

陳淑珍 編譯

軟體工程入門



全華科技圖書股份有限公司 印45
世界圖書出版公司 印

陳淑珍 編譯

軟體工程入門



全華科技圖書股份有限公司 印行
世界图书出版公司重印

软体工程入门

陈淑珍 编译

全华科技图书股份有限公司出版

世界图书出版公司 重印

(北京朝内大街 137 号)

北京中西印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

1991年2月第1版 开本: 711 × 1245 $\frac{1}{4}$

1991年2月第1次印刷 印张: 13

印数: 001—880 字数: 27.4万字

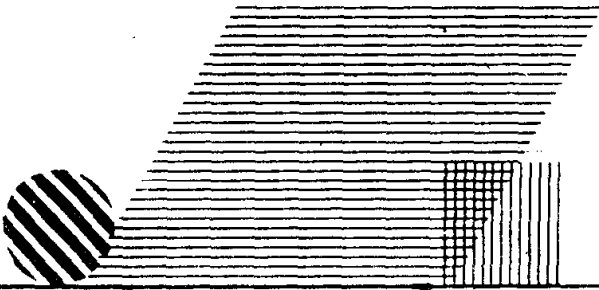
ISBN 7-5062-0837-7/TP·15

定价: 6.20元

本书经全华科技图书股份有限公司香港和中国大陆总代理

香港出版社有限公司特许世界图书出版公司独家重印

限国内发行



原 序

在1960年代的後期，當軟體工程的觀念首度被提出來時，看起來像是一種急進的思想。畢竟，程式規劃比較像是一門藝術而不像是一門工程學科，而軟體經常祇是略勝於事後的反思。

今日，幾乎每個人都承認軟體具有駕馭高科技系統的力量。然而，我們在發展軟體的過程中仍然繼續面臨著處理的問題。軟體發展者（無論是大學肄業生或是形容枯槁的老手），力圖建立可符合顧客需要的高品質程式。程式經常無法照預定時程交付，有時候是不可靠的，而且經常不容易修改。但是，當軟體工程科學的應用漸漸抬頭以後，事情開始改觀。

工業界在承攬大規模的軟體計畫時，會有規則地應用軟體工程。無論是在銀行業系統或機器人的應用，航空電子工程或醫療使用儀器上的應用，以有訓練的作法發展軟體是絕對必要的。然而，有關軟體工程應用性的錯誤觀念却是四處橫溢。由於大多數軟體工程被進行於主要的工業計畫，許多人相信軟體工程的作法祇適用於非常大的程式，沒有比這個更為錯誤的事！

如能明智地應用並配合應用而做最好的修改，軟體工程作法可應用於任何大小的程式。我們在分析一個問題，設計一種可信賴的解答，有效地製作程式並徹底加以測試時所使用的方法，對一個正在進行兩個星期的家庭作業的大學生和

一個正在建立一個新的計算機為基系統的實習軟體工程師而言，都是同樣的有價值。事實上，一個人愈早學習軟體工程方法，愈能有效地解決軟體問題。

本書的目的在於逐步詳細地介紹軟體工程，使大一或大二的學生，或剛進入工業界的專業人員可以容易地加以應用。本書並不對主題進行意義廣泛的論述，而以一系列的步驟和原理呈現出主題，使學生可以成功地應用於完成任何軟體計畫。事實上，所有的步驟可以組織成一張核對表，可供學生和教師使用，以確定所有重要的軟體工程活動都已完成。

本書可作為任何程式規劃導論課程的補充教材，也可作為軟體工程導論課程的主要教材。由於本書獨立於程式語言，因此可以有效地使用於任何的程式語言教學。本書亦可應用於任何需要軟體計畫工作，且教師要求應用軟體工程原理的課程。

需要重視的是軟體工程作法決不會減少學生的創造能力。事實上，本書對問題的表示法、解答與製作提供了有系統而合理的作法，排除了許多困擾、錯誤的開始以及經常伴隨軟體發展而導致的挫敗，可因此而加強學生的創造力。

本書與一般的教科書略有不同。除了第一章之外，每一章都介紹了軟體工程處理的一個主要活動。以一系列的步驟介紹系統工程、問題分析、設計、製作程式、測試及維護。在每一章之內，使用一系列的步驟、子步驟、原理與概念來描述一種軟體工程活動。提出單一的作法（方法論），以取代各種替換性作法的研究，將使學生在課程的應用上使用有效的模型。本書尚包含了三個附錄：①本書所介紹的所有步驟的核對表，可用於追蹤軟體工作；②介紹軟體品質保證技術，強調正式技術複查，以及③一個完整的例子，解說所有軟體工程步驟的應用。

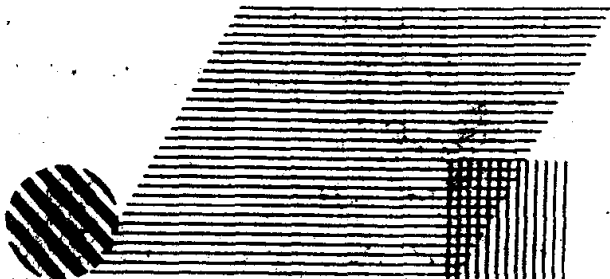
本書的展開工作開始於第二版的軟體工程專業途徑（

Software Engineering : A Practitioner's Approach , SEPA) 完成的時候。SEPA 對軟體工程技術有更廣泛的論述，已為大學的高級課程和專業訓練／參考所廣為採用。雖然本書和 SEPA 在大小、形式及風格上是完全不同，本書充分地借用 SEPA 一書的內容；包括闡述性討論、例子及許多圖畫的切除、修改、重組。

任何在軟體工程論著的作者如果不感謝成千上百對此類論著有長期貢獻的人們，將會有虧於職守。這些教育者、研究員以及工業界的從業人員已經影響了我對軟體工程的感知，而且毫無疑問地，這些都已塑入本書。同時，我要對本書的復查人員：Scranton 大學的 John Beider、Seattle 大學的 Robert Glass、Colgate 大學的 Allen Tucker 表示謝意，他們建設性的批評和建議都是非常可貴。最後，我在 Bridgeport 大學多年的教授生涯，使我能洞察學生在吸收與應用新觀念的最佳方法。本書所採用的作法，有許多來自那些曾經上過我在軟體工程及相關主題的課程的學生。

寫一本書，同時要兼顧課堂上的講學以及一個成功的顧問業務，須要面臨相當的挑戰。我的家人 Barbara、Mathew 和 Michael 提供了我許多的支持，以面對這些挑戰。所有這些，謹向我親愛的人們誌謝。

Roger S. Pressman



譯者序

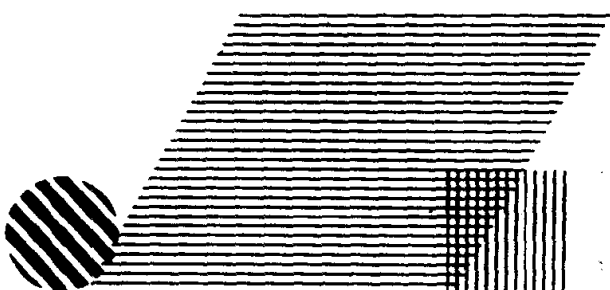
在數位電子計算機的發展史上，1950 與 1960 年代可說是硬體的進展時代，其後我們所面對的則是軟體時代的來臨。事實上，當今計算機的進展常受限於無法有效的生產品質良好的適用軟體。

近幾年來，無論是國內或國外，普遍面臨著軟體需求量大幅度成長而且更為複雜化的趨勢，而這種趨勢亦將隨著 1990 年代的到來而持續不衰。軟體需求雖然日益迫切，然而軟體從業人員的生產力却始終無法有效的提高，軟體經常無法在預定時程內完成，軟體品質亦無法有效的獲得保證。在計算機市場上，我們所看到的是，硬體設備的功能愈來愈進步且價格愈來愈便宜，相對的，軟體則是逐漸昂貴且甚而供不應求。因而，有所謂的軟體危機的存在。

軟體工程技術即是為了解決上述軟體問題因應而生。本書“軟體工程入門 (Software Engineering: A Beginner's Guide)”是作者 Roger S. Pressman 先生根據他在學術界及工業界從事軟體工程多年的經驗，為初學者所著述。本書首先介紹軟體的角色，從探討軟體在工業界所造成的問題，而強調軟體工程的重要性。其後，根據軟體工程的幾個主要活動，依序介紹系統、分析、設計、製作及維護。在每個主要的軟體工程活動中，簡要的陳述各個步驟、概念與原理。

本書內容富有見地且相當平易近人，不但能概觀軟體工程的全貌，而且易學易用。本書的編譯，除了力求忠實於原著的內容，在章節段落上亦做了清楚的劃分以期增加可讀性。匆匆完稿，難免有誤，尚祈各位先進，不吝指正。

譯者謹識



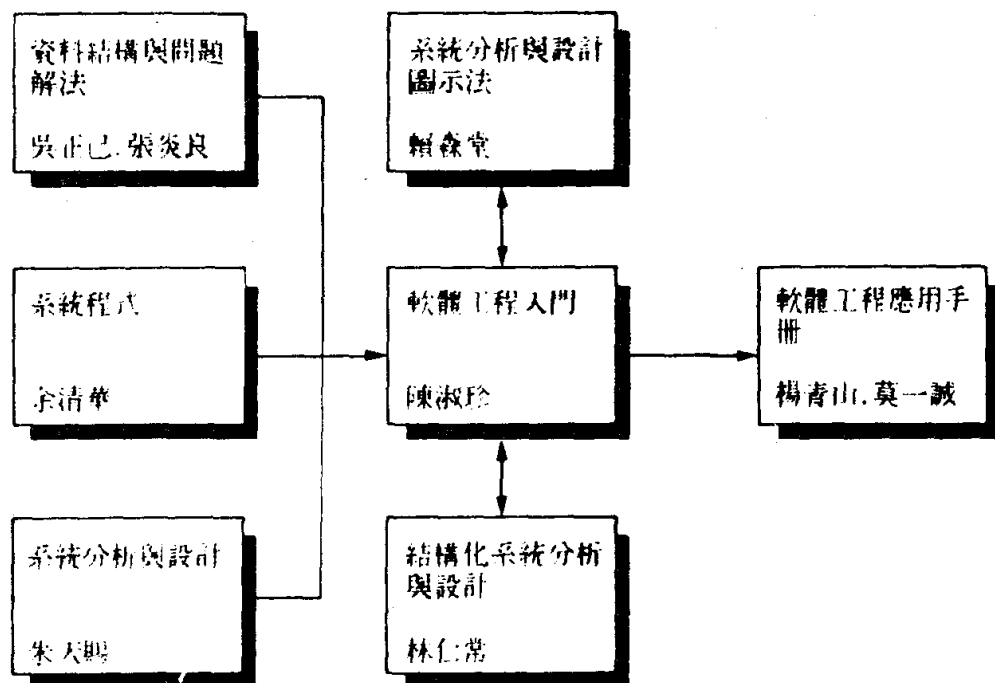
編輯部序

「系統編輯」是我們的編輯方針，我們所提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

「軟體工程」是最近越來越受到重視的一門學科。由於資訊時代的潮流所致，各行各業依賴電腦處理事務的比例急速增加，因而軟體品質的好壞亦愈顯重要，甚至關係企業的成敗。本書正針對「如何確保軟體品質」提出一套相當精闢及具體的方法，並使用一系列的步驟、子步驟、原理和概念來描述軟體工程實際的運作內容，全篇簡明易讀，措詞流暢，堪稱一本相當優秀的入門參考書。

同時，為了使您能有系統且循序漸進研習相關方面的叢書，我們以流程圖方式，列出各有關圖書的閱讀順序，以減少您研習此門學問的摸索時間，並能對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠為您服務。

流程圖





目錄

1	軟體的角色	1
1.1	何謂軟體	3
1.1-1	軟體特性	4
1.1-2	軟體應用領域	7
1.1-3	軟體危機	10
1.1-4	1990年代的課題	10
1.2	軟體工程應用模式	11
1.2-1	軟體工程：定義	12
1.2-2	傳統生命週期	13
1.2-3	原型製造	15
1.2-4	第四代技術	17
1.2-5	結合應用模式	19
1.3	軟體工程一般展望	20
1.4	結 論	23
1.5	給讀者的話	24
1.6	進修讀本	24
1.7	問 題	24
2	瞭解系統	27
2.1	系統範圍敘述	29
2.1-1	瞭解整體系統功能	29

2.1-2	描述整體系統功能	30
2.1-3	主要輸入與輸出	32
2.1-4	系統限制	33
2.1-5	平易的文字段落	35
2.2	上層處理與實體	36
2.2-1	選擇實體	36
2.2-2	選擇處理	39
2.3	分配	41
2.3-1	分配準則	41
2.3-2	替代性分配	42
2.3-3	替代性分配的適存性	44
2.3-4	選擇一種分配	45
2.4	改良範圍敘述	47
2.5	複查分配	48
2.6	反 覆	48
2.7	給讀者的話	49
2.8	進修讀本	49
2.9	問 題	49

3 分析問題 51

3.1	軟體範圍	51
3.2	目標與操作	53
3.2-1	識別目標與操作	55
3.2-2	資訊發源者與接收者	58
3.2-3	資訊儲存	59
3.2-4	資料與控制項目	59
3.2-5	資料與控制變換	59
3.3	三基本分析原理	61
3.3-1	分析原理 1	62

3.3-2	分析原理 2	63
3.3-3	分析原理 3	65
3.4	發展原型	66
3.4-1	良好的原型候選者	67
3.4-2	簡略的需求表示法	67
3.4-3	紙上原型	68
3.4-4	簡略的原型設計	68
3.4-5	建立原型軟體	69
3.4-6	試用原型	69
3.4-7	反覆進行	70
3.5	軟體流程模式	70
3.5-1	資料流程圖	71
3.5-2	控制流程圖	75
3.5-3	流程模式評估規則	78
3.6	資料字典	78
3.7	處理敘述	81
3.8	軟體確認	83
3.9	建立軟體規格	84
3.10	複查軟體規格	86
3.11	預期反覆	87
3.12	給讀者的話	87
3.13	進修讀本	88
3.14	問題	88

4 設計解答 91

4.1	改良軟體規格	91
4.2	基本設計概念	92
4.2-1	設計概念 1	92
4.2-2	設計概念 2	92

4.2-3	設計概念 3	94
4.2-4	設計概念 4	97
4.2-5	設計概念 5	99
4.2-6	設計概念 6	100
4.2-7	設計概念 7	102
4.2-8	設計概念 8	102
4.2-9	設計概念 9	104
4.3	資料設計	109
4.3-1	選擇資料結構	112
4.3-2	簡化資料結構	112
4.3-3	內部資料結構	114
4.3-4	資料庫管理系統	114
4.4	架構設計	115
4.4-1	改良流程模式	115
4.4-2	流程模式之特性	119
4.5	變換對映	121
4.5-1	變換中心	121
4.5-2	第一層因素化	123
4.5-3	第二層因素化	126
4.5-4	模組之處理敘述	128
4.5-5	第一切程式結構	128
4.6	交易對映	130
4.6-1	交易中心	133
4.6-2	DFD對映	134
4.6-3	交易結構	135
4.7	程序設計	136
4.7-1	結構化程式設計原理	138
4.7-2	改良處理敘述	141
4.7-3	程式設計語言	141

4.8	預先測試策略	148
4.9	建立設計文件	148
4.10	複查設計文件	151
4.11	給讀者的話	152
4.12	進修讀本	153
4.13	問題	153

5

製作程式

5.1	製作指引	158
5.2	模組製作次序	158
5.2-1	製作模組	160
5.2-2	全盤資料結構	163
5.2-3	內部資料結構	164
5.2-4	資料檔案	165
5.3	程式語言原始碼	166
5.4	程式製作技巧	168
5.4-1	程式製作規則 1	169
5.4-2	程式製作規則 2	169
5.4-3	程式製作規則 3	169
5.4-4	程式製作規則 4	172
5.4-5	程式製作規則 5	172
5.4-6	程式製作規則 6	174
5.4-7	程式製作規則 7	175
5.5	複查原始碼	176
5.6	給讀者的話	176
5.7	進修讀本	177
5.8	問題	177

6	測試結果	179
6.1	測試計劃	180
6.1-1	模組測試	181
6.1-2	整合策略	181
6.1-3	整合模組	182
6.1-4	驗證測試策略	185
6.2	基本測試原理	186
6.2-1	測試原理 1	186
6.2-2	測試原理 2	186
6.2-3	測試原理 3	187
6.2-4	測試原理 4	187
6.3	白盒子技術	187
6.3-1	基礎路徑技術	188
6.3-2	繪製流程圖	192
6.3-3	回轉複雜度	193
6.3-4	基礎集	194
6.3-5	準備測試個案	194
6.4	環路測試技術	196
6.5	黑盒子技術	200
6.6	複查軟體測試策略	204
6.7	除 錯	205
6.8	除錯原理	206
6.8-1	除錯原理 1	206
6.8-2	除錯原理 2	208
6.8-3	除錯原理 3	208
6.9	給讀者的話	208
6.10	進修讀本	209
6.11	問 題	209

7

進行變更

213

7.1 基本修改原理 214

7.1-1 修改原理 1 214

7.1-2 修改原理 2 214

7.1-3 修改原理 3 215

7.1-4 修改原理 4 216

7.1-5 修改原理 5 216

7.1-6 修改原理 6 217

7.1-7 修改原理 7 218

7.1-8 修改原理 8 218

7.1-9 修改原理 9 219

7.1-10 修改原理 10 220

7.2 軟體表示法 221

7.3 軟體維護程序 221

7.3-1 組織責任 222

7.3-2 報告措施 223

7.3-3 維護事件之順序 224

7.4 給讀者的話 225

7.5 進修讀本 226

7.6 問題 226

附錄 A 軟體工程核對表 229

附錄 B 軟體品質保證 235

附錄 C 綜合性例子 241

索引 285

294