，錯誤數會隨著程式碼的行數呈現指數的增加（謝爾廉, 2000）。也就表示，在一次次的更新之後，將使得程式碼的行數也跟著變多，導致能夠找到的錯誤數也變多，除了錯誤變多，程式碼過於複雜也會對軟體開發造成影響（Eisty et al., 2018）。而關於程式碼會出錯的原因，不外乎是邏輯的錯誤、人工輸入的錯誤又或者是對於功能的錯誤描述，再加上要發現失效或錯誤，還得要等到執行到該行程式碼才會被發現，而如果沒有執行到，就會一直存在於軟體之中，且可能都不會被發現。

就算把所有的程式碼都跑過一次，在除錯的過程當中，也會因為程式碼的變動而有新的錯誤產生，因此整個開發流程會在除錯與測試的階段花上大把大把的時間，導致軟體遲遲無法上市販售，因此測試的時間也必須視為成本，此時軟體可靠度的重要功能就表現出來了，那就是估計軟體內殘留的錯誤數，並與開發日程互相配合，進而將三者盡量持平的時間點上市，才能夠為軟體公司節省成本，賺進利潤，畢竟軟體開發的成本是非常高的，需要盡量節省成本，上市之後才能夠讓公司有足夠的利潤，讓軟體能夠進行維護與發佈後續更新。

而軟體可靠度模型可從本質上區分為兩種（Wood, 1996；李嘉華，2014）：

1. 從設計參數中預測軟體可靠度（又稱缺陷密度，Defect Density）模型，可以從程式碼特性、行數、輸入和輸出的資料等參考資料計算出軟體內的錯誤數目。
2. 從測試資料預設軟體可靠度（又稱軟體可靠度成長模型，SRGM），模型試圖透過具有一定特性的已知函數與過去實際失效數據來預測未來軟體失效行為。

另外 SRGM 根據以下假設成立（Huang & Lin, 2006）：

1. 故障偵測過程遵循非齊次卜瓦松過程（NHPP）。
2. 在任何時間中軟體系統失效為系統中的剩餘故障所引起。
3. 所有偵測到的錯誤可以被歸為前置錯誤（Leading Faults）或相依錯誤（Dependent Faults），而錯誤總數是有限的。
4. 在時間間隔中偵測到的前置錯誤平均數與系統中的剩餘前置錯誤平均數成正比。此外，該比例會隨時間而變動。
5. 在時間間隔中偵測到的相依錯誤平均數與系統中的剩餘相依錯誤平均數成正比，及系統時間內總錯誤數中的前置錯誤移除率。該比例會隨時間而變動。
6. 偵測到的相依故障可能不會立即移除，且會延後至錯誤偵測過程。
7. 在錯誤移除過程中沒有新的錯誤產生。

目前現有的軟體可靠度模型，大多為觀察軟體產品在測試階段中的失效次數，也就是 SRGM，其中又以 NHPP 為基礎建立之 SRGM 較符合實際的情況。而最早將此概念運用至 SRGM 的專家學者為 Goel 和 Okumoto（Goel & Okumoto, 1979），後續則有 Ohba（Mitsuru Ohba, 1984）、Yamada 和 Osaki（Yamada & Osaki, 1985）及 Pham 和 Zhang（Pham & Zhang, 1997）等學者延續此概念並陸續提出不同的軟體可靠度成長模型，進行軟體測試過程中的失效數據分析。因此，本研究將針對此種 SRGM 進行探討。

### 第三節 卜瓦松過程與非齊次卜瓦松過程

#### 一、卜瓦松過程 (Poisson Process)

卜瓦松過程是隨機過程的一種，是以事件的發生時間來定義的，也是所有計數過程最重要的基礎理論，必須滿足以下條件，方可稱為卜瓦松過程 (游卓群，2012；高婉蕙，2012；朱家儀，2013；祝英哲，2013)：

1.  $P[N(t_0) = 0] = 1$ ，初始時間點發生 0 次事件的機率為 1， $t_0 = 0$ 。
2.  $\{N(t) ; t \geq 0\}$  具有獨立增量，即不同的時間區間內，發生事件的機率為互相獨立的。
3.  $P[N(t + \Delta t) - N(t) = 1 | H_t] = \lambda(t)\Delta t + o(\Delta t)$ ， $H_t$  為時間  $t$  之前所發生的事件。
4.  $P[N(t + \Delta t) - N(t) \geq 2] = o(\Delta t)$ ， $o(\Delta t)$  滿足  $\lim_{\Delta t \rightarrow 0} \left\{ \frac{o(\Delta t)}{\Delta t} \right\} = 0$ ，表示在極短的時間以內發生兩次或兩次以上事件的機率幾乎為零。

#### 二、非齊次卜瓦松過程 (NHPP)

非齊次卜瓦松過程，又可稱為非均質性卜瓦松，其概念與卜瓦松過程非常類似，同樣需要滿足某些條件。但是卜瓦松過程是假設累積事件個數， $N(t)$  則是一個計數過程 (Counting Process)，並且平均失效率為一常數  $\lambda$ ；而非齊次卜瓦松過程則考慮到軟體失效率函數是一個與時間相關的函數，也就是失效率函數為  $\lambda(t)$  (江文馨，2013；王亭雅，2015)。另外非齊次卜瓦松過程為無記憶性，而強度函數  $\lambda(t)$  會隨著時間  $t$  而改變，即  $\lambda(t)$  為時間  $t$  之函數。

當滿足下列四點條件時即為非齊次卜瓦松過程 (林宗煥，2012；朱家儀，2013；祝英哲，2013；謝爾廉，2000；Pham, 2006；李嘉華，2014)：

1.  $P[N(t_0) = 0] = 1$ ，初始時間點發生 0 次事件的機率為 1， $t_0 = 0$ 。
2. 失效過程具有獨立增量，在兩個不重疊的區間中的失效數目為互相獨立的隨機變數。
3.  $P[N(t + \Delta t) - N(t) = 1] = \lambda(t)\Delta t + o(\Delta t)$

$$4. P[N(t + \Delta t) - N(t) \geq 2] = o(\Delta t)$$

#### 第四節 非齊次卜瓦松過程模型

基於非齊次卜瓦松過程模型 (NHPP) 的軟體可靠度成長模型 (SRGM) 能夠用來估計有用的指標，例如：剩餘錯誤數、故障強度等 (Nagaraju et al., 2017)，雖然輸入參數這一做法仍然存在著某種程度的不確定性，但仍舊是一種有效的評估方式 (Pietrantuono et al., 2018)。而在上一小節提到過非齊次卜瓦松過程的 $\lambda(t)$ 為失效率函數，而其 $N(t)$ 則是假設為累積之前所偵測到的錯誤數，並且屬於一計數過程，而 $N(t)$ 的期望偵測錯誤數則為均值函數 $m(t)$ (Mean Value Function, MVF)，根據 NHPP 即可求得在時間 $t$ 時，能夠找到的錯誤數 (謝爾廉，2000)。

而 NHPP 模型的基本假設與上一小節當中 NHPP 的假設一樣，因此於此不再贅述。

在 NHPP 當中，假設累積失效個數為 $N(t)$ ，其計數過程 $\{N(t), t \geq 0\}$ 可以推導出 (Huang, Lyu & Kuo, 2003)：

$$\Pr\{N(t) = k\} = \frac{(m(t))^k}{k!} e^{-m(t)}, k = 0, 1, 2, \dots \quad (2.1)$$

以及

$$m(t) = \int_0^t \lambda(x) dx \quad (2.1)$$

而以 NHPP 為基礎的模型可以分為以下四種類型 (Pham, 2000)：

1. NHPP 指數型 (Exponential) 模型
2. NHPP S 型 (S-Shaped) 模型
3. NHPP 不完美除錯模型
4. NHPP 不完美除錯 S 型模型

由於 NHPP 模型假設失效強度函數 $\lambda(t)$ 與軟體內所剩餘的錯誤數成正比，並且可以透過微分以下方程式可以求得 (Pham et al., 1999 ; Roy et al., 2014)：

$$\frac{dm(t)}{dt} = b(t)[a(t) - m(t)] \quad (2.3)$$

其中  $a(t)$ =故障內容函數，即在時間  $t$  時在軟體內的總錯誤數，包含了初始以及之後所產生的錯誤。

$b(t)$ =在時間  $t$  時的故障偵測率函數

$m(t)$ =均值函數，即在時間  $t$  時期望偵測的錯誤數。

透過給予  $a(t)$ 跟  $b(t)$ 不同的數值，可以得到不同的 NHPP 模型，將在下列詳細解釋。

## 一、 NHPP 指數型模型

### (一) Goel-Okumoto 模型

Goel-Okumoto (G-O)模型是由 Amrit Goel 與 Kazuhira Okumoto 於 1979 年提出，此模型是最常見的軟體可靠度，在相關論文當中常常可以看見，也可以其為基礎，延伸出新的模型，此模型的基本假設如下：

- (1) 於時間  $t$  偵測到的累積錯誤數遵循卜瓦松分配 (Poisson Distribution)。
- (2) 當偵測到錯誤時，錯誤會立即被移除，且不會產生新的錯誤，亦即完美除錯。
- (3) 所有錯誤各自獨立，被偵測到的機率相同。
- (4) 錯誤的發生機率為恆定的。

假設  $a(t) = a$  及  $b(t) = b$ ，且邊際條件  $m(0) = 0$ ，可以得到：

$$m(t) = a(1 - e^{-bt}) \quad (2.4)$$

## 二、 NHPP S 型模型

在 NHPP S 型模型當中，由於根據以下的假設，導致軟體可靠度的生長曲線為 S 型（洪翠雲，2007）：

1. 造成 S 型是因為有一部分錯誤被其他的錯誤隱藏，因此在偵測到錯誤並移除之前不會被發現，導致在偵測過程中不會被發現。

2. 由於軟體測試屬於一個學習過程，測試技巧的不成熟及不熟悉的測試項目直接導致錯誤偵測的成效不彰。

#### (一)NHPP 延遲 S 型(Delayed S-shaped, DSS)模型

DSS 模型是由 Yamada 等人考慮到測試效率，導致軟體失效曲線變形為 S 型，此種模型有以下數點假設（Yamada et al., 1983）：

1. 在任何時間下，軟體的失效行為皆為軟體中的錯誤造成的。
2. 所有的錯誤互相獨立，偵測到的機率也相同。
3. 故障內容函數最初為一隨機變數。
4. 第  $(n-1)$  與第  $n$  次失效發生的間隔取決於第  $(n-1)$  次失效發生時間。
5. 當偵測到錯誤時，錯誤會立即被移除，且不會有新的錯誤產生。

假設  $a(t) = a$  及  $b(t) = (b^2t)/(bt + 1)$ ，且邊際條件  $m(0) = 0$ ，可以得到：

$$m(t) = a(1 - (1 + bt)e^{-bt}) \quad (2.5)$$

#### (二)NHPP 曲折 S 型(Inflection S-shaped, ISS)模型

ISS 模型是由 Ohba 所提出，由於觀察到的軟體可靠度的曲線為 S 型而提出，有以下基本假設（Huang & Lin, 2006）：

1. 軟體中的錯誤相互依賴。
2. 部分錯誤被移除前，部分故障是無法被檢測到的。
3. 任何時間的錯誤偵測率與軟體偵測到的錯誤數成正比。
4. 被偵測到的錯誤失效率相同且為常數。
5. 錯誤可以完全被移除。

假設  $a(t) = a$  及  $b(t) = b/(1 + \beta e^{-bt})$ ，且邊際條件  $m(0) = 0$ ，可以得到：

$$m(t) = a(1 - e^{-bt})/(1 + \beta e^{-bt}) \quad (2.6)$$

### 三、 NHPP Yamada 不完美除錯模型

#### (一) Yamada 不完美除錯模型 1

Yamada 不完美除錯模型 1 又可以稱為(Yamada Exponential Imperfect Debugging Model, YEI Model)由 Yamada 於 1992 年提出，該模型具有以下假設 (Yamada, 1992)：

1. 當偵測到錯誤被移除時，會有新錯誤產生的可能。
2. 故障內容函數為一指數型函數，而錯誤偵測率則為一常數。
3. 錯誤偵測率與剩餘的錯誤數成比例。

假設  $a(t) = ae^{at}$  及  $b(t) = b$ ，且邊際條件  $m(0) = 0$ ，可以得到：

$$m(t) = \frac{ab}{a+b}(e^{at} - e^{-bt}) \quad (2.7)$$

#### (二) Yamada 不完美除錯模型 2

Yamada 不完美除錯模型 2 又可以稱為(Yamada Linear Imperfect Debugging Model, YLI Model)同樣由 Yamada 於 1992 年提出，該模型具有以下假設 (Yamada, 1992)：

1. 當偵測到錯誤被移除時，會有新錯誤產生的可能。
2. 錯誤產生率與偵測率皆為一常數。
3. 錯誤偵測率與剩餘的錯誤數成比例。

假設  $a(t) = a(1 + at)$  及  $b(t) = b$ ，且邊際條件  $m(0) = 0$ ，可以得到：

$$m(t) = a(1 - e^{-bt})\left(1 - \frac{a}{b}\right) + aat \quad (2.8)$$

### 四、 NHPP S 型不完美除錯模型

#### (一) Pham-Zhang NHPP 模型

Pham-Zhang(P-Z)模型由 Hoang Pham 與 Xuemei Zhang 提出，模型基本假設如下 (Pham & Zhang, 1997)：

1.  $a(t)$ 反應出不完美除錯的現象。
2.  $b(t)$ 是一個具有曲折的非遞減函數。

假設  $a(t) = c + a(1 - e^{-\alpha t})$  及  $b(t) = b/(1 + \beta e^{-bt})$ ，且邊際條件  $m(0) = 0$ ，可以得到：

$$m(t) = \left( (c + a)(1 - e^{-bt}) - \frac{ab}{b-a}(e^{-\alpha t} - e^{-bt}) \right) / (1 + \beta e^{-bt}) \quad (2.9)$$

## (二)Pham-Nordmann-Zhang NHPP 模型

Pham-Nordmann-Zhang(P-N-Z)模型基本假設如下 (Pham et al., 2006)：

1. 故障內容函數為一線性函數
2.  $b(t)$ 是一個非遞減的 S 型模型。

假設  $a(t) = a(1 + \alpha t)$  及  $b(t) = b/(1 + \beta e^{-bt})$ ，且邊際條件  $m(0) = 0$ ，可以得到：

$$m(t) = \frac{a}{1 + \beta e^{-bt}} \left[ (1 - e^{-bt}) \left( 1 - \frac{\alpha}{b} \right) + \alpha t \right] \quad (2.10)$$

## 五、考慮 NHPP 與錯誤生成率的不完美除錯模型

### (一) Roy NHPP 模型

Roy 模型是由 Roy 等人所提出，模型基本假設如下 (Roy et al., 2014)：

1. 錯誤的偵測與移除過程遵循 NHPP。
2. 軟體中剩餘的錯誤導致軟體在執行期間出現新的錯誤。
3. 測試過程中的錯誤檢測率是恆定的。
4. 軟體的錯誤率與剩餘錯誤成正比。
5. 當偵測到的錯誤被刪除時，有可能產生新的錯誤。
6. 在測試時可能會產生錯誤，並且在錯誤移除時可能會產生新的錯誤，亦即不完美除錯。

7. 在測試開始時，測試團隊對於軟體並不熟悉，也沒有如何正確除錯的相關知識。隨著時間過去，團隊也獲得了越來越多關於除錯的知識。
8. 故障內容函數為測試時間的指數增加函數。
9. 測試開始時的錯誤產生率高於結束時的錯誤產生率。

令 $a(t) = a(\alpha - e^{-\beta t})$  及  $b(t) = b$ ，且邊際條件 $m(0) = 0$ ，可以得到：

$$m(t) = a\alpha(1 - e^{-\beta t}) - ab/(b - \beta)(e^{-\beta t} - e^{-bt}) \quad (2.11)$$



表 3 NHPP 軟體可靠度模型與均值函數表

模型名稱	均值函數 $m(t)$
Goel-Okumoto Model	$m(t)=a(1-e^{-bt})$ $a(t)=a$ $b(t)=b$
Inflection S-shaped Model	$m(t)=a(1-e^{-bt})/(1+\beta e^{-bt})$ $a(t)=a$ $b(t)=b/(1+\beta e^{-bt})$
Delayed S-shaped Model	$m(t)=a(1-(1+bt)e^{-bt})$ $a(t)=a$ $b(t)=(b^2t)/(bt+1)$
Yamada Exponential Imperfect Debugging Model	$m(t)=(ab/(\alpha+b))(e^{\alpha t}-e^{-bt})$ $a(t)=ae^{\alpha t}$ $b(t)=b$
Yamada Linear Imperfect Debugging Model	$m(t)=a(1-e^{-bt})(1-\alpha/b)+a\alpha t$ $a(t)=a(1+\alpha t)$ $b(t)=b$
Pham Exponential Imperfect Debugging Model	$m(t)=(ab/(b+\beta))((e^{(b+\beta)t})/(e^{bt}+c))$ $a(t)=ae^{bt}$ $b(t)=b/(1+ce^{-bt})$
Pham-Nordmann-Zhang Model	$m(t)=a[(1-e^{-bt})(1-\alpha/b)+\alpha t]/1+\beta e^{-bt}$ $a(t)=a(1+\alpha t)$ $b(t)=b/(1+\beta e^{-bt})$
Pham-Zhang Model	$m(t)=[(a+c)(1-e^{-bt})-(bc/(b-a))(e^{-\alpha t}-e^{-bt})]/(1+\beta e^{-bt})$ $a(t)=c+a(1-e^{-\alpha t})$ $b(t)=b/(1+\beta e^{-bt})$
Yamada exponential	$m(t)=a(1-e^{-\gamma\alpha(1-e^{-\beta t})})$ $a(t)=a$ $b(t)=\gamma\alpha\beta e^{-\beta t}$
Roy Model	$m(t)=a\alpha(1-e^{-bt})-ab/(b-\beta)(e^{-\beta t}-e^{-bt})$ $a(t)=a(\alpha-e^{-\beta t})$ $b(t)=b$

資料來源：Pham 與 Zhang (2003)

## 第五節 不完美除錯

### 一、 不完美除錯

早期的研究大多假設除錯的過程為完美除錯 (Perfect Debugging)，也就是偵測到的錯誤會立即被移除，並且在移除錯誤的過程中不會有產生新錯誤的可能性，但在實際測試的過程中，可能會因軟體開發人員對程式碼進行的修正，導致軟體產生一個或多個新的錯誤，所以此假設於現實較不合理。

而後續的研究大多假設除錯的過程為不完美除錯 (Imperfect Debugging)，此一概念於 1987 年被 Musa, Iannino 與 Okumoto 等人提及，形成此種假設的原因大多是因為軟體測試人員在除錯時變動了程式碼而產生新的錯誤。一般的軟體測試過程為錯誤產生時，找出錯誤並以相對應的方法移除錯誤，但由於是不完美除錯的緣故，因此錯誤沒有辦法百分之百的被移除，因此我們可以很明確的知道，當測試時間縮短時，在軟體當中剩餘的錯誤也會增多，因此後續除錯成本也會增加。近年來也有研究者為了減少測試人員或開發人員掌握錯誤報告所需要的時間 (Jiang et al, 2019) 而進行相關的研究。

軟體測試的目的就是為了偵測和修正全部隱藏在軟體系統內的錯誤，但即使測試人員充分的規劃，並且使用不同除錯方法與技術，仍然不能保證軟體在發佈後不會出現錯誤，而軟體發佈後存在於軟體中的錯誤可以分為以下三類 (Kapur et al, 2011)：

1. 整個測試期間都未測試到的錯誤。
2. 整個測試期間有被偵測到，但尚未完全被移除及修正。
3. 整個測試期間修正及移除中產生的新錯誤。

第一類錯誤可以藉由測試覆蓋率及其他方法提升測試效率，進而改善，但礙於時間及資源的限制，仍然無法完全移除錯誤；第二、三類錯誤涉及測試團隊的經驗及效率，熟練的團隊可以提供一個更好的測試方法以減少錯誤量。在軟體可靠度文獻中，測試團隊的經驗效率被納入不完美除錯的軟體可靠度模型之一，分

別為不完美錯誤移除(Imperfect Fault Debugging)及錯誤產生(Fault Generation)，如圖 4 所示。

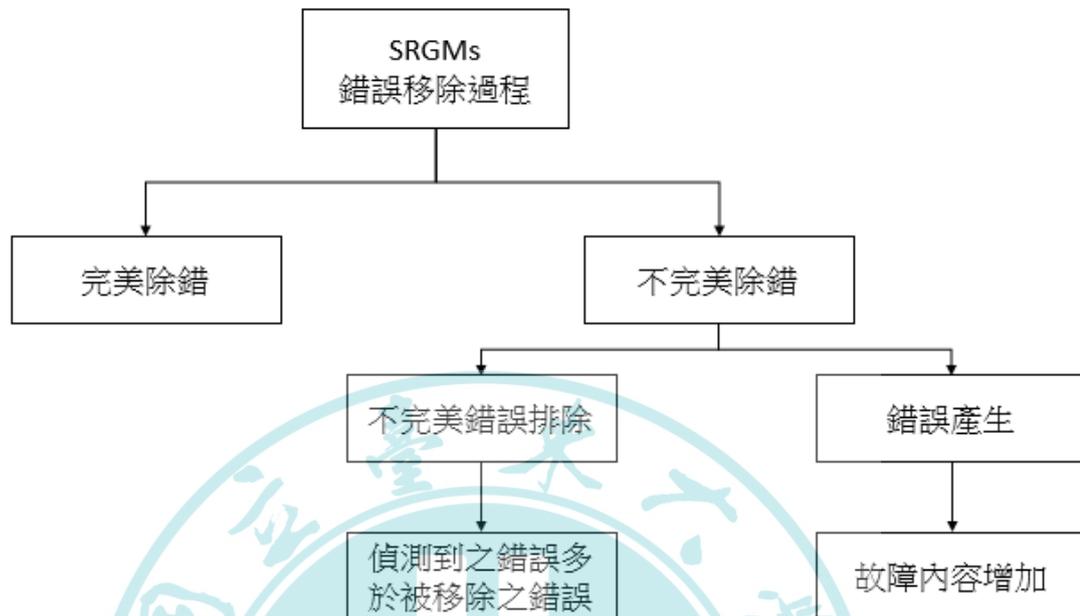


圖 4 兩種類型之除錯環境

資料來源：Kapur et al, (2011)

而在測試軟體的時候，各種各樣的因素都會影響到測試階段的可靠度，環境 (Li & Pham, 1997)、效率、成本，皆有可能成為主要因素。與 SRGM 相關的早期研究大多將除錯假設為每次除錯都能夠成功移除錯誤，並且不會產生新的錯誤，但是此種假設在現實中較難發生，是由於每次的除錯都有可能產生或造成新的錯誤，因此近些年來開始有學者以不完美除錯為假設的 SRGM 進行研究，例如：Yamada、Pham、..... 等人，使得軟體可靠度能夠更有效地應用於實際情形，並做出更加可靠的預測。

## 第六節 學習曲線與團隊學習

由於過去十年來的研究多以指數型 SRGM 占了大多數，而其中也會連帶提到關於學習效果的概念，主要是因為當測試時間增加，測試或開發人員對於整體熟悉度與經驗也會跟著增加，並有助於減少偵測與移除錯誤的時間，使除錯率與可靠度隨著增加。而近年來的 SRGM 相關研究當中也逐漸有將測試團隊當作影響可靠度的因素之一的趨勢，最主要的原因是軟體的規模越來越龐大，僅靠一、兩個人無法完整測試，但是一旦組成團隊，人多嘴雜的情形之下，測試的情形勢必會與原本的情況完全相反。因此本節除了探討學習曲線之外，也將探討何謂團隊學習。

### 一、學習曲線 (Learning Curve)

學習理論是由心理學家發展出來的，心理學將學習定義為「因經驗而使行為或行為潛意識產生較持久改變之歷程」(張春興，2001)，而其研究方向中便有探討學習時間相對於成就之間的關係。

此種理論應用於實務之中，則能夠得出生產的單位時間與連續生產單位之間的關係。而當個體或組織在一項任務中學習到更多的經驗，會使得他們效率提升(江姮臻，2014)。此種理論由 Ebbinghaus 於 1885 提出，而由 Wright 應用於觀察飛機製造過程，Wright 的研究指出，隨著產品數量的累積，生產成本會以一定的百分比下降 (Wright, 1936)。而將累積產量相對於生產時間繪成圖形，以生產時間為縱軸 (Y)，以生產量為橫軸 (X)，其所得之圖形稱為「學習曲線 (Learning Curve)」，如圖 6 所示，亦可稱為「經驗曲線 (Experience Curve)」或稱「改善曲線(Improvement Curve)」。這種「學習」的情形相當於績效與生產力的進步 (邱人文，2013)。

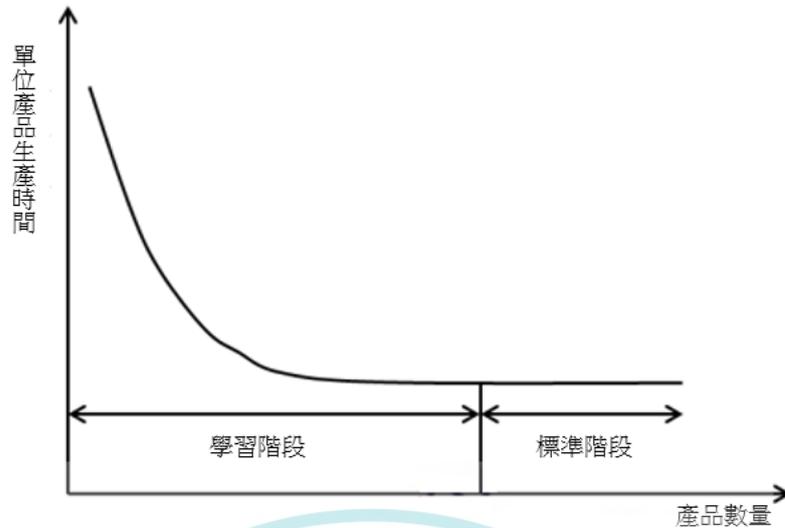


圖 5 學習曲線圖

資料來源：陳世坤（2005）

生產作業之績效會隨著作業天數之增加而有所變化，這種變化稱之為學習效應。也就是說：越是經常地執行一項任務，每次所需的時間就越少（江姮臻，2014）。而學習效應會隨著經驗累積、技術進步、規模效應而有所差異（陳之薇，2009）。

## 二、 團隊學習

團隊的意思是指為了實現某一目標而由相互協作的個體所組成的正式群體。有的學者提出團體(Group)不等於團隊(Team)，並指出團體是不需要共同努力的集體工作（李茂興，2001），而有的學者則是認為，「一個團體成員傾向互相合作交流」是由群體變成團隊的重要關鍵（Hambrick, 1994）。而團體和團隊最大的差別在於，團體內個人工作表現的總合即為團體表現的總合；團隊則是經過資訊分享和制訂決策，並協調成員，努力產生正向的共同效果，團隊內的個人努力之結果經整合之後，往往會高於個人付出的總合。

而 Nadler（Nadler et al, 1998）等人則認為組成團隊進行工作，可以產生以下的功能：集思廣益、組織效能提升、成員認同感、工作動機、擴展視野、創意發想、危機分擔、經驗傳承、及得到更多的社會性支持。也就是說，團隊能夠呈現團結一致的氣氛以及積極的行動力，是使組織在環境中取得平衡與成長的關鍵。

團隊學習是團隊成員經由分享而產生的集體知識以及技術，能夠造成團隊持久性的改變，並在知識與技能上有顯著的整體性成長。也就是說，在團隊學習的過程當中包含成員彼此雙向的知識交換（mutually exchange their knowledge），以及共同創造新知識（Ellis et al., 2003; Hoof & Ridder, 2004）。因此，從交換理論（Theory of social exchange）的觀點出發，人與人的互動會計算得失，所以團隊學習也可視為一種交換的行為，此種交換行為是維持社會秩序的基礎之一（宋鎮照，1997）；團隊成員衡量彼此的報酬與付出，以此決定是否願意分享經驗與知識，當交換關係具有吸引力時才會與對方有持續的互動（田靜婷，2003）。

團隊學習的本質是成員將其所擁有的知識擴散傳播給他人。但若是團隊學習並無實質上的知識分享，團隊的討論就會受限於原有的知識，並無法產生學習行為，導致團隊學習情況停滯不前（Liao, 2008; Stasser & Titus, 1987）。

由於過去十年來的研究多以指數型 SRGM 占了大多數，而其中也會連帶提到關於學習效果的概念，主要是因為當測試時間增加，測試或開發人員對於整體熟悉度與經驗也會跟著增加，並有助於減少偵測與移除錯誤的時間，使除錯率與可靠度隨著增加。而近年來的 SRGM 相關研究當中也逐漸有種趨勢將測試團隊當作影響可靠度的因素之一，最主要的原因是軟體的規模越來越龐大，僅靠一、兩個人無法完整測試，但是一旦組成團隊，人多嘴雜的情形之下，測試的情形勢必會與原先的情況完全相反。因此本節除了探討學習曲線之外，也將探討何謂團隊學習。

## 第三章 研究方法

### 第一節 研究架構

本研究之架構如圖所示，首先透過文獻探討了解過去學者所提出的軟體可靠度成長模型及其特性，並運用部分模型之參數作為本研究模型參考依據，再進行模型建構。模型建構完成後，針對模型進行評估，並且檢查是否有錯誤，完成之後，再利用實際失效數據與參數估計法針對模型當中的未知參數進行估計，並反覆調適參數，直至模型與失效數據之間的適配度（Goodness of Fit）不會差異過大。最後再將本研究所提出的模型與現有模型進行評估標準的比較，觀察本研究所提出的模型是否能更有效的預測軟體失效行為。

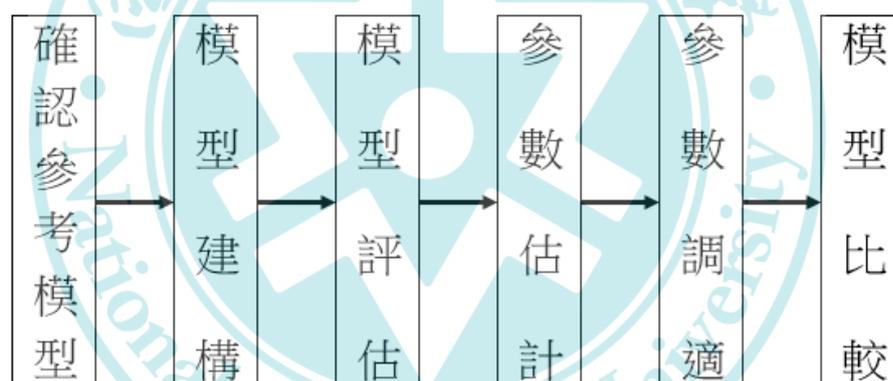


圖 7 研究架構

### 第二節 資料來源

本研究模型建構完成後，為了計算、調適模型之參數，將透過帶入實際失效數據進行模型參數的估算與調適以及模型之間的分析與比較。本研究將利用三組過去學者公開發表於論文當中所使用的資料集（Data Set）數據（表 4），藉以估計各模型之參數與適配度，以進行模型的比較分析。

表 4 實際失效數據來源

資料集	實際失效資料來源	來源文獻
一	即時指揮與控制系統	Musa(1987)
二	即時指揮與控制系統	Ohba(1984)
三	Tandem Computers 軟體產品	Wood(1996)

### 第三節 參數估計法

模型建構完成後，為了計算模型中所有未知的參數 (Parameters) 值，必須將實際失效數據套用至所建構的模型中，並使用參數估計法進行計算。而在 NHPP 軟體可靠度成長模型的參數估計方法中最常使用的兩種估計方法為：最小平方估計法 (Least Squares Estimation, LSE) 與最大概似估計法 (Maximum Likelihood Estimation, MLE)，兩種估計方法於以下做說明。

#### 一、最小平方估計法 (Least Squares Estimation, LSE)

最小平方估計法是使樣本觀察值與估計值的差異之平方和為最小的估計方法 (林惠玲、陳正倉，2008)。其公式如下

$$LSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - \hat{\alpha} - \hat{\beta}x_i)^2 \quad (3.1)$$

其中  $x_i$  為自變數第  $i$  個觀察值

$y_i$  為依變數第  $i$  個觀察值的實際值

$\hat{y}_i$  為第  $i$  個觀察值

$\hat{\alpha}$  為回歸模型截距 (Intercept) 的估計值

$\hat{\beta}$  為回歸模型斜率 (Slope) 之估計值

透過對  $\hat{\alpha}$ 、 $\hat{\beta}$  進行微分，並令微分方程式為零，即可求得迴歸模型之參數值。

## 二、最大概似估計法 (Maximum Likelihood Estimation, MLE)

最大概似估計法一般應用於母體參數 $\theta$ 為未知，若單純從母體抽出一組隨機樣本，則無法得知此組樣本的可能性。因此，若能找到一個估計值 $\theta$ ，且使此組樣本發生之可能性為最大時，則此估計值 $\theta$ 即為最大概似估計值(程大器, 2012)。

假設 $(x_1, x_2 \dots x_n)$ 為抽自母體 $f(x; \theta)$ 之一組隨機樣本，則其概似函數即為此 $n$ 個隨機變數的聯合機率分配 $f(x_1, x_2 \dots x_n; \theta)$ 。但因為參數 $\theta$ 未知，因此將概似函數寫為

$$L(\theta) = f(x_1, x_2 \dots x_n; \theta) = f(x_1; \theta) \cdot f(x_2; \theta) \cdot \dots \cdot f(x_n; \theta) = \prod_{i=1}^n f(x_i; \theta) \quad (3.2)$$

透過對 $L(\theta)$ 進行微分，令微分方程式為零，且 $d^2 \frac{L(\theta)}{d\theta^2} < 0$ ，便可求得估計值 $\hat{\theta}$

由於最小平方估計法便可以簡單的求出各個未知之估計值，並使所求得之估計值和實際值之間誤差的平方和為最小，因此，本研究主要透過最小平方估計法來求解個模型之參數，並進一步畫出模型之適配度曲線。

### 第四節 模型評估標準

在本研究中，我們使用均方差(Mean Square-Error, MSE)、多元判定係數(Coefficient of Multiple Determination,  $R^2$ )、偏差比率(Bias Ratio, Bias)、變量(Variation)、均方根預測誤差(Root Mean Square Prediction Error, RMSPE)等指標評估模型的適合度。該模型在 MSE、Bias、Variation、RMSPE 等衡量標準，相對於在同一數據集上運行的其他模型有更小的數值，及更好的適合度。另一方面， $R^2$ 的值越大，模型相對於其他模型有更高適合度。

#### 一、均方差(Mean Square-Error, MSE)

均方差是透過將預測值與實際數據之間的差距作為量測之標準，兩者之間的差距通常在於隨機變量的預測與參數的估計，公式如下：

$$MSE = \sum_{i=1}^n \frac{(\hat{m}(t_i) - y_i)^2}{n} \quad (3.3)$$

其中  $\hat{m}(t_i)$  = 在  $t_i$  時模型的期望錯誤數

$y_i$  = 在  $t_i$  時偵測到的錯誤數

若 MSE 的值越低，表示適配誤差(Fitting Error)越低，代表模型的適配程度越高。(Kapur et al, 1999)。

## 二、多元判定係數(Coefficient of Multiple Determination, $R^2$ , R square)

為了衡量解釋變數的解釋能力或迴歸方程式的適配度，將多元判定係數判定為回歸差異(Sum of Squares Due to Regression, SSR)占總差異(Sum of Squares Total, SST)的比例。公式如下：

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \quad (3.4)$$

等於

$$R^2 = 1 - \frac{\sum_{i=1}^n [y_i - \hat{m}(t_i)]^2}{\sum_{i=1}^n [y_i - \bar{y}]^2} \quad (3.5)$$

其中  $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$

若  $R^2$  值在 0 到 1 之間，越接近 1，代表模型的解釋能力越佳。

## 三、 偏差比率(Bias Ratio, Bias)

偏差比率即為預測誤差的平均值。公式如下：

$$\text{Bias} = \frac{\sum_{i=1}^n PE_i}{n} \quad (3.6)$$

Bias 的值不論正負，越接近一，代表模型的適配度越好。

## 四、 變量(Variation)

變量為預測誤差的標準差，其實際意義為一同質事件中個體間的差異，來自人為或無法控制，甚至於不明的因素，其公式如下：

$$\text{Variation} = \sqrt{\left(\frac{1}{n-1}\right) \sum (PE_i - \text{Bias})^2} \quad (3.7)$$

若 Variation 越小，代表模型的適配程度越好。

五、 均方根預測誤差(Root Mean Square Prediction Error, RMSPE)

均方根預測誤差是用來測量預測值與實際觀察值之間的接近程度，主要透過偏差比率與變量兩者求得，其公式如下：

$$\text{RMSPE} = \sqrt{(\text{Bias}^2 + \text{Variation}^2)} \quad (3.8)$$

若 RMSPE 的值越小，代表模型的適配程度越高。



## 第四章 模型建構

### 第一節 問題描述

過去數十年間，非齊次卜瓦松過程(NHPP)已被運用於軟體可靠度模型中，而之後的學者雖然也提出了許多不同的模型，但那些模型的特性較為單一，很少會有學者將多種特性結合於同一模型中，再加上近幾年的軟體可靠度研究也逐漸轉移方向與觀點，使得以特性作為研究核心的軟體可靠度模型愈來愈少。

本研究期望能夠建構有效預測軟體失效行為之 SRGM，以其做為軟體可靠度之評估與參考。因此本研究以 NHPP 為基礎，將三種不同特性之模型結合於故障內容函數當中，並依照特性假設錯誤偵測率函數為一常數，最後與現存之模型進行分析與比較，以評估所提出之模型運用於實際失效數據時，能否獲得更佳的預測結果。

### 第二節 軟體可靠度模型

#### 一、符號說明

$m(t)$	均值函數：期望在時間 $t$ 時偵測到的錯誤數
$a$	軟體初始的錯誤數
$a(t)$	故障內容函數：在時間 $t$ 時軟體的總錯誤數
$b$	錯誤偵測率
$b(t)$	錯誤偵測率函數
$\alpha$	錯誤生成率
$\beta$	曲折因子
$r$	模型組合常數 1
$K$	模型組合常數 2
$\lambda(t)$	失效強度函數；每單位時間內的錯誤觀測率

## 二、模型建構

模型的基本假設如下：

1. 錯誤的偵測與移除過程遵循 NHPP。
2. 軟體中剩餘的錯誤導致軟體在執行期間出現新的錯誤。
3. 錯誤檢測率在測試過程中是恆定的。
4. 軟體的錯誤率與剩餘錯誤成正比。
5. 當偵測到的錯誤被刪除時，有可能產生新的錯誤。
6. 在測試時可能會產生錯誤，並且錯誤移除時可能會產生新的錯誤，亦即不完美除錯。
7. 在測試開始時，測試團隊對於軟體並不熟悉，也沒有如何正確除錯的相關知識。隨著時間過去，團隊也獲得了越來越多關於除錯的知識。

根據以上的假設，可得故障內容函數與錯誤偵測率函數如下：

$$a(t) = (1 - k)[a(1 - r) + rae^{\alpha t}] + ka(\alpha - e^{-\beta t}) \quad (4.1)$$

和

$$b(t) = b \quad (4.2)$$

其中  $a(t)$  為包含三種模型特性之故障內容函數。 $b(t)$  代表錯誤偵測率，為一常數。

將  $a(t)$  與  $b(t)$  代入方程式(2.3)中，並令邊際條件  $m(0)=0$ ，再透過對方程式進行積分，即可得此模型的均值函數如下：

$$m(t) = \frac{1}{e^{bt}} \left( \begin{aligned} & a(e^{bt} - 1) - ak(e^{bt} - 1) - ar(e^{bt} - 1) \\ & + abk \left( \frac{1}{e^{\beta(b-\beta)}} - \frac{e^{bt}}{e^{\beta}e^{bt}(b-\beta)} \right) + ak\alpha(e^{bt} - 1) \\ & + akr(e^{bt} - 1) + \frac{abre^{\alpha}(e^{bt+\alpha} - 1)}{b + \alpha} - \frac{abkre^{\alpha}(e^{bt+\alpha} - 1)}{b + \alpha} \end{aligned} \right) \quad (4.3)$$

其中，可發現當  $k=0$  時，均值函數即為 M1 模型；當  $k=1$  時，則為 M2 模型。

失效強度函數則為：

$$\begin{aligned}
\lambda(t) &= m'(t) \\
&= -ab \left( -re^{\alpha+\alpha t} + \frac{k}{e^{\beta+\beta t}} + kre^{\alpha+\alpha t} - 1 \right) \\
&\quad - \frac{b}{e^{bt}} \left( \begin{aligned} &a(e^{bt} - 1) - ak(e^{bt} - 1) - ar(e^{bt} - 1) \\ &+ abk \left( \frac{1}{e^{\beta}(b - \beta)} - \frac{e^{bt}}{e^{\beta}e^{\beta t}(b - \beta)} \right) + aka(e^{bt} - 1) \\ &+ akr(e^{bt} - 1) + \frac{abre^{\alpha}(e^{bt+\alpha t} - 1)}{b + \alpha} - \frac{abkre^{\alpha}(e^{bt+\alpha t} - 1)}{b + \alpha} \end{aligned} \right)
\end{aligned}
\tag{4.4}$$

### 第三節 資料分析與模型比較

為評估所提出的模型之有效性，本研究利用三組實際失效資料數據來評估各模型之參數與適配度，並進行模型評估標準分析。參數估計部分使用最小平方估計法來進行模型參數估計，其他模型算法與此相似，選擇最小平方估計法作為計算法是因為其可簡便求得未知估計值，並使所求得估計值與實際值之間誤差的平方和降至最小，另外其所產生的結果較為客觀。此外，因評估標準種類較多，因此本研究僅提出五種標準進行比較，以提供客觀且有效的評估方法。

#### 一、參數估計

本研究參數估計部分是使用最小平方估計法來進行模型參數估計，其他模型算法與此相似，在研究中僅列出新模型之估計方式。求得各模型之參數值後，將其代入至各模型的均值函數中，即可描繪出各模型之適配度曲線。

假設  $a \cdot b \cdot \alpha \cdot \beta \cdot r \cdot k$  是透過  $n$  組觀察數據： $(t_0, m_0), (t_1, m_1), (t_2, m_2), \dots, (t_n, m_n)$  所決定的，其中  $m_i$  為在時間  $(0, t_i)$  內偵測到之總錯誤數目。透過最小平方估計法，即可得模型估計函數 (Evaluation Function) 如下：

$$\text{Min } M(a \cdot b \cdot \alpha \cdot \beta \cdot r \cdot k) = \sum_{i=1}^n (m_i - m(t_i))^2 = \sum_{i=1}^n \left( m_i - \frac{1}{e^{bt}} \left( a(e^{bt} - 1) - ak(e^{bt} - 1) - ar(e^{bt} - 1) + abk \left( \frac{1}{e^{\beta(b-\beta)}} - \frac{e^{bt}}{e^{\beta} e^{\beta t}(b-\beta)} \right) \right) + ak\alpha(e^{bt} - 1) + akr(e^{bt} - 1) + \frac{abre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} - \frac{abkre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} \right) \right)^2 \quad (4.5)$$

透過分別對方程式中的  $a \cdot b \cdot \alpha \cdot \beta \cdot r \cdot k$  進行微分，並令其偏導數 (Partial Derivatives) 等於零，即可得下列方程式：

$$\frac{\partial M(a \cdot b \cdot \alpha \cdot \beta \cdot r \cdot k)}{\partial a} = \sum_{i=1}^n -\frac{2}{e^{bt}} \left( m_i - \frac{1}{e^{bt}} a(e^{bt} - 1) - ak(e^{bt} - 1) - ar(e^{bt} - 1) + abk \left( \frac{1}{e^{\beta(b-\beta)}} - \frac{e^{bt}}{e^{\beta} e^{\beta t}(b-\beta)} \right) + ak\alpha(e^{bt} - 1) + akr(e^{bt} - 1) + \frac{abre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} - \frac{abkre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} \right) \left( e^{bt} - k(e^{bt} - 1) - r(e^{bt} - 1) + k\alpha(e^{bt} - 1) + kr(e^{bt} - 1) + bk \left( \frac{1}{e^{\beta(b-\beta)}} - \frac{e^{bt}}{e^{\beta} e^{\beta t}(b-\beta)} \right) + \frac{bre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} - \frac{bkre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} - 1 \right) \right) = 0 \quad (4.6)$$

$$\frac{\partial M(a \cdot b \cdot \alpha \cdot \beta \cdot r \cdot k)}{\partial b} = \sum_{i=1}^n -2 \left( \frac{1}{e^{bt}} \left( ak \frac{1}{e^{\beta(b-\beta)}} \left( 1 - \frac{e^{bt}}{e^{\beta t}} \right) + ate^{bt} - akte^{bt} - abk \left( \frac{1}{e^{\beta(b-\beta)^2} - \frac{e^{bt}}{e^{\beta} e^{\beta t}(b-\beta)^2} + \frac{te^{bt}}{e^{\beta} e^{\beta t}(b-\beta)} \right) - arte^{bt} + \frac{are^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} + akate^{bt} + akrete^{bt} - \frac{1}{b+\alpha} \left( \frac{abre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} - akre^{\alpha}(e^{bt+\alpha t} - 1) + abrte^{\alpha} e^{bt+\alpha t} + \frac{abkre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} - abkrte^{\alpha} e^{bt+\alpha t} \right) \right) \right) \left( m_i - \frac{1}{e^{bt}} a(e^{bt} - 1) - ak(e^{bt} - 1) - ar(e^{bt} - 1) + abk \left( \frac{1}{e^{\beta(b-\beta)}} - \frac{e^{bt}}{e^{\beta} e^{\beta t}(b-\beta)} \right) + ak\alpha(e^{bt} - 1) + akr(e^{bt} - 1) + \frac{abre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} - \frac{abkre^{\alpha}(e^{bt+\alpha t} - 1)}{b+\alpha} \right) \right) = 0 \quad (4.7)$$

$$\frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial \alpha} =$$

$$\sum_{i=1}^n \frac{-2}{e^{bt}} \left( \begin{array}{c} m_i - \frac{1}{e^{bt}} a(e^{bt} - 1) - ak(e^{bt} - 1) \\ -ar(e^{bt} - 1) + abk \left( \frac{1}{e^{\beta}(b-\beta)} - \frac{e^{bt}}{e^{\beta}e^{\beta t}(b-\beta)} \right) \\ + aka(e^{bt} - 1) + akr(e^{bt} - 1) \\ + \frac{abre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} - \frac{abkre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} \end{array} \right) \left( \begin{array}{c} ak(e^{bt} - 1) + \frac{abre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} \\ - \frac{abre^{\alpha}(e^{bt+at} - 1)}{(b+\alpha)^2} + \frac{abrte^{\alpha}e^{bt+at}}{b+\alpha} \\ - \frac{abkre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} + \frac{abkre^{\alpha}(e^{bt+at} - 1)}{(b+\alpha)^2} \\ - \frac{abkrte^{\alpha}e^{bt+at}}{b+\alpha} \end{array} \right)$$

$$= 0 \quad (4.8)$$

$$\frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial \beta} =$$

$$\sum_{i=1}^n \frac{-2}{e^{bt}} abk \left( \begin{array}{c} m_i - \frac{1}{e^{bt}} a(e^{bt} - 1) - ak(e^{bt} - 1) \\ -ar(e^{bt} - 1) + abk \left( \frac{1}{e^{\beta}(b-\beta)} - \frac{e^{bt}}{e^{\beta}e^{\beta t}(b-\beta)} \right) \\ + aka(e^{bt} - 1) + akr(e^{bt} - 1) + \frac{abre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} \\ - \frac{abkre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} \end{array} \right) \left( \begin{array}{c} \frac{1}{e^{\beta}(b-\beta)^2} - \frac{1}{e^{\beta}(b-\beta)} \\ + \frac{e^{bt}}{e^{\beta}e^{\beta t}(b-\beta)} - \frac{e^{bt}}{e^{\beta}e^{\beta t}(b-\beta)^2} \\ + \frac{te^{bt}}{e^{\beta}e^{\beta t}(b-\beta)} \end{array} \right)$$

$$= 0 \quad (4.9)$$

$$\frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial r} =$$

$$\sum_{i=1}^n \frac{2}{e^{bt}} \left( \begin{array}{c} m_i - \frac{1}{e^{bt}} a(e^{bt} - 1) - ak(e^{bt} - 1) \\ -ar(e^{bt} - 1) + abk \left( \frac{1}{e^{\beta}(b-\beta)} - \frac{e^{bt}}{e^{\beta}e^{\beta t}(b-\beta)} \right) \\ + aka(e^{bt} - 1) + akr(e^{bt} - 1) \\ + \frac{abre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} - \frac{abkre^{\alpha}(e^{bt+at} - 1)}{b+\alpha} \end{array} \right) \left( \begin{array}{c} a(e^{bt} - 1) - ak(e^{bt} - 1) \\ - \frac{abe^{\alpha}(e^{bt+at} - 1)}{b+\alpha} + \frac{abke^{\alpha}(e^{bt+at} - 1)}{b+\alpha} \end{array} \right)$$

$$= 0 \quad (4.10)$$

$$\begin{aligned}
& \frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial k} = \\
& \sum_{i=1}^n \frac{-2}{e^{bt}} \left( \begin{array}{l} m_i - \frac{1}{e^{bt}} a(e^{bt} - 1) - ak(e^{bt} - 1) \\ -ar(e^{bt} - 1) + abk \left( \frac{1}{e^\beta(b-\beta)} - \frac{e^{bt}}{e^\beta e^{\beta t}(b-\beta)} \right) \\ + ak\alpha(e^{bt} - 1) + akr(e^{bt} - 1) \\ + \frac{abre^\alpha(e^{bt+\alpha t} - 1)}{b+\alpha} - \frac{abkre^\alpha(e^{bt+\alpha t} - 1)}{b+\alpha} \end{array} \right) \left( \begin{array}{l} a\alpha(e^{bt} - 1) - a(e^{bt} - 1) \\ +ar(e^{bt} - 1) + ab \left( \frac{1}{e^\beta(b-\beta)} - \frac{e^{bt}}{e^\beta e^{\beta t}(b-\beta)} \right) \\ - \frac{abre^\alpha(e^{bt+\alpha t} - 1)}{b+\alpha} \end{array} \right) \\
& = 0 \tag{4.11}
\end{aligned}$$

最後，將實際失效資料數據代入並求解下列聯立方程式即可得參數值

$a, b, \alpha, \beta, r, k$ 。

$$\left\{ \begin{array}{l} \frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial a} = 0 \\ \frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial b} = 0 \\ \frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial \alpha} = 0 \\ \frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial \beta} = 0 \\ \frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial r} = 0 \\ \frac{\partial M(a, b, \alpha, \beta, r, k)}{\partial k} = 0 \end{array} \right. \tag{4.12}$$

## 二、軟體失效數據

### 1. 資料集 一

資料來源為 Musa (1987) 根據即時指揮與控制系統的系統測試數據。  
系統測試時間為 25 小時，系統偵測到的錯誤數目共有 136 個。詳細失效數據內容如表 5 所示。

### 2. 資料集 二

資料來源為 Ohba (1984) 根據即時指揮與控制系統的系統測試數據。  
系統測試時間為 21 天，系統偵測到的錯誤數目共有 46 個。詳細失效數據內容如表 5 所示。

### 3. 資料集 三

資料來源為 Wood (1996) 根據 Tandem Computers 軟體產品的系統測試數據。  
系統測試時間為 20 周，系統偵測到的錯誤數目共有 100 個。詳細失效數據內容如表 5 所示。



表 5 資料集

測試時間 (小時/天/周)	資料集一	資料集二	資料集三
	累積錯誤數	累積錯誤數	累積錯誤數
1	27	2	16
2	43	3	24
3	54	4	27
4	64	5	33
5	75	7	41
6	83	9	49
7	84	11	54
8	89	12	58
9	92	19	69
10	93	21	75
11	97	22	81
12	104	24	86
13	106	26	90
14	111	30	93
15	116	31	96
16	122	37	98
17	122	38	99
18	127	41	100
19	128	42	100
20	129	45	100
21	131	46	
22	132		
23	134		
24	135		
25	136		

資料來源：Musa (1987)、Ohba (1984)、Wood (1996)

### 三、模型測試與比較

模型建構完成後，本研究為瞭解所提出的模型是否能夠在實際情況下有效的預測軟體失效行為，利用三組實際失效數據的資料集進行測試。而在模型比較部分則是選擇與本研究所提出的模型相似之現有模型以及較為知名的軟體可靠度模型進行比較，因此在選擇現有模型上挑選了 G-O、DSS、YEI、YLI、PNZ、Roy 等模型與新模型進行比較。以下針對比較結果進行各資料集之說明。

#### 1. 資料集 一

表 6 為透過資料集一所估計出之各模型的參數值，分別代入到各模型的均值函數中，即可計算出各時段的期望累積錯誤數目，並畫出各模型之適配度曲線(圖 6-圖 12)。圖 13 則為各模型與資料集之間的適配度曲線之比較，由此可以看出除了 PNZ 與 DSS 這兩個模型以外的適配度皆非常高。而除了利用適配度曲線進行比較之外，本研究還使用了數個評估標準進行比較，用以評估所提出的模型是否較其他模型更能夠準確預測錯誤，其比較結果記於表 7 中。

表 6 模型參數估計 (資料集一)

	a	b	$\alpha$	$\beta$	r	k
Propose model	184.2839	0.917	0.0013	0.1763	11.3595	0.4562
G-O model	142.2796	0.1248				
DSS model	136.9149	0.284				
YEI model	95.3378	0.2628	0.0167			
YLI model	83.4377	0.3315	0.0287			
P-N-Z model	7.8692	16.0551	0.6774	8.1289		
Roy model	14.0946	0.1462	9.9870	0.0625		

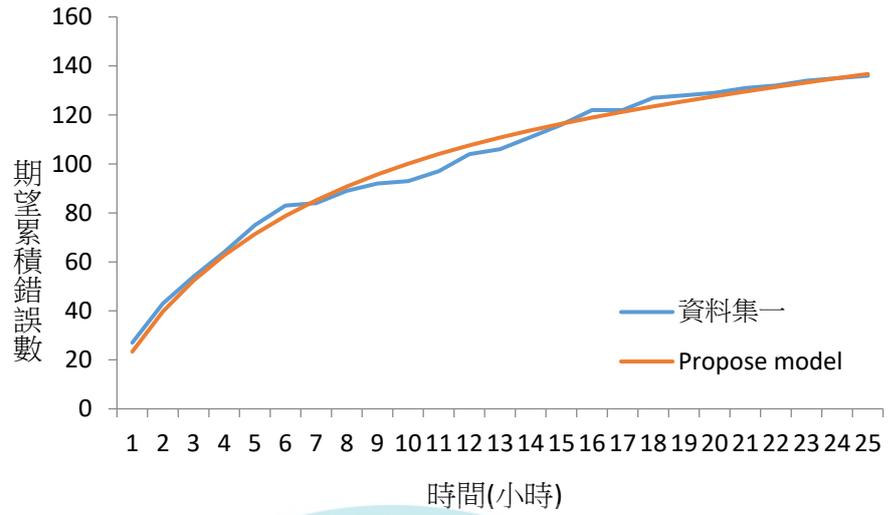


圖 6 Propose 模型之適配度曲線 (資料集一)

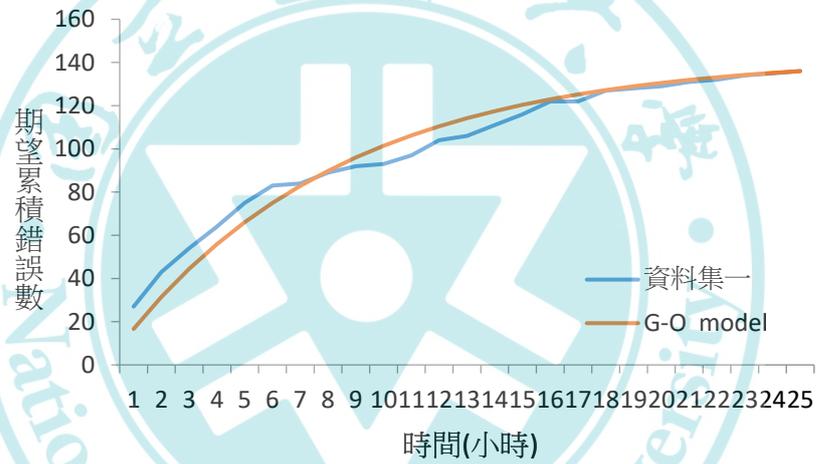


圖 7 G-O 模型之適配度曲線 (資料集一)

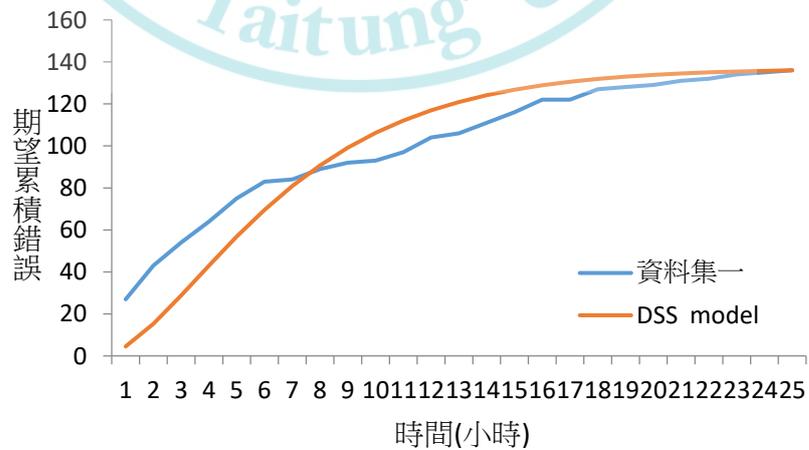


圖 8 DSS 模型之適配度曲線 (資料集一)

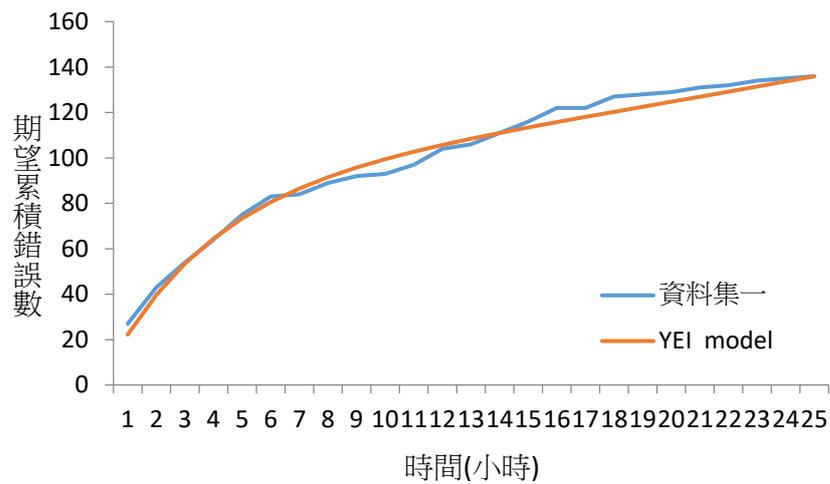


圖 9 YEI 模型之適配度曲線 (資料集一)

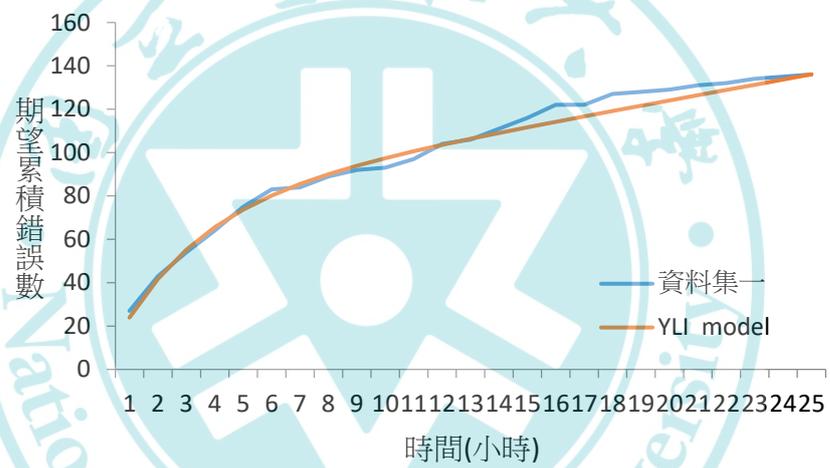


圖 10 YLI 模型之適配度曲線 (資料集一)

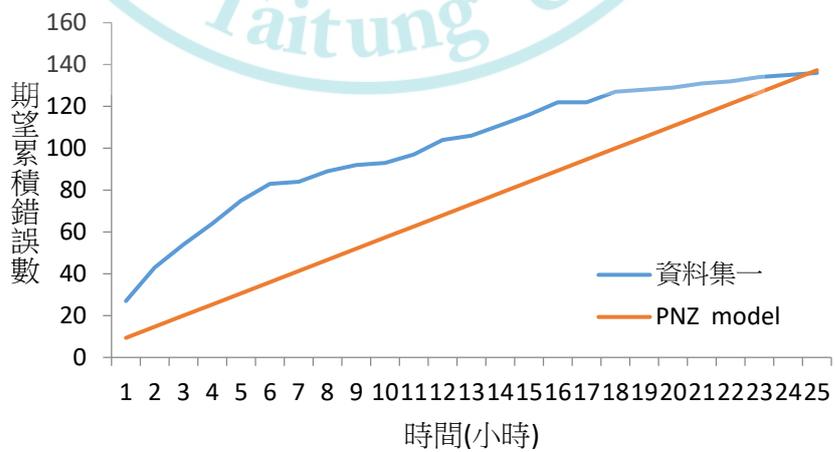


圖 11 PNZ 模型之適配度曲線 (資料集一)

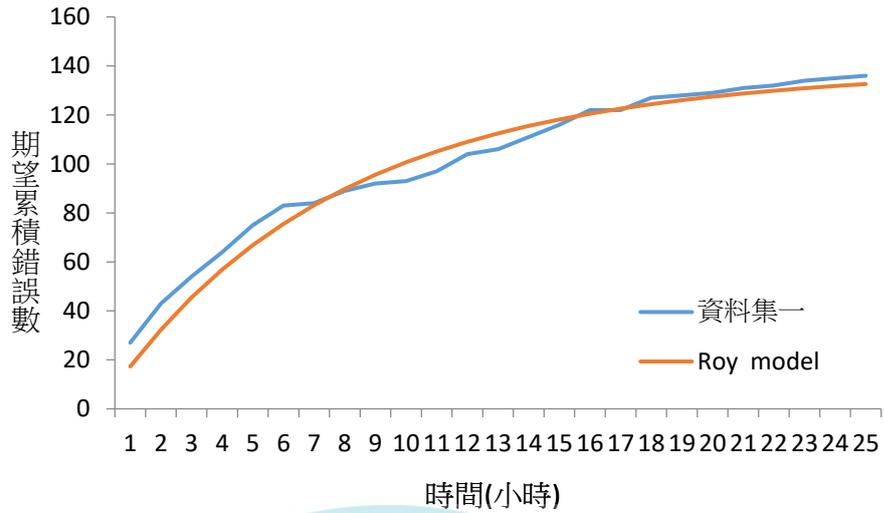


圖 12 Roy 模型之適配度曲線 (資料集一)

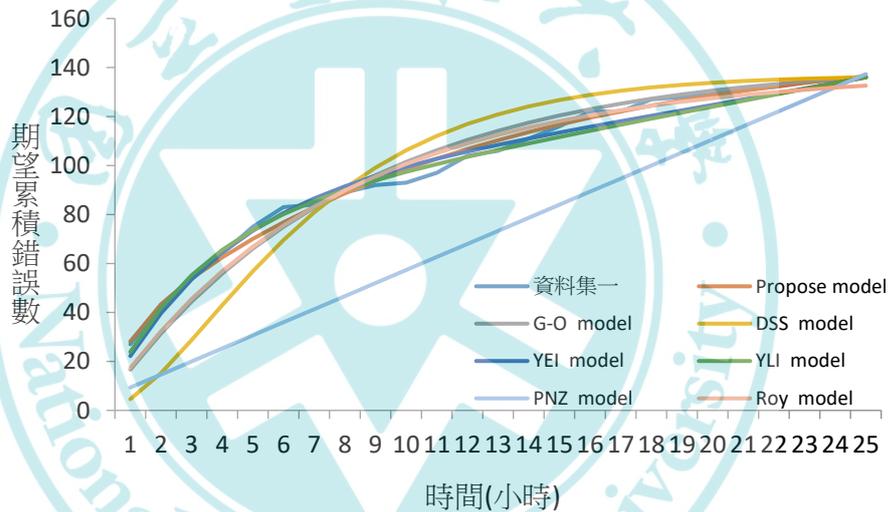


圖 13 本研究推出模型與現有模型之適配度比較圖 (資料集一)

表 7 模型比較結果

	MSE	R <sup>2</sup>	Bias	Variation	RMSPE
Propose model	*10.1938	*0.9888	0.0532	*3.2582	*3.2586
G-O model	36.1133	0.9603	*-0.0058	6.1334	6.1334
DSS model	168.7172	0.8184	-0.1475	13.2561	13.2569
YEI model	13.6394	0.9850	-1.0870	3.6024	3.7628
YLI model	13.8487	0.9848	-1.7695	3.3413	3.7809
P-N-Z model	768.9232	0.1558	-24.5251	13.2069	27.8550
Roy model	29.4868	0.9674	-1.4129	5.3705	5.5532

註：\*為評估標準比較中最佳者。

透過表 7 可以看到在 MSE、Variation 與 RMSPE 的部分，本研究提出的模型的值皆小於其他模型。在  $R^2$  的部分則是大於其他模型，雖然在 Bias 的部分並未獲得最好的結果，但其餘四項標準皆為最佳的，因此，整體來看本研究提出的模型在預測失效行為上的效果較其他模型佳。

## 2. 資料集 二

表 8 為透過資料集二所估計出之各模型的參數值，分別代入各模型的均值函數當中，計算出各時段的期望累積錯誤數目，並畫出各模型之適配度曲線（圖 14-圖 20）。圖 21 則為各模型與資料集之間的適配度曲線之比較，由此可以看出除了 G-O 與 YEI 這兩個模型以外的適配度皆非常高。而除了使用適配度曲線進行比較之外，本研究還使用了模型評估標準進行比較，以此評估所提出的模型是否較其他模型有更佳的預測能力，其比較結果記於表 12。

表 8 模型參數估計（資料集二）

	a	b	$\alpha$	$\beta$	r	k
Propose model	35.15533	0.13524	0.01923	0.13516	5.0505	0.469115
G-O model	54.2479	0.0657				
DSS model	67.1874	0.1121				
YEI model	5.9387	29.8974	0.0976			
YLI model	5.8037	0.1943	0.4356			
P-N-Z model	2.6435	0.3110	0.9257	0.0209		
Roy model	14.8590	0.0136	12.6678	0.2097		

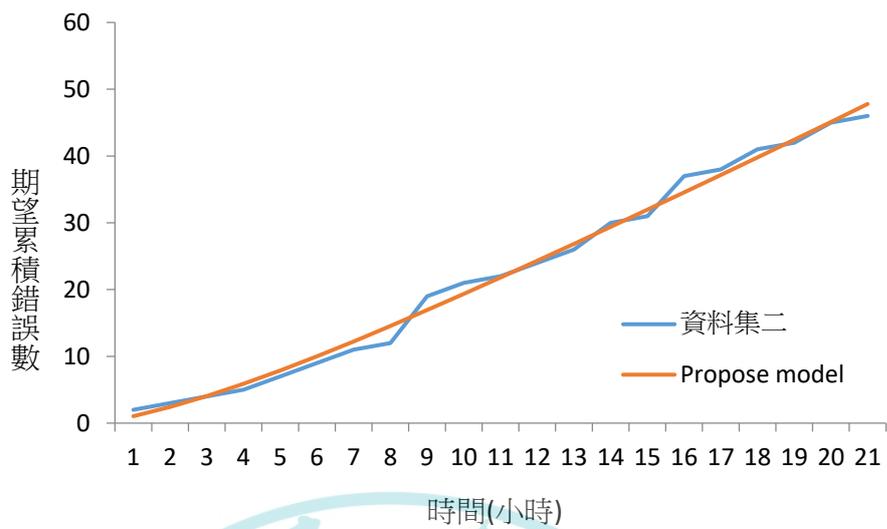


圖 14 Propose 模型之適配度曲線 (資料集二)

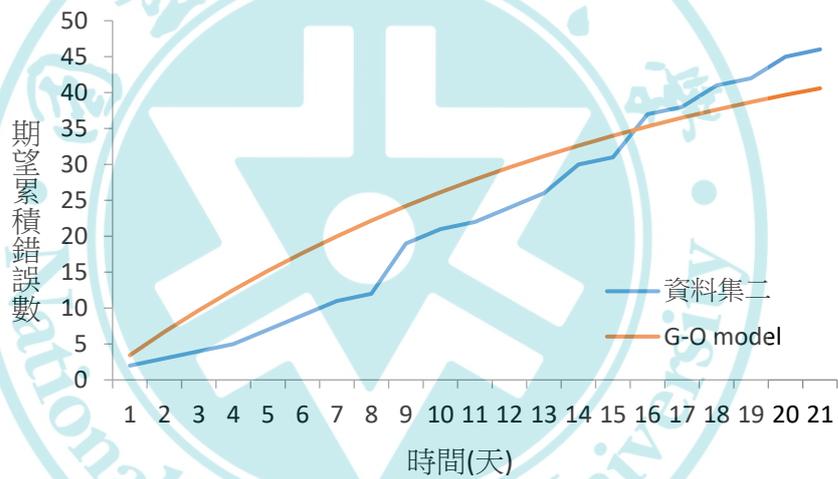


圖 15 G-O 模型之適配度曲線 (資料集二)

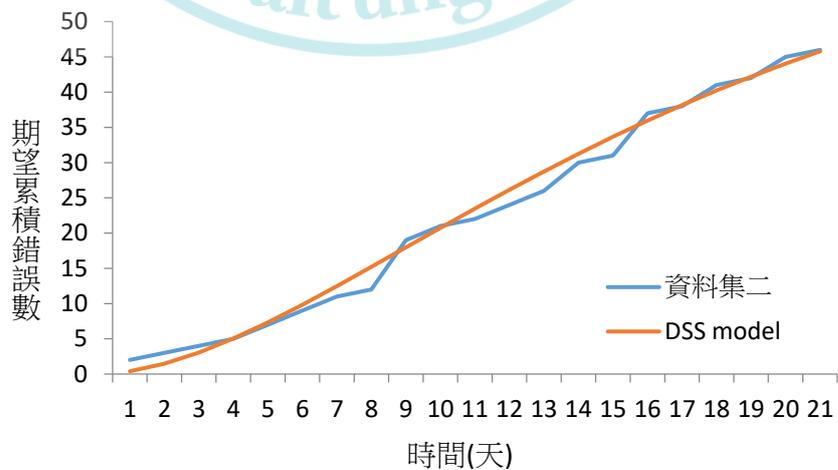


圖 16 DSS 模型之適配度曲線 (資料集二)

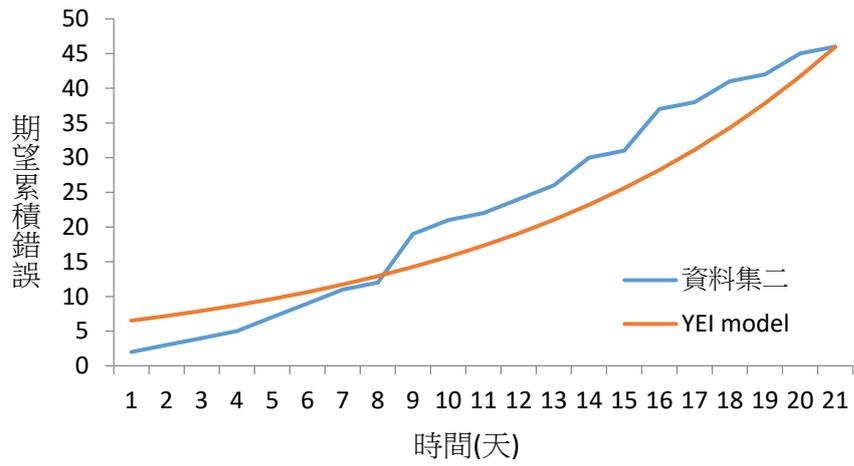


圖 17 YEI 模型之適配度曲線 (資料集二)

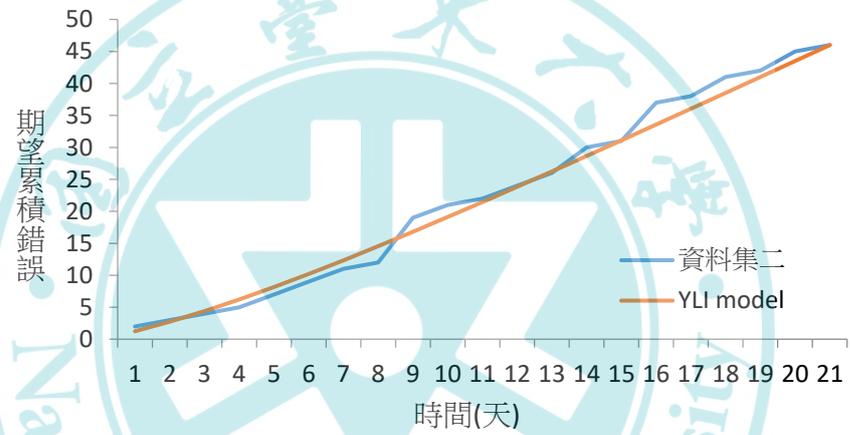


圖 18 YLI 模型之適配度曲線 (資料集二)

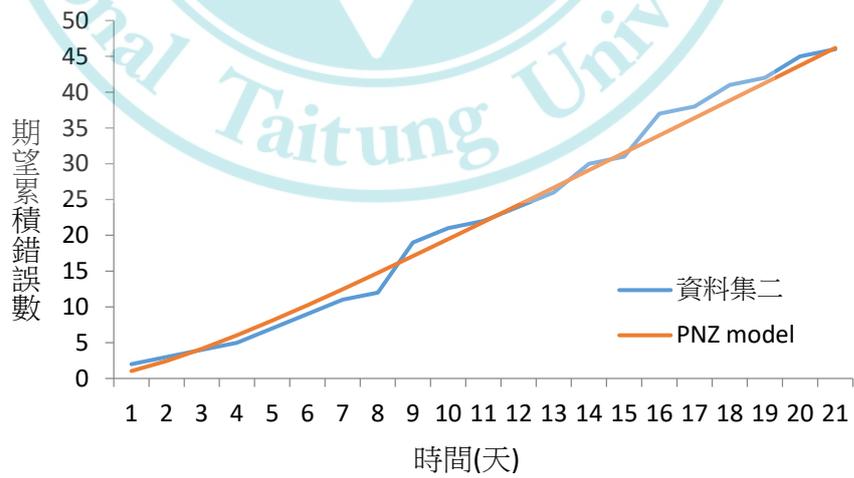


圖 19 PNZ 模型之適配度曲線 (資料集二)

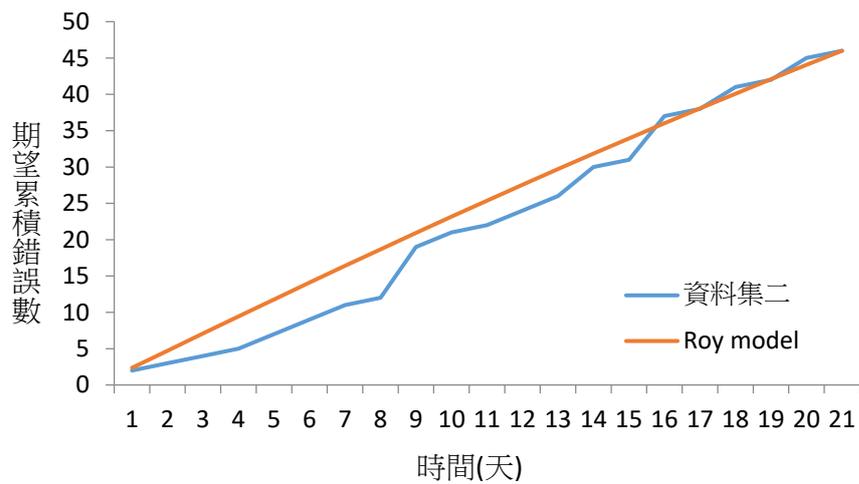


圖 20 Roy 模型之適配度曲線 (資料集二)

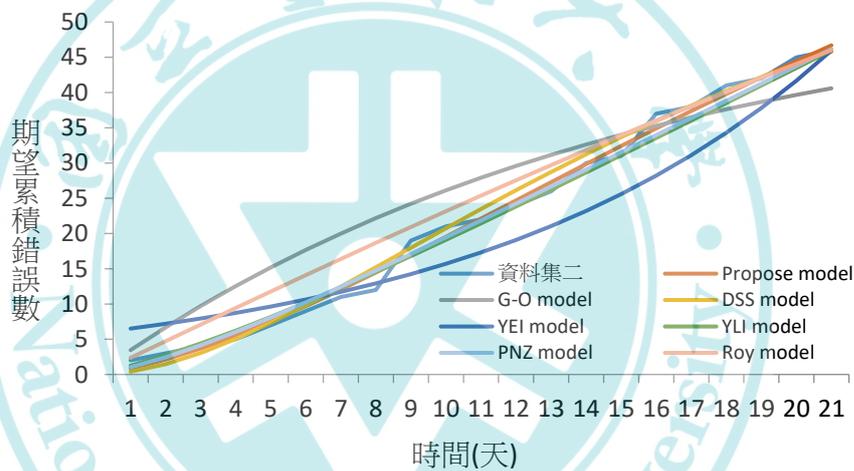


圖 21 本研究推出模型與現有模型之適配度比較圖 (資料集二)

表 9 模型比較結果

	MSE	R <sup>2</sup>	Bias	Variation	RMSPE
Propose model	*1.5522	*0.9928	*0.0268	*1.2763	*1.2766
G-O model	32.3441	0.8492	3.1609	4.8446	5.7846
DSS model	2.1843	0.9898	0.3868	1.4617	1.5120
YEI model	22.5297	0.8949	-2.1130	4.3553	4.8408
YLI model	2.3338	0.9891	-0.4443	1.4977	1.5622
P-N-Z model	1.9551	0.9909	-0.2614	1.4075	1.4316
Roy model	10.2599	0.9522	2.3025	2.2818	3.2416

註：\*為評估標準比較中最佳者。

透過此表可以瞭解在 MSE、Variation 與 RMSPE 的部分，本研究的模型的值皆小於其他模型。在  $R^2$  的部分則是大於其他模型，雖然在 Bias 的部分並未獲得較好的結果，但其餘四項標準皆為最佳的。整體看來本研究提出的模型在預測錯誤的效果較其他模型更佳。

### 3. 資料集 三

表 10 為透過資料集三所估計出之各模型的參數值，分別代入到各模型的均值函數中，即可計算出各時段的期望累積錯誤數目，並畫出各模型之適配度曲線（圖 22-圖 28）。圖 29 則為各模型與資料集之間的適配度曲線之比較，由此可以看出除了 YEI、YLI 與 PNZ 這三個模型之外，其餘的模型適配度與曲線皆非常相似。而除了利用適配度曲線進行比較之外，本研究還進行了模型評估標準之比較，用以評估所提出的模型是否較其他模型優秀，其比較結果記於表 14 之中。

表 10 模型參數估計（資料集三）

	a	b	$\alpha$	$\beta$	r	k
Propose model	484.0048	0.9358	0.0221	0.0397	-0.2455	0.4965
G-O model	110.1675	0.1194				
DSS model	96.5055	0.2941				
YEI model	29.2564	0.4115	0.0694			
YLI model	6.3476	13.4036	0.7544			
P-N-Z model	7.0691	14.3160	0.7030	3.9385		
Roy model	11.3416	0.1054	10.2026	0.2856		

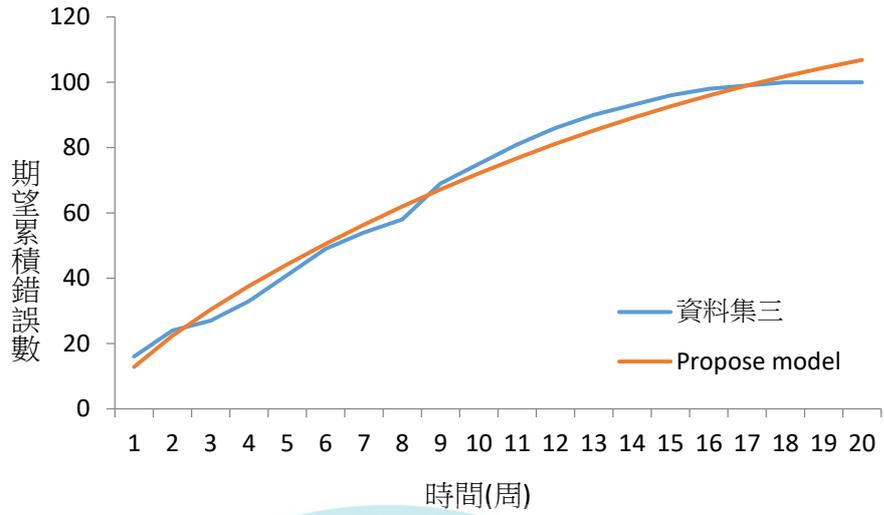


圖 22 Propose 模型之適配度曲線 (資料集三)

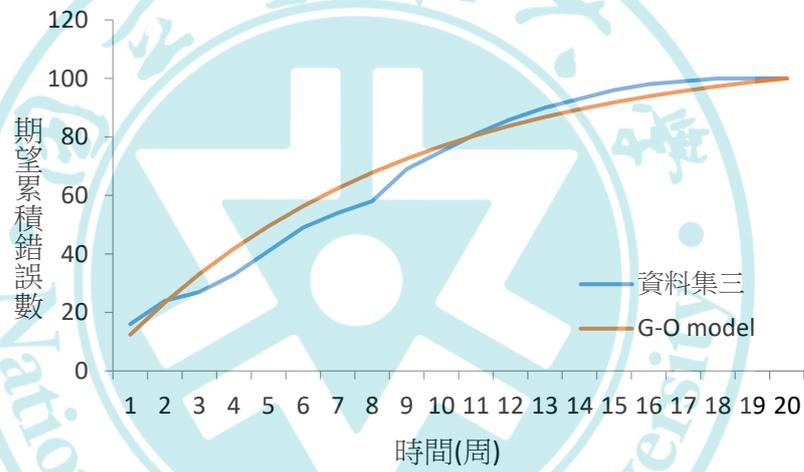


圖 23 G-O 模型之適配度曲線 (資料集三)

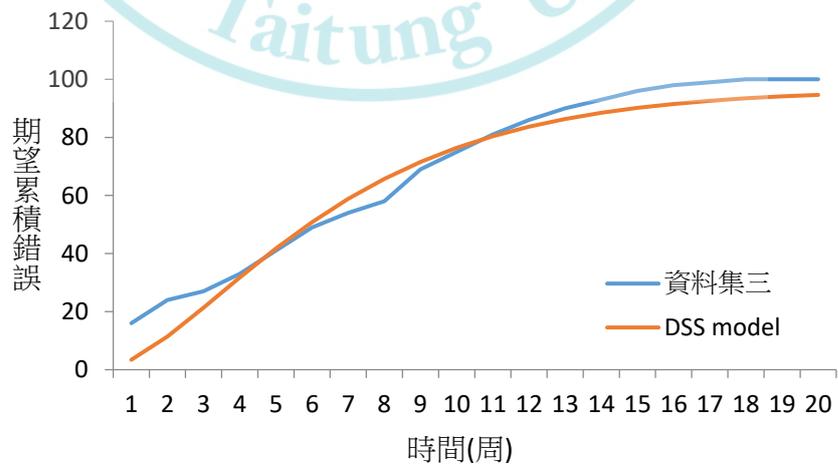


圖 24 DSS 模型之適配度曲線 (資料集三)

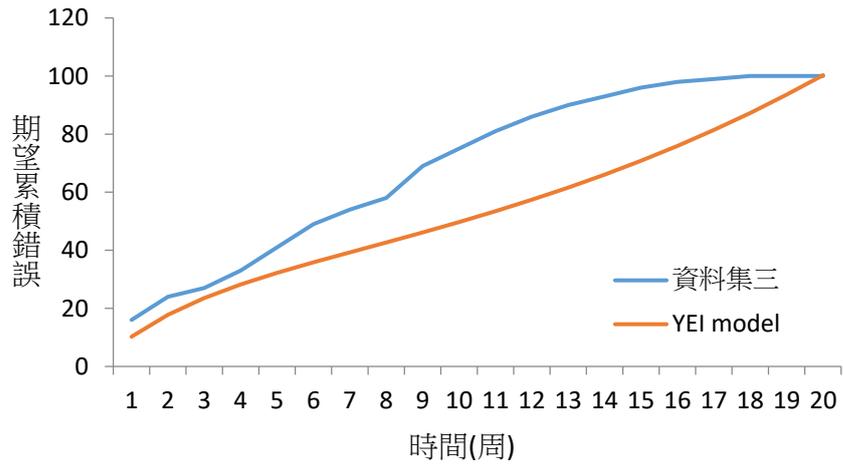


圖 25 YEI 模型之適配度曲線 (資料集三)

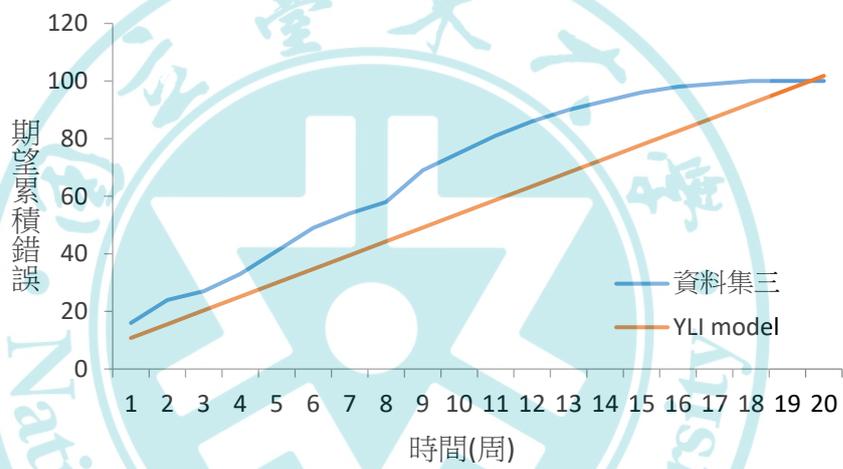


圖 26 YLI 模型之適配度曲線 (資料集三)

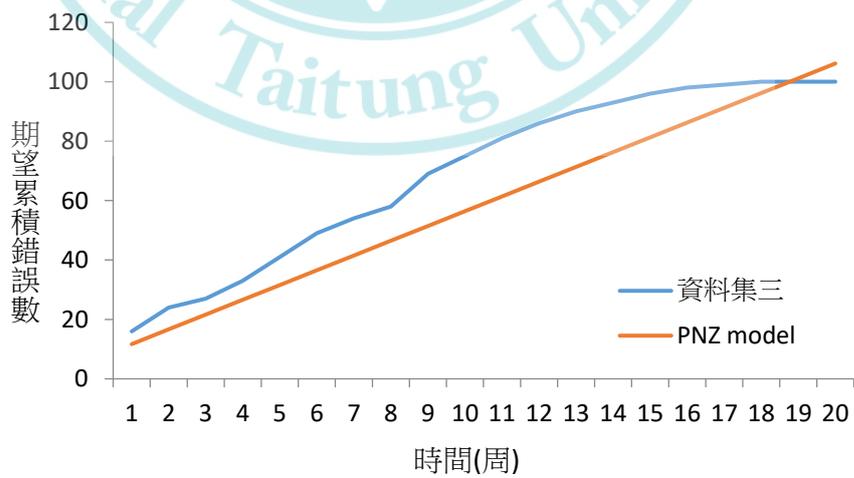


圖 27 PNZ 模型之適配度曲線 (資料集三)

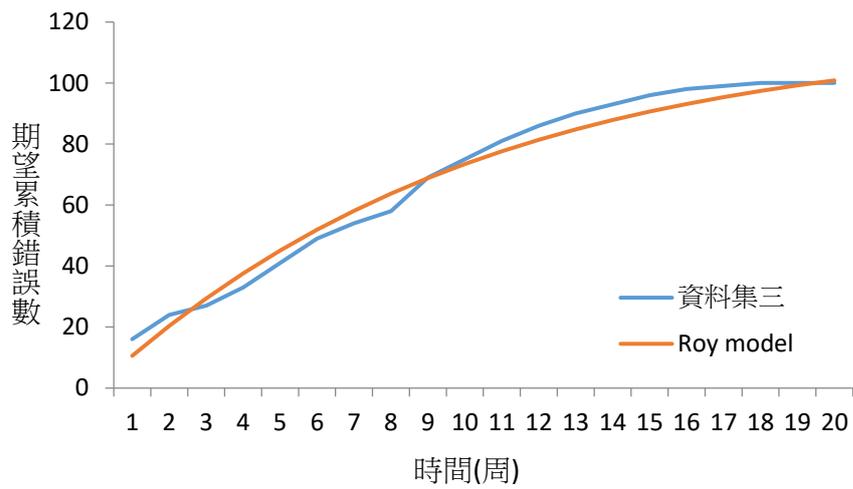


圖 28 Roy 模型之適配度曲線 (資料集三)

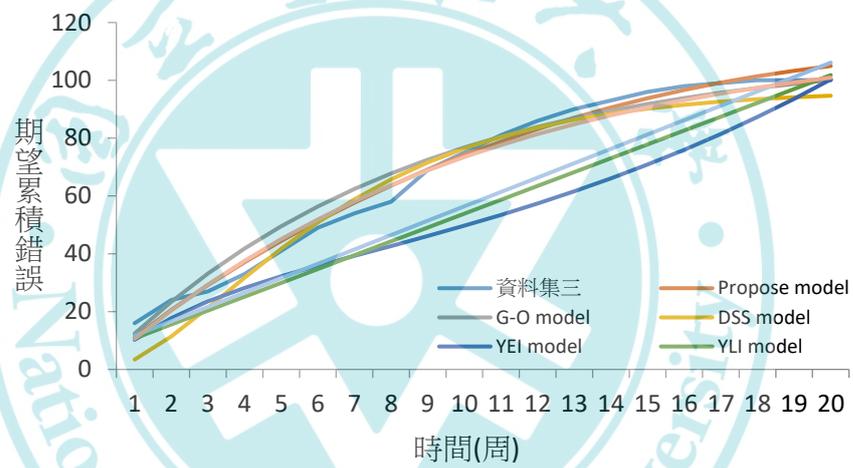


圖 29 本研究推出模型與現有模型之適配度比較圖 (資料集三)

表 11 模型比較結果

	MSE	R <sup>2</sup>	Bias	Variation	RMSPE
Propose model	*12.8059	*0.9842	*-0.0126	*3.67147	*3.67149
G-O model	25.9973	0.9680	1.2709	5.0661	5.2231
DSS model	35.4303	0.9564	-3.0297	5.2567	6.0673
YEI model	335.5480	0.5872	-15.7726	9.5572	18.4422
YLI model	221.2776	0.7278	-13.1791	7.0778	14.9594
P-N-Z model	159.8183	0.8034	-10.5475	7.1502	12.7426
Roy model	15.3078	0.9812	-1.0851	3.8567	4.0064

註：\*為評估標準比較中最佳者。

透過此表可以瞭解在 MSE、Variation、 $R^2$ 與 RMSPE 的部分，本研究的模型的值均優於其他模型。雖然在 Bias 的部分並未獲得最好的結果，但其餘四項標準皆為最佳的，整體看來本研究提出的模型在預測錯誤上的效果較其他模型更佳。

#### 第四節 小結

經過前一個小節的分析，我們可以得出以下的小結：

1. 本研究提出的模型實際運用於失效數據時，能夠得到極好的適配度曲線，這代表本研究的模型能夠有效地預測軟體失效。
2. 在五項模型評估標準當中，僅在第一個數據集當中有四項為最佳，其餘的數據集是五項皆為最佳。整體看來，本研究提出的模型在預測軟體失效的預測能力是比現有模型來得更準確。
3. 根據以上兩點，我們可以推斷出，若能夠在故障內容函數當中包含多種特性，其模型預測軟體失效的能力將會比單一特性的模型來得更加準確、有效。

## 第五章 結論與建議

### 第一節 結論

時至今日，因為軟體越來越普遍，突顯了軟體品質的重要性，也因此軟體可靠度的相關研究也依然在進行中，而過去的數十年以來，許多的專家學者提出各種各樣的軟體可靠度成長模型，這些模型的目的都是為了預測各種不同的軟體失效行為。

本研究提出了一個新的軟體可靠度成長模型，其故障內容函數包含了三種模型的特性，並以非齊次卜瓦松分配為基礎，以及考量不完美除錯，並運用數學計算得出了該模型的均值函數，而限制在於假設錯誤偵測率為一常數。當模型建構完畢之後，運用三組實際失效資料集進行測試，利用最小平方方法估計各模型之參數值，接著畫出適配度曲線圖，並以 MSE、 $R^2$ 、Bias、Variation、RMSPE 等五項比較標準，將各模型與本研究提出的模型進行評估與比較。

而根據表 7、表 9、表 11 的數值可以得知，本研究提出的模型在第一個數據集當中五項標準中有四項為最佳，而之後的數據集當中五項標準皆為最佳。由此得知，本研究所提出的模型比其他現有模型具有更好的適配度與預測能力。這也表示在軟體可靠度成長模型中，故障內容函數若能包含多種特性時，預測軟體失效能力會比單一類型的故障內容函數之模型來得準確許多。

### 第二節 建議

軟體的測試與除錯一直都是一項非常重要的課題，若實際等到錯誤發生可能是問題已經非常嚴重的時候了，整個過程會變的非常耗費時間與人力，但若是將含有重大錯誤的軟體發行到市面上，將會嚴重影響公司的評價或造

成實際的損失，因此若是開發人員能夠在一開始便將較為重大的錯誤解決，使得公司能夠在不虧損之前將軟體發布，對開發人員來說是一門重要的課題。

未來研究建議可以建構一個錯誤偵測率會變動的軟體可靠度模型，因為實際測試時間越長，其錯誤留存率會越低，而如何找出一個較好的錯誤偵測率，可以作為往後研究者的課題。



## 參考文獻

### 中文部分

- 王亭雅 (2015)。具指數型、線性與 S 型特性之不完美除錯軟體可靠度模型。碩士論文，國立臺東大學 資訊管理學系，臺東縣。
- 王耀祥 (2015)。電動車用鋰電池組之可靠度分析。碩士論文，國立臺灣大學 機械工程學研究所，臺北市。
- 田靜婷 (2003)。影響高科技產業研發團隊學習績效相關因素之研究。博士論文，國立彰化師範大學 工業教育學系，彰化市。
- 朱家儀 (2013)。台灣地區之地震發生頻度分析--以群集卜瓦松過程為適配工具。碩士論文，東海大學 統計學系，臺中市。
- 江文馨 (2013)。具指數型及線性特性之不完美除錯軟體可靠度模型。碩士論文，國立臺東大學 資訊管理學系，臺東縣。
- 江姮臻 (2014)。考慮學習曲線下多重產品生產線之動態人力資源配置。碩士論文，國立清華大學 工業工程與工程管理學系，新竹市。
- 宋鎮照 (1997)。社會學。臺北市：五南出版。
- 李茂興(譯) (2001)。組織行為。臺北市：揚智文化。(Stephen. P. Robbins, 1999)
- 李嘉華 (2014)。具有除錯延遲與測試心力函數之不完美除錯軟體可靠度模型。碩士論文，國立臺東大學 資訊管理學系，臺東縣。
- 林正平 (2019)。工具機安全系統之可靠度研究。碩士論文，國立臺灣海洋大學 機械與機電工程學系，基隆市。
- 林宗煥 (2012)。複合卜瓦松過程在定期保養上的應用。碩士論文，國立新竹教育大學 應用數學系，新竹市。
- 邱人文 (2013)。影響學習曲線效率之人員因素探討。碩士論文，龍華科技大學商學與管理研究所，桃園市。

- 洪翠雲 (2007)。應用有限及無限排隊理論模型於測試與操作階段之軟體可靠度塑模。碩士論文，國立清華大學 資訊工程學系，新竹市。
- 徐銘鴻 (2008)。液晶顯示器產品電子元件可靠度預估方法與試驗之研究。碩士論文，國立成功大學 工學院工程管理專班，台南市。
- 祝英哲 (2013)。特定型態強度函數在以最大概似法估計卜瓦松點過程之計算敏感現象探討。碩士論文，東海大學 統計學系，台中市。
- 高婉蕙 (2012)。複合卜瓦松過程之可靠度計算應用。碩士論文，國立新竹教育大學 應用數學系碩士班，新竹市。
- 張春興 (2001)。教育心理學。心理出版社，新北市。
- 陳之薇 (2009)。影響醫師學習曲線之相關因子探討—以腹腔鏡膽囊切除手術為例。碩士論文，國立臺灣大學 公衛學院醫療機構管理所，臺北市。
- 陳世坤 (2005)。應用學習曲線建構人員職能評核模式。碩士論文，國立臺灣大學 電機工程學研究所，臺北市。
- 陳登寶 (2015)。可靠度預估試驗研究-以 LED T8 燈管產品為例。碩士論文，明新科技大學 工業工程與管理系碩士在職專班，新竹縣。
- 游卓群 (2012)。複合卜瓦松過程在離散時間定時保養的應用。碩士論文，國立新竹教育大學 應用數學系，新竹市。
- 楊宗展 (2019)。電動車動力系統之可靠度研究。碩士論文，國立臺灣海洋大學 機械與機電工程學系，基隆市。
- 謝爾廉 (2000)。具轉折點及多錯誤型態之非完美除錯軟體可靠度模型。碩士論文，淡江大學 資訊管理學系，新北市。
- cccarfield (2010)。可靠性概念。上網日期：2019 年 2 月 19 日。網址：  
<https://wenku.baidu.com/view/329d55eef8c75fbfc77db284.html>

## 英文部分

- Cai, K. Y. (2000). Towards a conceptual framework of software run reliability modeling. *Information Sciences*, 126, pp.137-163.
- Chondamrongkul, N., Sun, J., Wei, B., & Warren, I. (2018). Parallel Verification of Software Architecture Design. 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE).
- Eisty, N. U., Thiruvathukal, G. K., & Carver, J. C. (2018). A Survey of Software Metric Use in Research Software Development. 2018 IEEE 14th International Conference on e-Science (e-Science).
- Ellis, A. P., Hollenbeck, J. R., Ilgen, D. R., Porter, C. O., West, B. J., & Moon, H. (2003). Team learning: Collectively Connecting the Dots. *Journal of Applied Psychology*, 88, 821-835.
- Goel, A. L. & Okumoto, K. (1979). Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures. *IEEE Transactions on Reliability*, 28, pp.206-211.
- Hambrick, D. C. (1994). Top management group : A conceptual integration and consideration of the —teamll label. In B. M. Staw, & L. L. Cummins(eds.). *Research in Organizational Behavior*, 16, pp.171-213.
- Hoof, B. V., & Ridder, J. A. (2004). Knowledge sharing in context: The influence of organizational commitment, communication climate and CMC use on knowledge sharing. *Journal of Knowledge Management*, 8, pp.117-130.
- Huang, C. Y. and Lin, C. T. (2006). Software Reliability Analysis by Considering Fault Dependency and Debugging Time Lag. *IEEE Transactions on Reliability*, 55, pp.436-450.
- Huang, C. Y., Lyu, M. R. and Kuo, S. Y. (2003). A Unified Scheme of some

- Nonhomogenous Poisson Process Models for Software Reliability Estimation. IEEE Transactions on Software Engineering, 29, pp.261-269.
- Huang, C. Y., & Kuo, T.-Y. (2017). Queueing-Theory-Based Models for Software Reliability Analysis and Management. IEEE Transactions on Emerging Topics in Computing, 5, pp. 540–550.
- Ibarra, S., & Munoz, M. (2018). Support tool for software quality assurance in software development. 2018 7th International Conference On Software Process Improvement (CIMPS).
- Jiang, He., Li, X., Ren, Z., Xuan, J., Jin, Z. (2019). Toward Better Summarizing Bug Reports With Crowdsourcing Elicited Attributes. IEEE Transactions on Reliability., 68, pp.2-22.
- Kapur, P. K., Garg, R. B. and Kumar, S. (1999). Contributions to Hardware and Software Reliability. Singapore: World Scientific Publishing Co. Ltd.
- Kapur, P.K, Pham, H., Gupta, A. and Jha, P.C. (2011). Software Reliability Assessment with OR Applications. Springer-Verlag London Limited.
- Kasl, E., Marsick, V. J., & Dechant, K. (1997). Team as learner: A research-based model of team learning. Journal of Applied Behavioral Science, 33, pp.227-246.
- Lakshmanan, I., Ramasamy, S. (2015). An artificial neural-network approach to software reliability growth modeling. Procedia Computer Science 57, pp. 695 - 702.
- Li, Q., Pham, H. (1997). NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage. Applied Mathematical Modelling, 51, pp.68–85
- Liao, L. F. (2008). Knowledge-Sharing in R and D Departments: A Social Power and Social Exchange Theory Perspective. The International Journal of Human Resource Management, 19, pp.1881-1895.
- Nadler, D. A., Spencer, J. L., & Associates. (1998). Executive teams. San Francisco:

- Jossey- Bass, pp.4-5.
- Nagaraju, V., Fiondella, L., Zeephongsekul, P., Jayasinghe, C. L., & Wandji, T. (2017). Performance Optimized Expectation Conditional Maximization Algorithms for Nonhomogeneous Poisson Process Software Reliability Models. *IEEE Transactions on Reliability*, 66, pp.722–734.
- Ohba, M. (1984) . Software reliability analysis models. *IBM Journal of Research and Development*, 28, pp.428 - 443
- Pham, H., Zhang, X., (1997) . An NHPP software reliability models and its comparison. *International Journal of Reliability, Quality and Safety Engineering* , 4 , pp.269-282.
- Pham, H., Zhang, X. (2003) . NHPP Software Reliability and Cost Models with Testing Coverage. *European Journal of Operational Research*, 145, pp.443-454.
- Pham, H. (2006) . *System Software Reliability*. Springer-Verlag London Limited.
- Pham, H., Nordmann, L. and Zhang, X. (1999) . A General Imperfect-software-Debugging Model with S-Shaped Fault Detection Rate. *IEEE Transactions on Reliability*, 48, pp.169–175.
- Pham, H , (2000) . *Software Reliability*. Springer-Verlag London Limited
- Pietrantuono, R., Potena, P., Pecchia, A., Rodriguez, D., Russo, S., & Fernandez-Sanz, L. (2018) . Multiobjective Testing Resource Allocation Under Uncertainty. *IEEE Transactions on Evolutionary Computation*, 22, pp.347–362.
- Roy, P., Mahapatra, G. S., Dey K. N. (2014) AN NHPP SOFTWARE RELIABILITY GROWTH MODEL WITH IMPERFECT DEBUGGING AND ERROR GENERATION. *International Journal of Reliability, Quality and Safety Engineering*, 21.
- Stasser, G., & Titus W. (1987) . Effects of information load and percentage of shared information on the dissemination of unshared information during group discussion. *Journal of Personality and Social Psychology*, 53, pp.81-93.

- Teng, X., Pham, H. and Jeske, D. R. (2006) . Reliability Modeling of Hardware and Software Interactions, and Its Applications. *IEEE Transactions on Reliability*, 55, pp.571-577.
- Wang, J., Wu, Z. (2016) . Study of the nonlinear imperfect software debugging model. *Reliability Engineering and System Safety.*, 153, pp.180-192.
- William, M. J., John, T. D., DeBardleben, N. (2012) . Application monitoring and checkpointing in HPC: looking towards exascale systems. *Proceedings of the 50th Annual Southeast Regional Conference*, pp.262-267.
- Wood, A. (1996) . Software Reliability Growth Models. Tandem Computers, Technical report, pp. 96.
- Wright, T.P. (1936) "Factors Affecting the Cost of Airplanes", *Journal of Aeronautical Sciences*, 3(4), pp.122-128
- Yamada, S.(1992) . A stochastic software reliability growth model with Gompertz curve, *Trans.IPSJ* , 33, pp.964-969 .
- Yamada, S., Ohba, M., Osaki, S. (1983) . S-shaped reliability growth modeling for software error detection, *IEEE Trans, Reliab. R-32*, pp.475-478.
- Yamada, S., Osaki, S. (1985) . Software reliability growth modeling: Models and Applications,*IEEE Transaction on Software Engineering*, SE-11, pp.1431-1437.