

## IEEE/EIA 12207 程序本質與程序模擬

### IEEE/EIA 12207 Process Nature and Process Simulation

王德劭

Te-Shao Wang

元智大學資訊工程研究所

中壢遠東路 135 號

Yuan-Ze University, Chung-Li, Taiwan

范金鳳

Chin-Feng Fan

元智大學資訊工程研究所

中壢遠東路 135 號

Yuan-Ze University, Chung-Li, Taiwan

csfanc@saturn.yzu.edu.tw

#### 摘要

本文探討軟體生命週期各活動的本質、源起及相互關係，希望經由對本質的了解可發展進一步的應用。我們以 IEEE/EIA 12207 為藍本，以翻譯過程 (translation)、隱晦概念探索 (tacit knowledge exploration) 及團隊因素 (team) 三個方向來探究軟體程序的本質，再應用探討的結果於軟體程序的適裁 (tailoring) 上。除了上述靜態的分析外，本研究並發展了 IEEE/EIA 12207 的模擬工具，針對上述的分析進行模擬，且進一步探討軟體活動相關因素間的相互影響關係。

關鍵詞：生命週期程序、模擬、IEEE/EIA 12207、隱晦概念、適裁。

#### Abstract

This paper aims to investigate the intrinsic nature of software life cycle process activities as well as relationship among these activities so that further application can be developed. We discussed the nature of IEEE/EIA 12207 software process based on "translation", "tacit knowledge exploitation", and "team" aspects. The discussion results are utilized to perform process tailoring. Besides the analytical study, we provided dynamical analysis through simulation to further examine the interaction among related factors of software activities.

Keywords: life cycle process, simulation, IEEE/EIA 12207, tacit knowledge, tailoring.

#### 1 緒論

IEEE/EIA 1220 為最新的軟體生命週期程序的工業標準，整合了過去各種軟體標準。IEEE/EIA 1220 廣受各方的重視，然而因為這個標準仍然相當新，至今少有大型專案計劃應用的經驗。本研究探討軟體生命週期各活動的本質、源起及相互關係，希望見微知著，從本質的探討可發展更進一步的應用，例如專案活動的適裁 (tailoring)、排程與人力資源的選擇等，目的在

增進軟體計劃成功的機會。除了靜態本質的分析外，本研究亦以動態的軟體生命週期之模擬，以進一步研究軟體活動之間以及相關因素之間的相互關係，希望提供應用這套標準進行軟體專案計劃者一有效的工具，以預估軟體專案成本、品質、時程、人員等各方面的安排。

本文第 2 節簡介相關背景及研究，第 3 節深入探討 IEEE/EIA 1220 軟體程序本質，第 4 節提出我們的程序適裁經驗法則。第 5 節我們針對 IEEE/EIA 1220 特色建立模擬模型並發展一套模擬工具。第 6 節則為模擬工具及實驗結果，最後為結論。

#### 2 背景及相關研究

IEEE/EIA 12207 [5] 源自於 ISO/IEC 12207 [6]，其目的在於整合過去軟體工程的各種標準，並提供軟體系統專案的發展及管理一個通用性的骨幹架構。在結構上，IEEE/EIA 1220 包括主要軟體生命週期程序 (primary life cycle)、組織性 (organizational) 程序、支援 (supporting) 程序及附錄的適裁 (tailoring) 程序。主要軟體生命週期程序內含採購、供應、發展、操作、維護五個程序。支援性的生命週期程序內含文件、組態管理、品保、驗證、確認、審查、稽核與問題解決八個程序。組織性生命週期程序內含管理、基礎、改善、教育訓練四個程序。每個程序下皆有多項活動 (activity)，每項活動下又分為多項工作項目 (task)。IEEE/EIA 1220 總共定義了 17 個程序，74 個行為活動，231 個工作項目以及 84 份文件。

在內容上，整體 IEEE/EIA 1220 包括了三部分：IEEE/EIA 12207.0-1996 基礎標準)、IEEE/EIA 12207.1-1997 生命週期的資料導引) 即軟體文件、IEEE/EIA 12207.2-1997 (實踐程序之導引)。基礎標準部分中包涵了 ISO/IEC 12207 主要文字部份以及額外的附註解釋，而另外兩部分則包涵了 MIL-STD-498 中的指引 (guideline) 以及其資料項目描述。

IEEE/EIA 1220 的特色在於強調計劃與檢查的必要，因此它定義的活動皆具有計劃-執行-檢查-行動 (plan-do-check-act, PDCA) 的循環。同時 IEEE/EIA 1220 適用於任何生命週期模型。此外，IEEE/EIA 1220 也強調程序適裁 (tailoring) 的必要性。

相關研究上，軟體程序適裁方面的研究並不多，Software

Technology Support Center(STSC) 的程序管理技術小組於 199 年[2]曾經提出針對 MIL-STD-498 標準的軟體進行經驗法則式的程序適裁法。至於軟體程序模擬的研究則眾多, Hanakawa [3] 提出建立一個軟體發展程序的模擬模型。Kusumoto [7]提出利用 Petri-net 延伸的軟體發展模型來進行模擬。JPL 的 SEPS 計劃[9]設計了一個具有回饋能力的軟體發展程序的動態模擬。

### 3 軟體程序本質探討

軟體專案的活動,不僅有“工程”的部份,同時也涉及創作、認知的行為以及團隊合作的因素。因此我們提議以三方面來探討軟體活動的本質及起因:(1)以工程的觀點來看,軟體各階段的活動可視為翻譯過程。(2)以認知的觀點來看,軟體有些階段間的語意差距(semantic gap)頗大,即上階段有說不出的概念,下階段需加以探索,故有創造的成分,我們可視相關的活動為隱晦概念的探索活動(tacit knowledge exploration)。(3)大型專案為團體的活動,故許多軟體的活動緣起於克服團隊溝通的困難並增進合作的效果,於是團隊因素需要考慮。因此我們以翻譯過程(Translation)、隱晦概念探索(Tacit knowledge exploration)與團隊因素(Team)三個方向來探討軟體程序的本質,我們稱之為 3T 模型,其符號表達如下:

#### 3 T Model =

$$\Sigma(\text{Translation}, \text{Tacit knowledge exploration}, \text{Team work})$$

試述如下:

#### (1) 翻譯過程(Translation)

軟體生命週期階段如圖 1 所示。例如需求分析階段的工作將使用者需求的自然語言翻譯為需求階段的規格語言,設計階段的工作將規格語言翻譯為設計階段的設計語言等等。相鄰兩個階段的發展可視為翻譯過程。

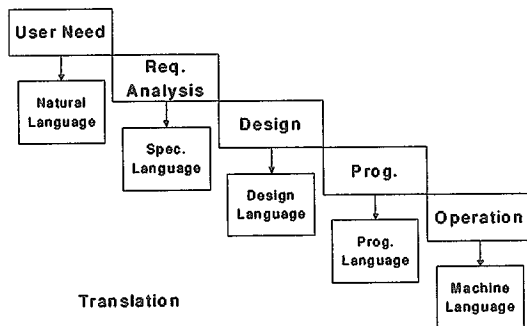


圖 1 翻譯過程

軟體生命週期各階段的活動可由狀態轉變圖來描述,後繼階段將較先前階段具有更多的不同的狀態。如圖 2 所示,因為上、下階段為翻譯(translation)或細化(refinement)過程,上、下階段的狀態轉變,應有相對應的部份。例如:

$$U=(U_1, U_2, \dots, U_n); \quad R=(R_1, R_2, \dots, R_m); \\ D=(D_1, D_2, \dots, D_n)$$

(U: user need; R: requirement; D: design;  $n \leq m \leq k$ )

$$R_i = \text{State}_i \xrightarrow{\text{Input}_i^R} \text{State}_i$$

$$D_i = P(\text{State}^D_i) \xrightarrow{P(\text{Input}^D_i)^*} P(\text{State}^D_i)$$

where p: power set

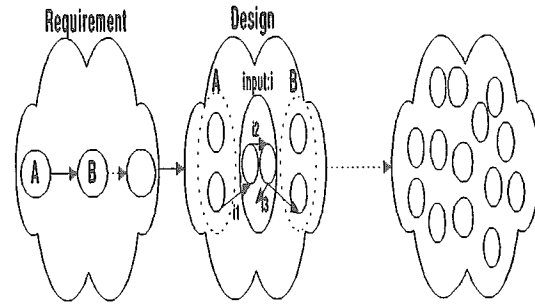


圖 2 翻譯過程內部

#### (2) 隱晦概念探索(Tacit knowledge exploration)

然而軟體的發展,並不能完全靠機械式的翻譯來完成,各階段間需要概念上的移轉。由於認知與表達上的問題,各階段的軟體需求仍有無法完全表達或了解的隱晦概念存在,因此軟體生命週期中許多活動都扮演著探索、發掘、並避免隱晦概念的角色,例如文件、審查、測試、離型、驗證、確認等活動都屬於這類活動。

#### (3) 團隊因素(Team)

大型的軟體專案由多人參與,團隊的合作,有助於專案的進行的,也相對產生溝通上的額外工作(overhead),且溝通過程可能出現許多溝通上的失誤,而影響專案的發展品質。團隊相關因素包括工作分配、計畫人員素質、經驗能力、以及人與人溝通上的可能失誤:

TeamWork =

$$(\text{project assignment}, \text{Member/team capability}, \text{Communication})$$

因為有團隊因素的存在,軟體發展過程中產生了許多為了克服團隊溝通上的困難並增進合作效果的活動。例如文件、整合測試、驗證等活動都屬於這類活動。

### 4 程序的適裁(Tailoring)

我們提出以下兩種適裁方式:

1. 經驗法則式。
2. 利用 3T 模型。

第一種方式先對軟體專案加以分類,然後再依照其特色找尋經驗法則,最後依照經驗法則,決定該類型軟體專案重要的行為活動以及工作項目。我們將結果列示於表 1。舉例而言,若是有有一個軟體專案,經過分析之後,歸納它屬於新型態軟體專案;然後由表 1 可知新型態軟體專案具備的特性,包括缺少參考資料、事前需求/分析較長、規格不易一次確定、必須學習相關專業知識、經常回頭修改、輔導上線時間長與單元發展時間長等現象。這些現象經由經驗法則可以推知,此類型專案需要加強描述詳細需求分析、雇用專業背景開發人員、進行詳細的驗證、進行詳細單元測試與進行發展人員教育訓練。最後由上述經驗法則,我們可以找出其相對應之行為活動,如表 1 所列,包括分析、需求驗證、離型系統、設計驗證等活動為專案計劃中必須特別強調的部份。

表 1 適裁經驗法則表

軟體專案類型	特點	經驗法則	相對應之重要行為活動
新型態軟體專案	缺少參考資料 事前需求 / 分析較長 規格不易一次確定 學習相關專業知識 經常回頭修改 輔導上線時間長 單元發展時間長	詳細需求分析 專業背景開發人員 詳細的驗證(verification) 詳細單元測試 發展人員的教育訓練	Analysis Requirement verification Design verification Prototype
架上軟體	具經濟效益 降低成本 縮短開發時間 詳細的操作手冊 快速修改	嚴謹的計劃與時程規畫 避免開發成員溝通不良 豐富的需求分析與設計 以測試取代其他品管方式 詳細操作手冊 使用者回應協助修改	Plan Testing Manual
軍規軟體專案	詳細操作手冊 經常維護 高安全(機密)性	詳細的操作程序 詳細的維護計畫與程序 高保密措施 嚴謹的品質管制 仔細的稽核	Operation Documentation QA Audit Safety Analysis C.M. Maintenance
安全關鍵軟體專案	高安全性 高整合度	安全分析 整合設計 完整測試 完整的文件 詳細計劃 IV&V	Safety analysis Plan Documentation Review Test Analysis Validation Formal method
即時控管軟體	正確性很重要 穩定性很重要 具系統相關專業知識 測試很多次	單元及整合測試 通訊及界面技術 穩定性測試 修改困難	Testing Timing Qualification Integrated

第二種方式則是應用上述軟體本質於適裁上。不同的專案類型，所需要偏重的活動項目不同，有些偏重在翻譯活動上，有些在概念探索創作上，有些則特別需要團隊的合作。我們可以用本質上的這三方向來挑選軟體行為活動及工作項目。首先我們分析每一個行為活動，以“重要”、“普通”與“不重要”三種程度等級定義其具有翻譯過程、隱晦概念探索以及團隊因素三種本質之程度。部份結果如表 2 所示。在應用上，同樣先將專案分類，再考慮其 3T 因素傾向，然後再進行活動以及工作項目的適裁。總之其步驟如下：

1. 列出各軟體專案活動中 3T 所扮演的重要性。
2. 應用軟體專案的類別，以考慮此專案應著重的

因素(翻譯過程、隱晦概念探索、以及團隊因素)。  
3. 挑選具偏重這些因素的軟體程序活動。

舉例而言，當我們分析要發展的軟體專案後，發現它屬於新型態軟體專案類型。因為此類型專案缺乏經驗以及可能也缺乏相關背景知識，所以在發展的過程中，特別容易出現隱晦概念，我們可以由表 2 參考列表挑選適切的行為活動或是工作項目，並在成本控制允許下，加重專案特別需要的行為活動或是工作項目的成本與努力。在此例中，參考列表 2 裏“系統需求分析”以及“軟體需求分析”兩項軟體程序活動在本質上，隱晦概念探索程度屬於重要等級，因此在適裁程序中，這些活動有較高的優先選擇性。

表 2 軟體活動 3T 程度表

Activity	Ta.	Tr.	Te.
System requirement analysis	Imp	Imp	Mod
System architecture design	Mod	Imp	Mod
Software requirement analysis	Imp	Imp	Mod
Software architecture design	Mod	Imp	Mod
Software detail design	Mod	Imp	Mod
Software coding and testing	Mod	Mod	Imp
Software integration	Mod	Mod	Imp
Software qualification testing	Mod	Imp	Imp
System integration	Mod	Mod	Imp
System qualification testing	Mod	Imp	Imp
Software installation	Mod	Mod	Imp
Software acceptance support	Mod	Mod	Imp
Documentation	Imp	Mod	Imp
Configuration management	Imp	Mod	Imp
Qualification assurance	Imp	Mod	Mod
Verification	Imp	Imp	Mod
Validation	Imp	Imp	Mod
Joint review	Imp	Mod	Imp
Audit	Imp	Mod	Imp
Problem Resolution	Mod	Mod	Imp

Imp: importance, Mod: moderate

Tr: translation, Ta: tacit knowledge exploration, Te: team

### 5 軟體程序模擬

我們進一步以動態的模擬來分析軟體的程序。此節將說明我們發展的模擬模型以及模擬工具。模擬的目的為：  
(1)進一步動態的探討前述的本質問題。(2)希望能夠動態的瞭解生產力、缺失與溝通等因素間互動關係。(3)希望瞭解發展及支援程序之行為活動以及工作項目間 依 (dependency) 關係。(4)動態的檢查前述適裁經驗法則。

我們所建的軟體模擬模型細分為五個副模型，分別為主要生命週期程序行為活動模組、支援程序行為活動模組、軟體專案參與人員模組、品質指標缺失計算模組與生產力指標工作量計算模組。如圖 3 所示。每個模組本身都具有表達本身特性的各項屬性值(property)，以及展現本身行為的方法(method)。我們把每一個軟體程序視為一個物件，相似的程序物件歸納為同一類別。

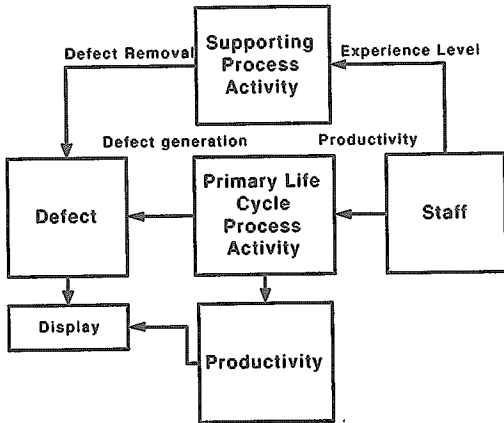


圖 3 IEEE/EIA 12207 模擬模型

我們發展的模擬工具以物件導向方式實作，亦提供 IEEE/EIA 12 2 0 電子檢索。此模擬工具使用 Microsoft Visual FoxPro Ver. 5.0 開發，其部份功能將利用 Microsoft Graph9 當作 OLE Server，整個系統在個人電腦單機運作，可在 Windows 95/9 或 Windows N 下執行。工具輸入方式如圖 4 所示。使用者可以在表格中進行軟體活動的各屬性設定。另外亦提供參與人員物件屬性設定的畫面，使用者可設定參與人員物件屬性，包括人數、經驗(專家程度、一般程度、新手程序)、標準生產力(productivity)、飽和生產力與缺失(defect)產生率。

模擬中各模組的輸入及運作如下：

主要生命週期程序活動模組 (Primary life cycle process activity module) 計算發展、操作、維護各階段的生產力

圖 4 程序屬性設定物件化畫面輸入模式功能畫面

及缺失變化。當一個行為活動完成至某一程度 (即百分比或飽和點) 後，因整合和困難度的增加，其單位時間的生產力，將大幅度的衰退。為模擬此現象，在此模組中我們也設定該行為活動超過飽和狀態的生產力。不可避免的，計畫參與人員在計劃進行中會產生失誤；而且有一些失誤則導因於前一程序未進行，例如沒有對需求分析，即進行架構設計所導致的失誤。因此我們也設定義相關的屬性值。除此之外，本模組的屬性還包括了每一個行為活動之工作總量，以及時程計畫的安排 (schedule)。

支援程序行為活動模組 (Supporting Process Activity Module) 計算文件、組態管理、品保、驗證、確認、審查、稽核與問題解決等活動對缺失移除的效果。

軟體專案參與人員模組描述主要生命週期以及支援程序中參與人員的不同特性，使用者並可依計畫需要更改其設定。其屬性包括不同經驗等級程度的生產力、超過飽和生產力、缺失與缺失移除等項的變化幅度 (variation)。

品質指標缺失計算模組的目的在於計算缺失總量，並將缺失轉換工作負荷量。

生產力指標工作量計算中心模組主要功能在計算剩餘工作總量，並監管計劃進度。輸入設定的屬性包括時程與及時警示的檢查點。

系統中的程式運算主要邏輯在生產力、剩餘工作量以及缺失產生部份之運算，如圖 5、圖 6 所示。本工具提供手動模擬及自動模擬，其主要模擬畫面如圖 7 所示。

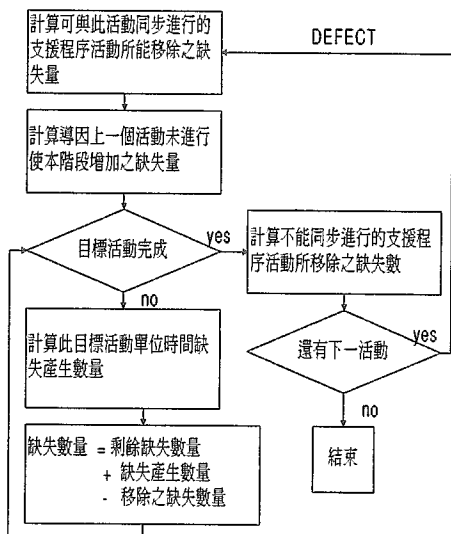


圖 5 缺失數量產生與移除的流程

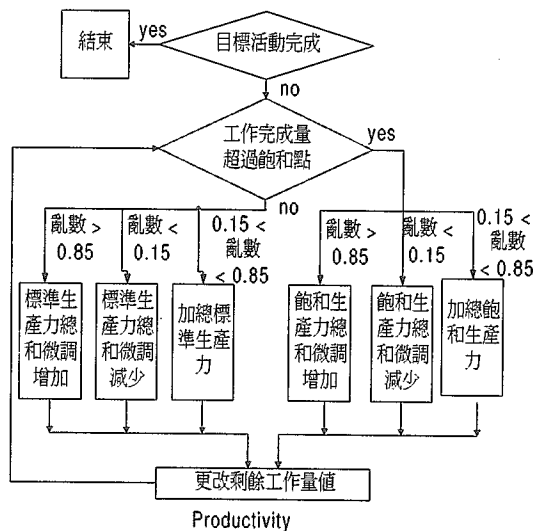


圖 6 生產力與剩餘工作量計算流程

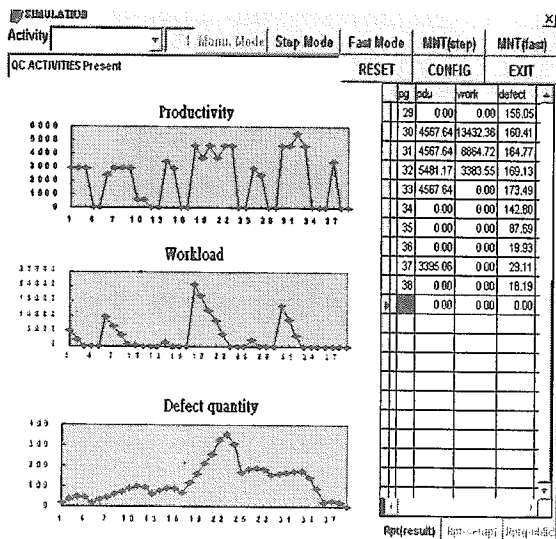


圖 7 模擬功能畫面

## 6 模擬結果

我們針對生產力、人員、缺失等因素進行模擬實驗，並比較 IEEE/EI 12207 與其他的標準，以及前述適裁法則的效率。實驗結果如下：

### (1) 參與人員數量與缺失產生的關係

團隊溝通的因素會造成缺失數量的增加，我們由公開的文獻資料[10]得到團隊組織溝通的因素對缺失數量的關係如下：

$$T(Q_{total}) = 1.03 * e^{(-0.02 * Q_{total})}$$

$T(Q_{total})$ 代表團隊溝通因素對缺失數量的影響函數。模擬結果如圖 8 所示。我們可以發現缺失產生隨參與人員數量的增加而增加，而且參與人員數量增加得越多，缺失產生的數量就越明顯。這現象反應團隊溝通所造成的影響。

### (2) 參與人員數量與生產力的關係

我們定義軟體行為活動程序的生產力總和如下：

$$P_{total} = (\sum_{i=expert\ level}^{novice\ level} average\ level * V_i) * C(Q_{total})$$

公式中  $P_{total}$  代表生產力總和，而  $P_{standard}$  代表標準生產力， $V_i$  代表生產力的變化率， $Q$  則代表人數， $C(Q_{total})$  則代表與參與人員數量有關的團隊性質溝通因素函數，其中團隊性質溝通因素函數我們使用的公式如上所述（即  $C(Q_{total}) = 1.03 * e^{(-c * Q_{total})}$ ），但我們更改自然底數  $e$  之指數部份  $c$  為 0.02、0.08、0.15、0.2 四種情形。模擬的結果顯示在圖 9。

由圖 9 生產力與發展參與人員關係的模擬，我們可以發現，當參與人數開始增加時，生產力總和的確快速增加，但是當參與人數增加到一定程度時，生產力總和就開始緩慢降低了，如前項實驗這種現象也說明了團隊溝通因素造成生產力的負面效應。

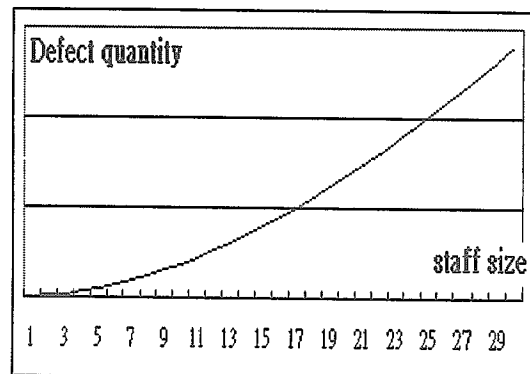


圖 8 缺失產生數量與發展參與人員人數關係

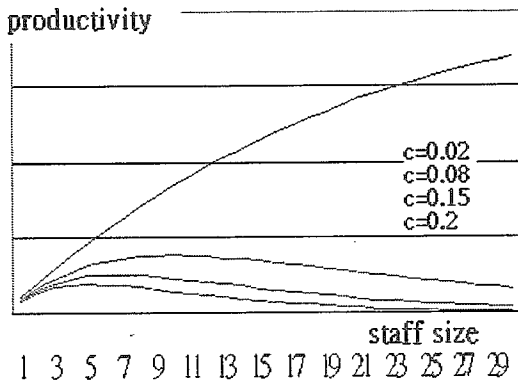


圖 9 生產力與發展參與人員人數關係

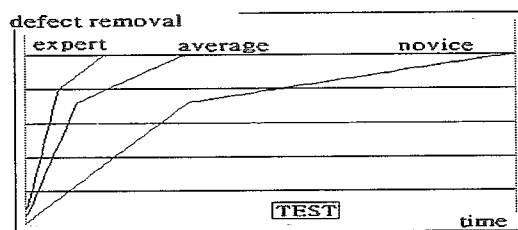


圖 10 缺失移除能力與參與人員經驗能力的關係(測試)

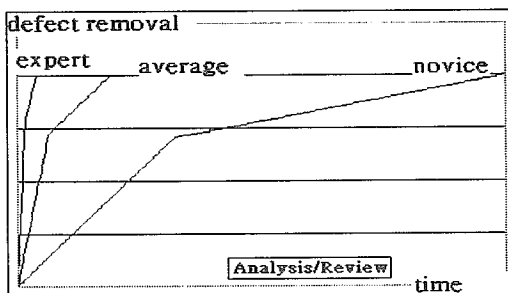


圖 11 缺失移除能力與參與人員經驗能力關係(分析/審查)

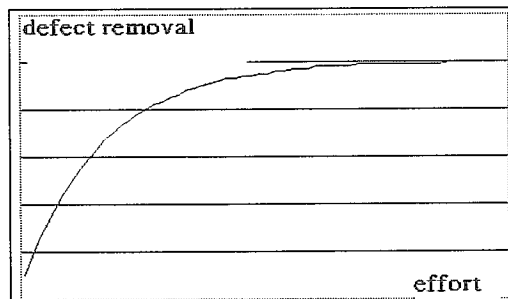


圖 12 投入之努力與缺失的移除

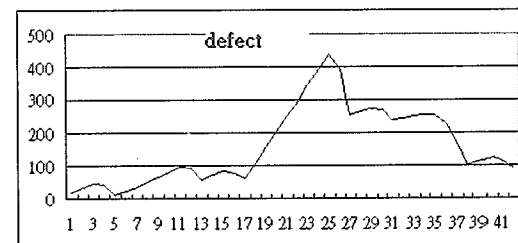
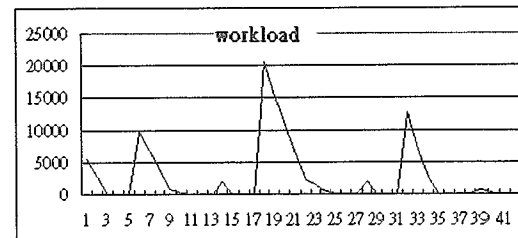
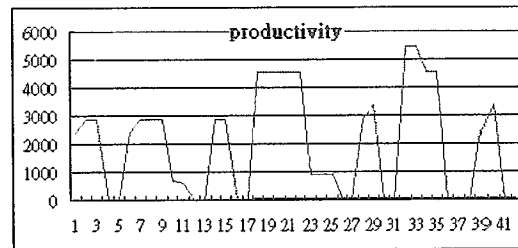


圖 13 IEEE/EIA 122 模擬範例

(3) V&V 參與人員經驗能力與缺失移除的關係

我們將針對以上兩項因素在 IEEE/EIA 122 支援程序活動中挑選分析(analysis)/審查(review)、測試(test)兩種軟體支援程序，並以專家、一般與新手三種不同程度等級的參與人員，進行對缺失移除量的模擬，模擬的結果分別展示於圖 10、圖 11。

由以上模擬的結果我們可以發現參與人員的經驗能力越高則其缺失移除的效率也越高。圖形中的轉折點代表 V& 活動超過某個百分比(飽和點)之後，缺失移除效率的下降情形。在我們的設定值中，假設專家對分析/審查的工作績效(缺失移除率)遠大於一般程度人員及新手，然而相較之下測試較不需要經驗，故專家、一般程度人員及新手的工作績效差異較小。因此圖 10 三種不同經驗的人員間之績效差異小於圖 11。

(4) 缺失移除能力(defect removal 與投入努力(effort)的關係

V& 活動中投入的人力程度與其實際移除的缺失量，將隨時間的增加呈現指數分配的圖形。如圖 12 所示。活動剛開始，移除缺失的能力隨著人力投入不斷快速增加，然而隨著缺失移除量的增加，存在的缺失越來越少，同時潛在的缺失也越來越難移除，因此移除缺失的效率漸漸趨於平緩，但是不同性質的 V& 活動，其移除缺失的效率趨於平緩的時機也各有不同，端視該支援程序對缺失移除的效率而定。

(5) IEEE/EIA 122 模擬範例

圖 13 為一 IEEE/EIA 122 模擬範例的結果。(橫軸皆為時間，單位為週數)。我們循序由一個階段至另一個

階段的模擬。圖 13 有三個子圖，由上而下分別是單位生產力與時間的關係、工作剩餘量與時間的關係、缺失與時間的關係。由上方生產力與時間進程的關係圖可以得知生產力在各個波段的最高點線段，代表在正常情況下的生產力，而在各個波段下降至零之前有較低的生產力線段，代表超過生產力飽和點後情況下的生產力，而生產力為零的部份則是我們模擬一個軟體程序結束到下一個軟體程序之間所進行的支援活動。由中間工作剩餘量與時間進程的變化圖，我們可以清楚看見每一個軟體程序的開始與結束，有些軟體程序工作量較長，有些較短，端視該程序的工作總量以及參與人員的總生產力而定。由下方缺失產生與移除與時間進程的變化圖形我們可以發現，每一個軟體程序的缺失都是緩緩的產生，但是在新的程序剛剛開始時，缺失的發生將快速增加，這種現象是因隱晦概念因素造成的影響。在該軟體程序快結束的時候，其缺失移除量就大幅增加，這是因為類似審查、驗證等支援程序開始進行並大量移除缺失的緣故。

#### (6) IEEE/EIA 12207 與 BTP14 軟體專案模擬結果及分析

我們希望利用模擬的方式將 IEEE/EIA 12207 與其他軟體工程的標準作一個比較，並分析其優缺點。本研究中我們選擇的比較標準是 BTP14 [4]。BTP14 [4] 為提供審核能數位控制系統的軟體生命週期程序的導引。其主要生命週期程序之活動包括了，計劃、需求、設計、實作、整合、驗證、安裝、操作與維護，大致上與 IEEE/EI 12207 的主要生命週期程序相同。但是比較起來 IEEE/EIA 12207 程序較為完整。BTP14 在支援程序之活動則是強調軟體安全分析、型態管理與驗證，而 IEEE/EIA 12207 支援程序分得較仔細並強調適裁。

我們將兩個標準都取其完整的主要生命週期程序之活動行為進行模擬，而且設定所有條件完全相同，不過在支援程序上，BTP14 僅挑選型態管理與驗證程序，其他與 IEEE/EIA 12207 有關的支援程序則隨機挑選，而 IEEE/EIA 12207 則不堅持挑選固定的支援程序行為活動，一切都隨機進行。IEEE/EIA 12207 正常程序運作與 BTP14 正常情況下模擬的比較結果如圖 14。

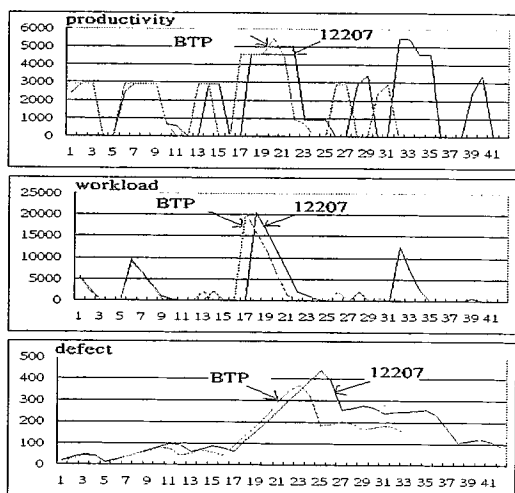


圖 14 IEEE/EIA 12207 與 BTP14 模擬比較圖

由以上模擬我們可以發現，BTP14 正常情況下模擬，整個軟體專案進行所花的時間比 IEEE/EIA 12207 正常情

況下模擬為短，這是因為完整的 BTP14 主要生命週期程序之行為活動，較完整的 IEEE/EIA 12207 主要生命週期程序之行為活動為少。同時我們也可以發現，BTP14 正常情況下模擬所發生的缺失，較 IEEE/EI 12207 正常情況下模擬多，這是因為 IEEE/EIA 12207 的支援程序較 BTP14 豐富，然而缺失的移除主要是因為支援程序的工作緣故。

#### (7) 軟體專案適裁經驗法則模擬結果及分析

我們對特定性質的軟體專案類型，進行依照前述我們所提出的軟體專案適裁經驗法則與不依照軟體專案適裁經驗法則加以模擬分析。模擬個案中我們選擇即時控管軟體專案。依據前述適裁經驗法則，我們知道即時控管軟體專案的特性包括許多測試的要求以及與系統整合的要求，依照經驗法則，此類型軟體專案必須強調單元與整合測試、品質驗證及整合，因此模擬的條件兩者並不相同，依照軟體專案適裁經驗法則進行的軟體專案將重視經驗法則適裁的程序，同時適裁經驗法則所強調的軟體程序之參與的人數以及經驗能力也將較多較強，代表我們將強調該軟體程序。反之不依照軟體專案適裁經驗法則進行的軟體專案，將不重視經驗法則適裁的程序，包括程序以及其參與的人數，經驗能力都隨機挑選，初步的模擬的比較分析結果如圖 15 所示。

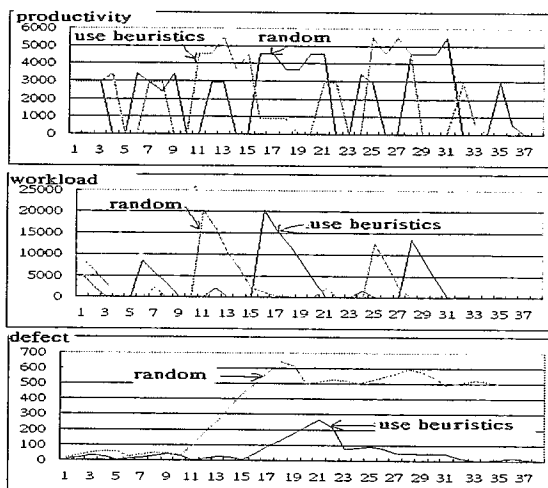


圖 15 依照適裁經驗法則與隨機挑選程序的模擬

生產力與時間的關係以及工作量剩餘與時間關係兩者並無顯著的差異，但是依照軟體專案適裁經驗法則所產生的缺失量遠小於不依照適裁經驗法則的軟體專案。

本模擬現可提供軟體專案的管理及發展者分析與比較缺失和生產力兩項指標與影響專案人數、經驗與活動等因素的關係。我們的目的是在於探討軟體發展程序的各項活動的行為相互關係。實驗結果並可應用於軟體程序的適裁、排程、本質探討各方面，我們的觀察分述於下。

- (1) 參與人員的增加若超過一定數目將產生許多額外的工作負擔並影響整體生產力，並因溝通問題產生缺失。
- (2) 參與人員經驗能力影響軟體活動成效，可針對不同素質的組員，挑選合適的 V& 活動。

(3) 支援程序對於缺失移除呈現指數分配圖形，因為缺失移除愈至後期愈困難。

(4) 隱晦概念以及團隊組織溝通等因素影響各種軟體程序活動。

(5) IEEE/EIA 1220 較其他軟體程序標準要完整複雜，故我們所建構的 IEEE/EIA 1220 模擬工具可用以模擬其他軟體標準。

(6) 實驗結果顯示我們的適裁經驗法則的確有效。

## 7 結論

軟體本質的探討有助於我們對軟體行為的瞭解，而由此瞭解可發展進一步的應用。例如決定軟體程序適裁的法則，專案的排程，人力資源的選擇等。而軟體模擬則為動態的方式深入了解軟體各行為活動的本質、相互的關係影響、對生產力、缺失與人員素質的關聯性，將有助於對軟體專案發展與管理。

本文以翻譯過程、隱晦概念探索、團隊因素三方面探討軟體程序的本質，並提出軟體程序適裁經驗法則供應用 IEEE/EIA 12207 的人們作為發展及管理軟體專案的參考。動態上發展 IEEE/EIA 1220 軟體程序模擬模型及工具幫助使用者預估最佳化的專案發展管理方式。未來可朝量化適裁方法及專案績效評估上繼續研究。

## 參考文獻

- [1] Victor Basili, Scott Green, "Software Process Evolution at the SEL", IEEE Software, Vol.11, No.4 pp.58-66, July 1994.
- [2] Fay C. Budlong, Paul A. Szulewski, Ralph J. Ganska, "Process Tailoring for Software Project Plans", Version 1.04, STSC, Utah, January 1996.
- [3] Noriko Hanakawa, Syuji Morisaki, Ken-ichi Matsumoto, "A Learning Curve Based Simulation Model for Software Development", International Conference of Software Engineering, pp.350-359, 1998.
- [4] NRC Branch Technical Position HICB -14, v15, Guidance on Software Reviews for Digital Computer-based Instrumentation and Control System 1999.
- [5] IEEE/EIA 12207, Industry Implementation of International Standard ISO/IEC 12207:1995, (ISO/IEC 12207) Standard for Information Technology-Software life cycle processes, April, 1998.
- [6] ISO/IEC 12207, Information technology-Software life cycle processes, 1995.
- [7] Shinji Kusumoto, Osamu Mizuno, Tohru Kikuno, Yuji Hirayama, Yasunari Takagi, Keishi Sakamoto, "A New Software Project Simulator Base on Generalized Stochastic Petri-net", Proceedings of 19<sup>th</sup> International Conference on Software Engineering, pp. 293-302, 1997.
- [8] Shari Lawrence Pfleeger, Joseph C. Fitzgerald Jr., Dale A. Rippey, "Using multiple metrics for analysis of improvement", J. Software Quality Vol. 1, pp27-36, 1992.
- [9] Chi Y. Lin, Tarek Abdel-Hamid, Joseph S. Sherif, "Software-Engineering Process Simulation Mode (SEPS)", J. SYSTEM SOFTWARE, 38, pp.263-277, 1997.
- [10] Tausworthe, R. C., "Information Models of Software productivity: Limits on Productivity Growth" J.SYSTEM SOFTWARE, 21, pp.185-201, 1992.